國立政治大學資訊科學系
Department of Computer Science
National Chengchi University

碩士論文
Master Thesis

基於免憑證的定時釋出加密系統以及其在可認證電子郵件系統之應用

Certificateless Timed-Release Encryption and Its Application to Certified Email System

研 究 生：林欣瑤
指導教授：左瑞麟

中華民國一百零一年七月
July 2012

基於免憑證的定時釋出加密系統以及其在可認證電子郵件系統之應用

Certificateless Timed-Release Encryption and Its Application to Certified Email System

研 究 生：林欣瑤　　Student：Shin-Yau Lin

指導教授：左瑞麟　　Advisor：Ray-Lin Tso

國立政治大學

資訊科學系

碩士論文

A Thesis
submitted to Department of Computer Science
National Chengchi University
in partial fulfillment of the Requirements
for the degree of
Master
in
Computer Science

中華民國一百零一年七月

July 2012

基於免憑證的定時釋出加密系統以及其在可認證電子郵件系統之應用

摘要

本論文提出了一個免憑證加密系統的方案，並且將此方案實作出來，使得此方案更具實用性。此方案主要架構為免憑證加密系統，利用此系統的特性消除傳統公開金鑰密碼系統中需要公開金鑰憑證認證的麻煩，也不會產生基於身分認證加密系統的私鑰託管問題，有效的結合了兩項系統的優點，並且提高了這兩種系統的安全性及方便性。本論文的協定中，在基於身分認證加密系統的公鑰部分還加入了階級以及時間戳記的概念，用以限制接收方取得部份私鑰的能力，並且也將接收方的部分公鑰加入其中，來增加部份私鑰的安全性。另外此協定也加入了提早解密金鑰的部分，可讓傳送方在傳出密文後更改解密時間，而不需要重新使用新的公鑰加密資訊，便可提早讓接收方取得相對應的明文資訊。

Certificateless Timed-Release Encryption and Its Application to Certified Email System

Abstract

In this paper, we propose a new certificateless public key encryption system, and implement it for securing e-mail systems. Certificateless cryptography, which is in contrast to traditional public key crypto-systems, does not require the use of certificates to guarantee the authenticity of public key. It does rely on the use of a trusted third party (TTP) who is in possession of a master key, just like the identity-based public key cryptography. However, certificateless public key crypto-system does not suffer from the key escrow property, whereas, it is a problem in the identity-based public key crypto-systems. Moreover, in our system, we add some new properties like level and time-stamp to limit the ability of receivers and to promote the safety of the system. Time-stamp ensures that the ciphertext cannot be decrypted before the indicated time and a level ensures that only the user with the corresponding identity and level can decrypt the ciphertext. In addition, a new feature is also introduced which is called the time-release encryption. Time-release encryption allows the encrypter to publish a release key so that the ciphertext can be decrypted by the receiver before the time indicated in the time-stamp when necessary.

致謝

一轉眼兩年的時間就這樣過去了，在政大的這兩年中，不只學到了學術上的知識，回想起剛開學時，實驗室還在整修，而同學們也都還互相不熟悉。剛好從日本來了一位短暫停留的研究生 Yanai，在短短一個月中，我們便與這位日本朋友建立了良好的友誼，也交到了一位外國朋友。可是天下無不散之宴席，很快的我即將離開這個熟悉的校園。

首先，我要誠摯的感謝我的指導教授左瑞麟老師，謝謝老師總是不厭其煩的告訴我錯誤的部分，並且耐心的教導我研究的內容並且指引我研究的方向，才能讓我能夠順利的通過口試。再來，我要感謝我們實驗室的凱彬、承峰、漢光，在我研究內容有疑問的時候與我討論，並且想出解決方法。還有學長士峰、致諺及圖學的明諺，謝謝你們陪我共同度過在這政大的兩年時光，因為有你們讓我的生活變得多采多姿。最後，我要謝謝我的家人，因為有你們的鼓勵和支持，我才能夠有現在的成就。

目錄

圖目錄

第一章　緒論

本章節大致介紹關於本篇論文的研究背景，詳細的介紹本論文的研究動機及目的以及本篇論文的貢獻，並且介紹本論文的基本架構。

1.1 研究背景

密碼學一開始使用於軍事戰爭中，在雙方交戰時為了避免讓敵方取得我方的軍事機密，因此開始產生了一些簡單的密碼學，但隨著時代及資訊科技的進步，密碼學的發展也越來越快。在電腦科技發展迅速的現在，密碼學已轉為主要針對電腦通訊傳輸以及電腦資料安全上的主要工具，並且不只將此使用於軍事戰爭中，也在許多的商業行為活動下，提升了電子商務的資訊安全性。

在 1976 年，Diffie 和 Hellman[1]提出了非對稱式的公開金鑰加密系統的概念，這項發表使得密碼學的領域及其應用變得更加的廣泛。由原來的對稱式金鑰加密系統，也就是加密及解密使用相同的鑰匙的觀念轉換到公開金鑰加密系統，亦即，加密與解密所使用的鑰匙分別為不同的一對鑰匙，這個概念改變了密碼學及網際網路訊息傳輸的發展。由於公開金鑰加密系統能解決傳統的對稱式金鑰加密系統所出現的密鑰分配問題、金鑰管理困難的問題及不可否認性的問題，因此公開金鑰加密系統很快的在網際網路傳輸訊息的使用上普及。善用公開金鑰加密系統所擁有之不可否認性的特性，因此也發展出了數位簽章(digital signature) [3]的理論。隨著數位簽章的發展越來越進步，使用率越來越高，隨之電子商務行為及 E 化的政府機關也逐漸的在許多的國家中普及。在這種環境的前提下，訊息的傳輸方式需要更有效更安全，或更有彈性的協議來避免訊息被竊聽。

而在 1984 年，Shamir[4]提出了基於身分認證的加密系統 IBE (identity-based cryptography)，此系統以另一種方式使用了公開金鑰加密系統，並且解決了公開加密系統中需使用公開金鑰基礎建設 PKI (Public key infrastructure)及需要憑證管理中心 CA (certificate authority)的條件。因傳統公開金鑰加密系統需要向 PKI 取得公開金鑰憑證的資訊，而在取得前必須先有認證之過程，而 IBE 則省略了這些過程。IBE 中用戶的公鑰是公開的，而用戶的公鑰便是用戶的公開身分資訊。用戶的公開身分資訊可用姓名+地址，手機電話號碼，身分證字號，或是 e-mail 等資訊來做為辨識，我們則稱這些資訊為 ID。我們在使用基於身分認證的加密系統進行保密與認證時，因已知對方的公開身分資訊，所以不需在資料庫中尋找，也不需要對公鑰的真實性做驗證，他人直接便可使用其公鑰。但在基於身分認證的加密系統中，所有使用者的私鑰都是由私鑰生成中心 PKG (Private key generation center)所產生。私鑰生成中心(PKG)利用系統主要密鑰(master key)而產生每個使用者的私鑰，因為如此私鑰生成中心(PKG)則知道每個用戶的私鑰，因此在安全性上也會造成金鑰託管的危險。

為了解決基於身分認證的加密系統(IBE)所產生私鑰託管的問題，因此由 Al-Riyami 和 Paterson[3]提出了免憑證公鑰加密系統(certificateless public key cryptography)。免憑證公鑰加密系統與需要使用 PKI 的傳統公鑰加密系統相比，免憑證公鑰加密系統與基於身分認證的加密系統(IBE)一樣不需要公鑰憑證，但是免憑證公鑰加密系統卻消除了基於身分認證的加密系統(IBE)所存在的私鑰託管問題。免憑證公鑰加密系統結合了傳統的公鑰加密系統及基於身分認證的加密系統(IBE)兩套系統中的優點，並且在一定程度上解決了兩套系統中的缺點。

雖然免憑證公鑰加密系統依舊存在著一個第三方的密鑰生成中心 KGC(key generation center)，KGC 一樣擁有系統主要密鑰(master key)，但是與私鑰生成中心(PKG)不同的地方在於 KGC 是根據用戶的身分及主要密鑰去計算用戶的部分私鑰，用戶收到部分私鑰後，再將接收到的部分私鑰與用戶自行挑選的密鑰產生

完整的私鑰，因此 KGC 便無法得到使用者完整的私鑰內容，因而達到克服基於身分認證的加密系統(IBE)所存在的私鑰託管問題。

1.2 研究動機與目的

在資訊科技及網際網路發展迅速的時代,有許多的個人及商業資訊都已數位化,雖然網際網路的發達及資訊的數位化使得我們的生活更加便利,但也使得個人或公司的機密資訊較容易為他人所取得。為了降低資訊被盜取的可能,便產生了像防火牆之類的方式來保護機密資訊,可是在機密資訊傳遞的情況下,機密資訊被從中竊取的機率增加,甚至會有被從中竄改的可能。在這些安全性的需求下便產生了許多不同的安全性運輸的協定或方式,像是 SSL 的安全性通道使得第三者無法取得資訊或是使用加密演算法,使得第三者取得文件但也無法解讀文件內容。

加密方式從最早的對稱式金鑰加密系統(Symmetric Key Cryptography),這種加密方式由於無法解決密鑰分配的問題,而產生後來的公開金鑰加密系統(Public Key Cryptography),另外,由於在公開金鑰密碼系統中公鑰憑證驗證的麻煩,才又出現了基於身分認證的加密系統(Identity-Based Encryption Cryptography)的加密方案。

基於身分認證加密系統則是不需使用公開金鑰憑證,只需使用關於個人資料的資訊做為公開金鑰。這種加密系統省去了公開金鑰憑證的麻煩,因此在實際的操作設計下,這種加密方式更容易實現在實際應用上。這種加密系統方便且容易應用,可是這種系統卻產生了私鑰託管的問題,因此本篇文章使用免憑證的加密系統(Certificateless Public Key Cryptography)來化解私鑰託管的問題。免憑證加密系統具有一項優點是有部分私鑰依舊由密鑰生成中心 PKG 所產生,在此系統中的使用者具有另一部份的私鑰。基於數學上的難問題,他人無法由密鑰生成中心所產生的部分私鑰推出使用者的機密值,因而達到資訊的安全性。並且可以利用完整私鑰內容需要密鑰生成中心所提供的部分私鑰的條件,控制接收方取得完整私鑰的時間點,因此可以做事先的文件傳送的部分,而不用擔心接收方提早得到資訊。

1.3 本文貢獻

本篇論文設計一個可實際運用在網際網路上的免憑證加密系統，並且實作這個系統。文章中利用免憑證加密系統的特性，解決了在基於身分認證的加密系統上私鑰託管的問題，使得免憑證加密系統在網際網路上的使用變得更加的廣闊。本論文除了消除了第三章中所提到的各種協定中的缺點外，使得文獻中的所有優點都被善加的利用，成為更加安全且有效利用的協定。

本研究結合了文獻[11][12][13]的優點。在這些文獻中我們發現了在使用秘密資訊交換時可在接收方的公鑰資訊上加上時間戳記，讓接收方只能在傳送方設定的時間點之後才能取得相對應的私鑰。但如果傳送方想要提早公開加密內容的資訊，則可不需透過 KGC 傳遞部分私鑰，而使用提早公開傳送方的機密值的方式，便可達到提早公布資訊的結果。在本論文中還加入了階級的概念，使得接收方的個人身分及階級被傳送方所限制。

我們利用了文獻[12]的概念修改了文獻[11]，使得我們協定的效能更好，花費的時間較低。並且取代了文獻[12]中無法提早公開加密內容資訊的問題，並且將文獻[13]中在基於身分認證的加密系統中的公鑰裡，加入使用者的部分公鑰，用以增加安全性。本篇論文將第三章中所提到的文獻優點集結，讓免憑證加密系統可運用的範圍更廣，在實際的傳輸上速度更快，安全性更佳，但卻不會增加使用者的麻煩。

1.4 論文架構

在本論文中，共分為六個章節來做探討，各章節內容架構大致如下：

- 第一章為緒論。以本篇論文的研究背景、研究動機及研究貢獻做簡單的介紹。
- 第二章為背景介紹，介紹關於後續會使用到的雙線性配對函數、在密碼學上的相關數學問題及數學上的難問題與定義做簡單的介紹，並對近代的加密系統做一完整的介紹。
- 第三章為相關文獻介紹，介紹對過去三篇相關文獻做簡要的介紹。
- 第四章為研究方法，介紹本篇論文主要提出的使用方式及解決方法。
- 第五章為安全性分析與系統實作，對於本篇論文提出的方式分析系統的安全性及探討實作結果。
- 第六章為結論與未來展望，對本篇論文做一完整的總結外，並提出此系統在未來能夠改進及探討的部分。

第二章　背景介紹

現今許多的加密方案中，都是使用到雙線性配對的運算，並且在這個運算下去實作。因此在本章中將會簡單的介紹雙線性配對的基本概念以及關於現代密碼學的基本介紹。

2.1 雙線性配對

本節將會介紹簡單雙線性配對的基本概念，並且針對雙線性配對的幾種數學難問題做介紹。

2.1.1 雙線性配對(Bilinear Pairing)

本篇文章中的演算法部分主要是使用雙線性配對的性質來計算，下面為介紹一些相關的基本運算。

雙線性配對是一線性映射函數(Bilinear Map)，由一個群對應到另外一個群，其中的 G_1 是一個循環式(Cyclic)的加法群(Additive Group)，而 G_2 是一個循環式的乘法群(Multiplicative Group)，G_1 及 G_2 當中的序(Order)皆為一質數 q，P 是 G_1 的生成元(Generator)，雙線性配對可表示為 $e: G_1 \times G_1 \rightarrow G_2$。假設離散對數問題(將在 2.1.2 中介紹)在 G_1 及 G_2 的群中都非常得困難，則我們可以得到下面三個性質：

(1) **雙線性(Bilinear)**：

所有的 $P, Q \in G_1$，所有的 $a, b \in Z_q^*$，我們會得到 $e(aP, bQ)=e(P, Q)^{ab}$

(2) **非退化性(Non-degenerate)**：

如果 P 是 G_1 中的生成元，那 $e(P, P)$也會是 G_2 的生成元，且滿足 $e(P, P)\neq 1$

(3) **可計算性(Computable)**：

如果所有的 $P, Q \in G_1$ 則存在有效率的演算法可計算 $e(P, Q) \in G_2$

2.1.2 相關數論介紹

密碼系統的安全性大多是建立在計算複雜度很高的難題上，大多都為數學理論上的因數分解及離散對數問題，以下為本篇論文會使用到的數學上的難問題。

定理一：

離散對數問題(Discrete Logarithm Problem ,DLP)：

首先定義一個乘法群 $G = Z_p^*$，接著找到一個生成元(generator) $g \in G$，其序(order)為 p-1，所有在乘法群 G 中的元素 y 均可表示為 $y=g^x \pmod p$，而 $1 \leq x \leq p$-1。給定 y, g, p，一般相信由 y, g, p 找到 x 滿足 $g^x = y \pmod p$是困難的，此一問題為離散對數問題。

定理二：

計算性 Diffie-Hellman 難問題(Computational Diffie-Hellman Problem, CDHP)：

令 p 為很大的質數，G 為一個乘法循環群，其序(order)為 q， g 是循環群 G 的生成元(generator)且 $p=2q+1$，在已知 g、$g^a \pmod p$ 、及 $g^b \pmod p$的條件下，算出 $g^{ab} \pmod p$是非常困難的，這個問題即為計算性 Diffie-Hellman 難問題。

定理三：

橢圓曲線下的計算性 Diffie-Hellman 難問題(Elliptic Curve Computational Diffie-Hellman Problem, ECCDHP)：

G_1 為一加法循環群，其序(order)為 q，且 $a,b \in Z_q^*$，如果 $P \in G_1$ 且 P 為 G_1 的生成元(generator)，在已知 P、aP、及 bP 的條件下，要求出解 abP 的問題是非常困難的，這個問題即為在橢圓曲線下的計算性 Diffie-Hellman 難問題。

定理四：

雙線性 Diffie-Hellman 難問題(Billnear Diffie-Hellman Problem, BDHP)：

G_1 為一加法循環群，其序(order)為 q，G_2 為一乘法循環群，其序(order)也為 q，且 e 是一個 bilinear map $e: G_1 \times G_1 \rightarrow G_2$。如果 $a,b,c \in Z_q^*$，$P \in G_1$ 且 P 為 G_1 的生成元(generator)，在已知 P、aP、bP、cP 的條件下，要求得 $e(P, P)^{abc} \in G_2$ 的問題是非常困難的，這個問題即為雙線性 Diffie-Hellman 難問題。

2.2 近代密碼學簡介

密碼學主要是研究如何秘密地傳送訊息的科學。直到近代之前的密碼學純粹指加密演算法，是將可理解的訊息轉換成難以理解的訊息，只有收到訊息的接收方才可利用解密演算法將訊息轉換成原本的形式。

一個用來加解密的系統稱為一個密碼系統(Cryptosystem)。加密前可解讀之訊息稱之為明文(Plaintext)，加密後的不可直接解讀之訊息稱為密文(Ciphertext)。其大概架構為圖一。

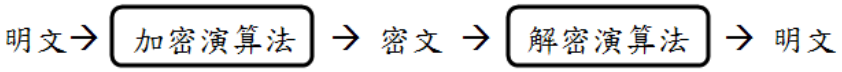

圖 一 密碼系統

在古典密碼學中，有兩個較為著名的加密方式，其一是換位加密法(Permutation Cipher)。主要是藉由改變字母順序達到加密效果的加密法，例如將「encryption system」改寫為「onpecyinrt mteyss」，而接收方在收到這份訊息後可利用解密本查到字母的調換順序，由此達到解密的效果。

另外一種則是替換加密法(Substitution Cipher)，例如著名的凱撒加密法，是將全部的字母替換成第三個字母。例如，將「encryption system」改寫為「gpetarvkqp uauvgo」，只有接收方知道如何將此訊息解密，在收到訊息後便可輕易的得到原本的訊息，因而達到機密性。

這兩種加密方式都是前人為了確保訊息的機密性而發展出的一些簡單的加密方式，但是這兩種都並不是十分安全，可能在傳送的過程中，除了傳送方和接收方外，有第三者擷取到資訊並且破解文件的內容，因此不足以提供足夠的機密性。

在現代的密碼體系中，大致將安全性的條件分成了五種：

1. 機密性：資訊內容除了傳送方與接收方外不應該讓第三者獲得

2. 完整性：確保接收方得到的資訊內容是完整的

3. 可用性：防止未經授權的第三者獨占系統或使系統當機，使得合法的使用者無法使用系統

4. 可驗證性：接收方可以確認資訊內容為傳送方所送出

5. 不可否認性：傳送方不可否認自己所發出的資訊

由於古典密碼學只達成機密性的安全性要求，因此近代為了滿足這些安全性的密碼系統便不斷的提出不同的加密協定，而目前這些密碼系統大致分成「對稱式金鑰加密系統」、「公開金鑰加密系統」，而在公開金鑰加密系統下，又分為「傳統公開金鑰加密系統」、「基於身分認證的加密系統」以及「免憑證公鑰加密系統」。

2.2.1 對稱式金鑰加密系統(Symmetric Key Encryption Cryptography)：

在使用這種加密系統前，傳送方與接收方必須先決定一對雙方共同持有的公共金鑰。在傳送訊息前，傳送方會利用這把金鑰對訊息加密，而接收方接到此加密後的密文後便利用相同的金鑰對密文解密。如圖二。

在這種加密系統中要確保的就是這把金鑰不能被第三者所取得，而在對稱式金鑰密碼系統中比較有名的就是 DES 密碼系統[7]及 AES 密碼系統[8]。

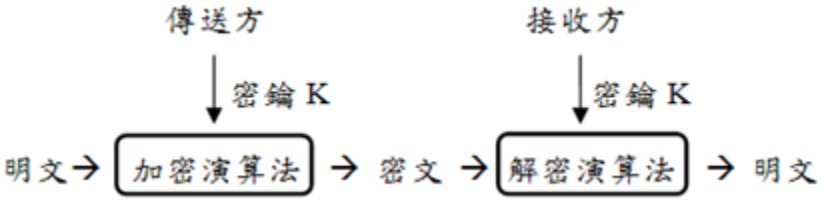

圖 二 對稱式金鑰加密系統

2.2.2 公開金鑰加密系統(Public Key Encryption Cryptography)：

公開金鑰加密系統又稱為「非對稱式金鑰加密系統」，這種密碼系統的特色不同於對稱式金鑰密碼系統，最大的特色就是加密與解密所使用的金鑰為兩種不同的金鑰。

在 1976 年，Diffie 和 Hellman[1]提出了非對稱式的公開金鑰加密系統[3] 的概念。這種密碼系統的金鑰是成對的方式存在，雖是一對不同的值但卻具有數學相關性質的一對金鑰。在這種加密系統中，兩把金鑰分別稱為「公開金鑰」以及「私密金鑰」。「公開金鑰」顧名思義便是公開給所有的使用者，而「私密金鑰」只為解密者所有，並且從數學計算中無法從「公開金鑰」推算出「私密金鑰」的數值。

在公開金鑰加密系統中，公鑰是開放式，可讓系統中所有的參與者皆可取得，但私鑰的部分只有自己擁有。一般使用此方案的加解密方式為：其他使用者利用公鑰將訊息加密給接收方，而接收方利用自己的私鑰去對訊息解密，如圖三。較為著名的系統為 RSA[9]、ElGamal[10] 等。

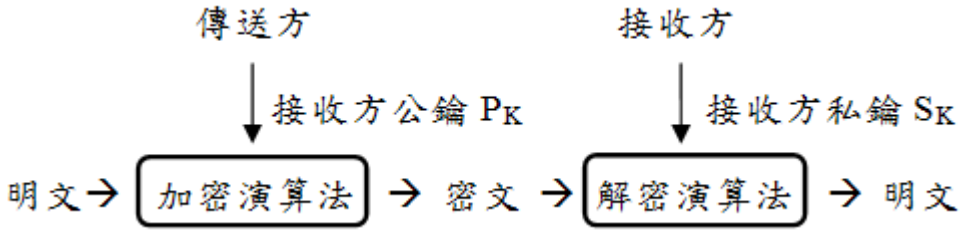

圖 三 公開金鑰加密系統

2.2.3 基於身分認證的加密系統(Identity-based Encryption Cryptography)：

而在 1984 年，Shamir[4]提出了基於身分認證的加密系統 IBE (identity-based cryptography)，此系統以另一種方式使用了公開金鑰加密系統，並且解決了公開金鑰加密系統中需使用公開金鑰基礎建設 PKI (Public key infrastructure)及需要憑證管理中心 CA (certificate authority)的條件。在基於身分認證的加密系統中所使用的公開金鑰的部分是利用使用者本身獨一無二的資訊來產生個別的公開金鑰，這些資訊像是電子郵件帳號、身分證字號、行動電話號碼…等等。而私鑰的部分則是由可信任的第三方扮演私鑰產生中心 PKG(Private Key Generator)的角色來產生相對應公開金鑰的私密金鑰，在此系統架構下的使用者可以減少在傳統公開金鑰系統中，使用者需對公鑰做認證及取得公鑰憑證的手續。

下列幾點要點為基於身分認證的加密系統(IBE)優於傳統基於公開金鑰基礎建設(PKI)的公鑰系統優點：

1.不需公鑰憑證，而且公鑰可由較容易計算及取得的資訊中得到，如：電話號碼，身分證字號，電子郵件地址等。

2.免除了需要憑證機構的麻煩。

3.容易分配所有的使用者公鑰，並且不需額外花費時間交換公鑰，因為所有的公鑰資訊接可直接取得。

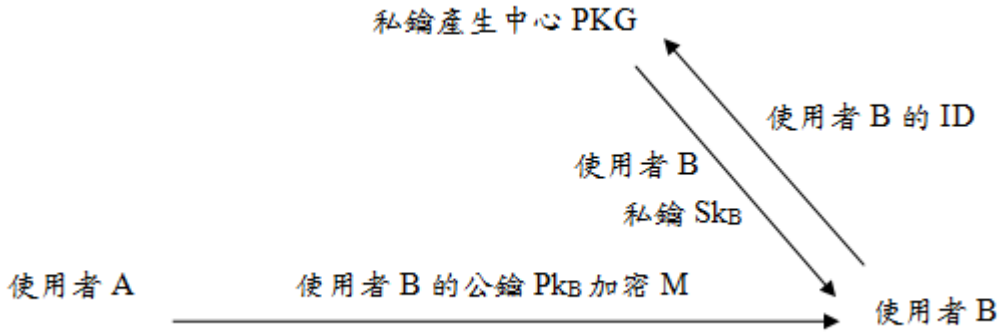

圖 四 基於身分認證加密系統

在基於身分認證的加密系統中，因已知對方的公開身分資訊，而公開金鑰的部分是公開且可以明顯辨識的型態，所以不需在資料庫中尋找，公開金鑰不需要利用傳統的基於 PKI 架構中的憑證來讓其他使用者相信這份公開金鑰，也不需要對公鑰的真實性做驗證，他人直接便可使用其公鑰。因此使用者彼此溝通之前，就不用再牢記每一位使用者所使用相對應的公鑰。

但在基於身分認證的加密系統中，所有使用者的私鑰都是由私鑰生成中心 PKG (Private key generation center)所產生。私鑰生成中心(PKG)利用系統主要密鑰(master key)而產生每個使用者的私鑰，因為如此私鑰生成中心(PKG)則知道每個用戶的私鑰，一旦遇到不誠實的 PKG 或者是針對 PKG 的破壞者或攻擊者，則所有使用者的私密金鑰則會遭到竊取或盜用，在安全性上也就會產生如中間者攻擊或 PKG 不可信任時的危險，而這就是所謂的「私鑰託管」問題。

在另外一個部分來看，PKG 對於使用者產生的私密金鑰必須透過一個安全不會被竊聽的通道來傳遞給使用者。而要在此系統中加入時戳的設計，便需要在每一個時間間隔中都必須重新建立一個新的安全性通道，如果使用者使用人數增加，系統則會面臨到效能上的問題。

2.2.4 免憑證公鑰加密系統(Certificateless Public Key Cryptography)：

為了解決基於身分認證的加密系統(IBE)所產生私鑰託管的問題，因此由 Al-Riyami 和 Paterson[3]提出了免憑證公鑰加密系統(certificateless public key cryptography)。免憑證公鑰加密系統與需要使用 PKI 的傳統公鑰加密系統相比，免憑證公鑰加密系統與基於身分認證的加密系統(IBE)一樣不需要公鑰憑證，但是免憑證公鑰加密系統卻消除了基於身分認證的加密系統(IBE)所存在的私鑰託管問題。免憑證公鑰加密系統結合了傳統的公鑰加密系統及基於身分認證的加密系統(IBE)兩套系統中的優點，並且在一定程度上解決了兩套系統中的缺點。

雖然免憑證公鑰加密系統依舊存在著一個第三方的密鑰生成中心 KGC(key generation center)，KGC 一樣擁有系統主要密鑰(master key)，但是與私鑰生成中心(PKG)不同的地方在於 KGC 是根據用戶的身分及主要密鑰去計算用戶的部分私鑰，用戶收到部分私鑰後，再將接收到的部分私鑰與用戶自行挑選的密鑰產生完整的私鑰，因此 KGC 便無法得到使用者完整的私鑰內容，因而達到克服基於身分認證的加密系統(IBE)所存在的私鑰託管問題。

而 Al-Riyami 和 Paterson 提出的免憑證公鑰加密系統主要包含了七個演算法，分別是**設定**，**部分私鑰提取**，**設定秘密數值**，**設定私密金鑰**，**設定公開金鑰**，**加密步驟**及**解密步驟**，下列分別對各個演算法流程做簡單的介紹。

設定：這個部分主要由密鑰生成中心(KGC)所執行。目的在產生一串公開系統參數以及 KGC 的主要密鑰的部分。

部分私鑰提取：此部分依舊為 KGC 所執行的部分，主要的工作在產生每個使用者的部分私密金鑰。

設定秘密數值：此部分由使用者自行執行，產生屬於自己個人的秘密數值，且此秘密數值只有使用者自己知道。

設定私密金鑰：使用者計算完秘密數值後，利用秘密數值及從 KGC 收到的部分私鑰，共同產生出自己的完整私鑰。

設定公開金鑰：使用者計算出自己的完整私鑰後，接著為計算出自己的公開金鑰。

加密步驟：在設定完所有的公開金鑰及私密金鑰後，此部分則是針對文件做加密動作。

解密步驟：這部分則是對加密的文件做相對應的解密動作。

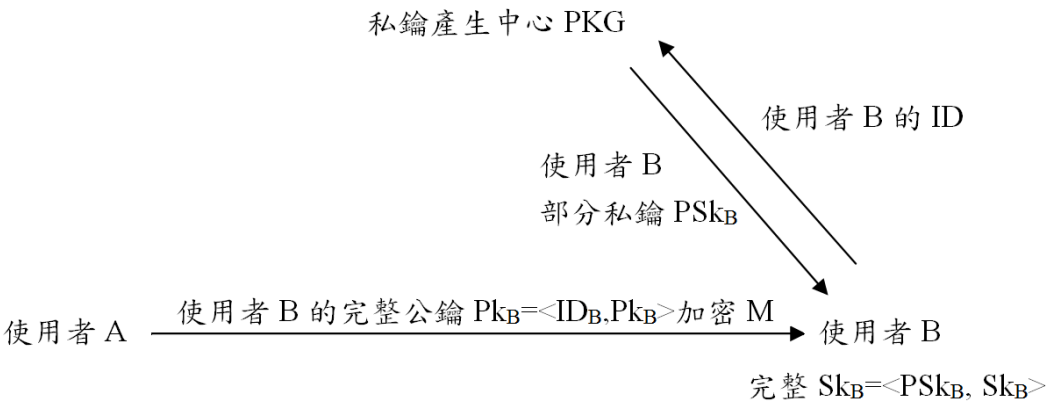

圖 五 免憑證公鑰加密系統

第三章 相關文獻介紹

本章節對於研究中所使用到的相關文獻做詳細的介紹，並提出文獻與本方案的優缺點提出討論。

3.1 Hwang 等學者提出的 *Timed-Release Encryption with Pre-open Capability and Its Application to Certified E-mail System*

此文獻由 Hwang 等學者所提出[11]，本論文使用此文獻作為研究的主要架構。

設定：密鑰生成中心(KGC)產生一串公開系統參數$< q, G_1, G_2, e, n, P, S, H_1, H_2 >$，$G_1$ 為一個加法群，G_2 為一個乘法群，q 為 G_1，G_2 的序，n 是一個大於 0 的整數，$P \in G_1$ 且 P 是 G_1 的生成元，S 是 KGC 的公鑰且 $S = sP$，$s \in Z_q^*$ 為 KGC 的主要密鑰，e 是一個 bilinear map $e: G_1 \times G_1 \rightarrow G_2$。明文長度為 n-bits，明文範圍為 $M=\{0,1\}^n$，而密文範圍為 $C= G_1 \times G_1 \times \{0,1\}^n$，單向雜湊函數 $H_1: \{0,1\}^* \rightarrow G_1$，$H_2: G_2 \rightarrow \{0,1\}^n$。

時間戳記提取： KGC 計算時間戳記為 $Q_t = H_1(t)$，系統時間到之後則會輸出 $TS_t = sQ_t$，其中，t 為傳送方所設定的解密時間，而接收方需在此時間後方可解密。

設定公開金鑰： A 使用私密金鑰 x_A 產生公開金鑰 $Y = x_A P$ 供其他使用者使用。

加密步驟：傳送方決定時間 t 後，然後選擇 $v \in_R Z_q^*$ 來做為提早解密的金鑰，再選擇一亂數 $r \in_R Z_q^*$ 並且依照下列方式加密訊息 M。

計算出密文 $C=<rP, vP, M\oplus H_2(g_t)>$，其中 $g_t = e(rY + vS, Q_t)$

產生提早解密金鑰：若傳送方想要提早給他人解密金鑰，則需計算出 $V_t=vQ_t$，並且將此結果提供給需要解密的使用者或是公開這項資訊。

解密步驟：在正常情況下，解密需由伺服器根據時間到之後 TS_t 公佈出來，解密者依照下面的方式將文件解密。

我們假設密文 $C = <U, V, W>$ ，則明文為 $M=W\oplus H_2(e(U, xQ_t) \cdot e(V, TS_t))$

若傳送方提前公開解密金鑰 V_t，則密文解密方式為 $M=W\oplus H_2(e(U, xQ_t) \cdot e(V_t, S))$

我們由以下公式可以確定這演算法的正確性：

$g_t = e(rY + vS, Q_t) = e(rY, Q_t) \cdot e(vS, Q_t) = e(rxP, Q_t) \cdot e(vsP, Q_t)$

$= e(rxP, Q_t) \cdot e(P, vsQ_t) = e(rP, xQ_t) \cdot e(vP, sQ_t)$

$= e(U, xQ_t) \cdot e(V, TS_t)$此為正常解密之步驟

$g_t = e(rY + vS, Q_t) = e(rY, Q_t) \cdot e(vS, Q_t) = e(rxP, Q_t) \cdot e(vsP, Q_t)$

$= e(rxP, Q_t) \cdot e(P, vsQ_t) = e(rP, xQ_t) \cdot e(vP, sQ_t) = e(rP, xQ_t) \cdot e(vQ_t, sP)$

$= e(U, xQ_t) \cdot e(V_t, S)$此為提前解密之步驟

文獻優點：傳送方不需將解密金鑰資訊提供給伺服器，而是使用傳送方及伺服器個別的兩把金鑰中的其中一把解密，並且在傳送方傳送資訊給伺服器之後，若想將其資訊提早公開或者提供給接收方，可公開提早解密金鑰，而不需透過伺服器給與部份私鑰後才可解密。

文獻缺點：解密時需要進行兩個雙線性配對的運算，所以需要耗費較多的時間以及資源，並且在這篇文章中的解密金鑰需由伺服器公布，因此還具有金鑰託管的問題。

3.2 Al-Riyami 等學者提出的 *Certificateless Public Key Cryptography*

此文獻由 Al-Riyami 等學者所提出[12]，本文中所使用的方案內容為文獻中的 basic CL-PKE Scheme 方案，並且將此方案運用於本文的加解密流程。

設定：密鑰生成中心(KGC)產生一串公開系統參數<$k, q, n, G_1, G_2, P, P_{pub}, e, H_1, H_2$>，$k$ 是一個安全參數，而 q 是一個 k-bits 的質數，n 是一個大於 0 的整數，G_1 是一個加法群，G_2 是一個乘法群，其中皆包含質數 q，$P \in G_1$ 且 P 是 G_1 的生成元，P_{pub} 是 KGC 的公鑰且 $P_{pub} = sP$，$s \in Z_q^*$ 為 KGC 的主要密鑰，e 是一個 bilinear map $e: G_1 \times G_1 \rightarrow G_2$。明文長度為 n-bits，明文範圍為 $M=\{0,1\}^n$，而密文範圍為 $C= G_1 \times \{0,1\}^n$，單向雜湊函數 $H_1: \{0,1\}^* \rightarrow G_1$，$H_2: G_2 \rightarrow \{0,1\}^n$。

部分私鑰提取：使用者 A 的 ID 為 ID_A，$ID_A \in \{0,1\}^*$，KGC 會計算 A 的公鑰 $Q_A = H_1(ID_A) \in G_1$，輸出部分私鑰 $D_A = sQ_A \in G_1$，並且確認 $e(D_A, P) = e(Q_A, P_{pub})$。

設定秘密數值：選定一個亂數 $x_A \in Z_q^*$，而 x_A 則為 A 的祕密數值。

設定私密金鑰：在此系統中 A 最後的私密金鑰為 A 的祕密數值與 IBE 的部分私鑰結合的結果，$S_A = x_A D_A = x_A s Q_A \in G_1$。

設定公開金鑰：在此系統中 A 最後的公開金鑰為 A 的原本公鑰與 IBE 部分公鑰的共同組合，A 最後的公開金鑰為 $P_A = <X_A, Y_A>$，其中 $X_A = x_A P$ 而 $Y_A = x_A P_{pub} = x_A sP$。

加密步驟：在下列的加密步驟中我們使用 A 的 $ID_A \in \{0,1\}^*$，並使用 A 的公鑰 $P_A = <X_A, Y_A>$ 對明文 $M \in M$ 加密：

1. 先確認 $X_A, Y_A \in G_1$，然後判斷 $e(X_A, P_{pub})$的結果與 $e(Y_A, P)$是否相等。
2. 計算 $Q_A = H_1(ID_A) \in G_1$。
3. 選擇一亂數 $r \in Z_q^*$。
4. 計算並輸出密文 $C=<rP, M \oplus H_2(e(Q_A, Y_A)^r)>$

解密步驟：我們假設密文 $C = <U,V> \in C$，使用 A 的私鑰 S_A 對密文 C 解密。

計算 $V \oplus H_2(e(S_A,U)) = V \oplus H_2(e(x_A s Q_A, rP))$

$= V \oplus H_2(e(Q_A, x_A sP)^r)$

$= V \oplus H_2(e(Q_A, Y_A)^r)$

$= M \oplus H_2(e(Q_A, Y_A)^r \oplus H_2(e(Q_A, Y_A)^r)$

$= M$

文獻優點： 在加密及解密的運算上皆只需要一個雙線性配對的運算，並且使用簡單的 XOR 運算計算明文及單向雜湊函數值，便可以達到免憑證的安全加密方式。

文獻缺點： 若傳送方需要將資訊提早公開則需要要求伺服器更改部分私鑰取得時間，再由伺服器提供部分私鑰給接收方解密。

3.3 Yang 等學者提出的 *An Improved Certificateless Authenticated Key Agreement Protocol*

此文獻由 Yang[13]等學者所提出，本論文將此文獻中所提出的 IBE 公鑰中加入了使用者原先的公鑰，使得原先公鑰不能被竄改。

設定：密鑰生成中心(KGC)產生一串公開系統參數< $G_1, G_2, e, P, P_0, H_1, H_2, n$ >，G_1 是一個加法群，G_2 是一個乘法群，其中皆包含質數 q，而 q 是一個 k-bits 的質數，k 是一個安全參數， e 是一個 bilinear map $e: G_1 \times G_1 \rightarrow G_2$，$P \in G_1$ 且 P 是 G_1 的生成元，P_0 是 KGC 的公鑰且 $P_0 = sP$，$s \in Z_q^*$ 為 KGC 的主要密鑰，單向雜湊函數 $H_1: \{0,1\}^* \times G_1 \rightarrow G_1$，$H_2: \{0,1\}^* \times \{0,1\}^* \times G_1 \times G_2 \times G_1 \times G_1 \times G_1 \rightarrow \{0,1\}^n$，$n$ 是一個大於 0 的整數，明文長度為 n-bits。

設定秘密數值：使用者 i 的 ID 為 ID_i，$ID_i \in \{0,1\}^*$，使用者 i 選定一個亂數 $x_i \in Z_q^*$，再計算出 $X_i = x_i P$，並將 X_i 的結果送至 KGC。

部分私鑰提取：KGC 會計算 i 的公鑰 $Q_i = H_1(ID_i, x_i P)$，輸出部分私鑰 $D_i = sQ_i$，並將 D_i 傳送給使用者 i。

設定私密金鑰：使用者 i 最後的私密金鑰為 i 的祕密數值與 IBE 的部分私鑰結合的結果，$S_i = x_i D_i$。

設定公開金鑰：使用者 i 最後的公開金鑰為 i 的原本公鑰與 IBE 部分公鑰的共同組合，i 最後的公開金鑰為 $P_i = <X_i, Y_i>$，其中 $X_i = x_i P$ 而 $Y_i = x_i Q_i$。

下面為使用者 A、B 使用此協定的例子：

使用者 A 於設定階段：

$Q_A = H_1(ID_A, x_AP)$

$D_A = sQ_A$

$P_A=<X_A,Y_A>=<x_AP, x_AQ_A>$

$S_A=x_AD_A$

使用者 B 於設定階段：

$Q_B = H_1(ID_B, x_BP)$

$D_B = sQ_B$

$P_B=<X_B,Y_B>=<x_BP, x_BQ_B>$

$S_B=x_BD_B$

使用者 A、B 分別產生會議金鑰：

使用者 A：

$a \in_R Z_q^*$

$T_A=aP$

將 X_A,Y_A,T_A 送至使用者 B

收到使用者 B 送出的 X_B,Y_B,T_B

$K_{AB}= e(aP_0+S_A, T_B+Y_B)$

$h=aT_B$

$K= H_2(Q_A, Q_B, h, K_{AB}, e(D_A, Q_B), x_AT_B, aX_B)$

使用者 B：

$b \in_R Z_q^*$

$T_B=bP$

將 X_B,Y_B,T_B 送至使用者 A

收到使用者 A 送出的 X_A,Y_A,T_A

$K_{AB}= e(T_A+Y_A, bP_0+S_B)$

$h= bT_A$

$K= H_2(Q_A, Q_B, h, K_{AB}, e(D_B, Q_A), bX_A, x_BT_A)$

A		B
$Q_A = H_1(ID_A, x_A P)$		$Q_B = H_1(ID_B, x_B P)$
$D_A = sQ_A$		$D_B = sQ_B$
$P_A = <X_A, Y_A> = <x_A P, x_A Q_A>$		$P_B = <X_B, Y_B> = <x_B P, x_B Q_B>$
$S_A = x_A D_A$		$S_B = x_B D_B$
$a \in_R Z_q^*$	$\xrightarrow{X_A, Y_A, T_A}$	$b \in_R Z_q^*$
$T_A = aP$	$\xleftarrow{X_B, Y_B, T_B}$	$T_B = bP$
$K_{AB} = e(aP_0 + S_A, T_B + Y_B)$		$K_{AB} = e(T_A + Y_A, bP_0 + S_B)$
$h = aT_B$		$h = bT_A$
$K = H_2(Q_A, Q_B, h, K_{AB}, e(D_A, Q_B), x_A T_B, aX_B)$		$K = H_2(Q_A, Q_B, h, K_{AB}, e(D_B, Q_A), bX_A, x_B T_A)$

圖 六 Yang 等學者的協定

文獻優點： 將接收方的部分長期公鑰放入基於身分認證的加密系統中的公鑰之中，除了可以確保接收方的部分長期公鑰為真正的使用者外，更重要的是可以避免中間者攻擊的情形發生。

文獻缺點： 使用者雙方的會議金鑰雖然是安全的，但其中有部分資訊是不重要的，因此花不必要的時間及資源做額外的資訊運算。

第四章 研究方法

本篇文章主要為設計一個利用免憑證加密系統傳遞一份重要文件。與傳統的加密系統不同的是不需做公開金鑰憑證的驗證，也不需做公開金鑰的傳遞。

4.1 提案方式

本提案方式包含了第三章所提到的三篇文獻[11][12][13]的協定部分。根據文獻[11]的協定，我們可以將時間戳記放入 IBE 的公鑰之中，並且接收方可使用 KGC 產生的部分私鑰或是傳送方給予的提早解密金鑰兩把金鑰的其中一把解密。而文獻[12]的協定，使用免憑證的加密系統以及簡單的 XOR 運算來對文件加密及解密。再加上文獻[13]的協定，將接收方的部分長期公鑰放入 IBE 系統中的公鑰，用以增加安全性。在下面的提案方式中，又加上了階級的條件，讓階級與使用者的個人身分共同用來限制使用者的解密條件。例如，使用者 *i* 為政治大學的學生，因此具有 nccu 的階級權限，此使用者 *i* 必須同時具有他個人的解密資訊及 nccu 的權限才能取得解密金鑰。若使用者 *i* 已畢業於政治大學，則使用者 *i* 便不具有 nccu 的權限，因此不能再向 KGC 取得解密金鑰。

此提案包含三個部分，分別為網頁、密鑰生成中心(KGC)、及資料庫，並於下列分別說明其用途。

網頁(使用者登記)：所有的使用者在使用本系統前可先於網頁上註冊基本資料，以方便其他使用者方便取得使用者的 IBE 公開金鑰及判斷使用者在 IBE 系統中的不同階級。

密鑰生成中心(KGC)：產生使用者的部分私鑰。

資料庫：主要存放所有使用者的基本資料及提供 PKG 使用者的公開金鑰。其中包含使用者的 ID，所擁有的權限階級，以及使用者的部分公鑰等資訊。

下列為本提案主要的協定部分：

參數設定：密鑰生成中心(KGC)產生一串公開系統參數<$k, q, n, G_1, G_2, P, P_{pub}, e, H_1, H_2$>，$k$ 是一個安全參數，而 q 是一個 k-bits 的質數，n 是一個大於 0 的整數，G_1 是一個加法群，G_2 是一個乘法群，其序為 q，$P \in G_1$ 且 P 是 G_1 的生成元，P_{pub} 是 KGC 的公鑰且 $P_{pub} = sP$，$s \in Z_q^*$ 為 KGC 的主要密鑰，e 是一個 bilinear map $e: G_1 \times G_1 \rightarrow G_2$。明文長度為 n-bits，明文範圍為 $\mathcal{M}=\{0,1\}^n$，而密文範圍為 $\mathcal{C}= G_1 \times \{0,1\}^n$，單向雜湊函數 H_1：$\{0,1\}^* \rightarrow G_1$，$H_2$：$G_2 \rightarrow \{0,1\}^n$。

設定秘密數值：使用者 i 的 ID 為 ID_i，$ID_i \in \{0,1\}^*$，使用者 i 選定一個亂數 $x_i \in Z_q^*$，其為秘密數值，再計算出 $X_i = x_i P$，並將 X_i 的結果透過公開通道送至 KGC。

部分私鑰提取：KGC 會計算 i 的公鑰 $Q_i = H_1(ID_i \| \text{level} \| \text{Timestamp} \| x_i P)$，輸出部分私鑰 $D_i = sQ_i$，並將 D_i 傳送給使用者 i，為了 D_i 的安全性，在這裡使用一個對稱式金鑰將 D_i 加密，然後將部分私鑰 D_i 加密後送出給解密者，在此系統中傳送方可在接收方的 IBE 公開金鑰中加入階級區分(level)和時間戳記(Timestamp)，用以限定接收方解密的權限及時間。接收方欲取得部份私鑰 D_i 需先登入網站後選取一亂數 k，並將 kP 值傳送給 KGC，而 KGC 利用 Diffie-Hellman 密鑰交換方式，可計算出會議金鑰 ksP，然後使用金鑰 ksP 將使用者的部分私鑰 D_i 加密傳至使用者，使用者接收到加密的 D_i 後使用相同的會議金鑰 ksP 解密，如圖七。

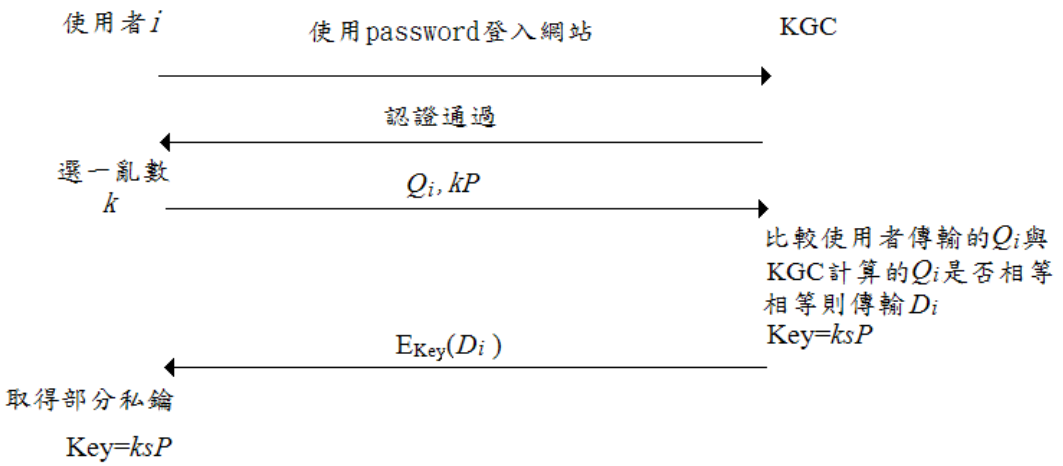

圖 七 使用者 i 取得部分私密金鑰流程圖

設定私密金鑰：使用者 i 最後的私密金鑰為 i 的祕密數值與 IBE 的部分私鑰結合的結果，$S_i=x_iD_i$。

設定公開金鑰：使用者 i 最後的公開金鑰為 $P_i=<X_i,Y_i>$，其中 $X_i=x_iP$ 而 $Y_i=x_i P_{pub}$。

加密步驟：在下列的加密步驟中我們使用 A 的 $ID_A\in\{0,1\}^*$，並使用 A 的公鑰 $P_A=<X_A,Y_A>$對明文 $M\in\mathcal{M}$ 加密：

1. 先確認 $X_A,Y_A\in G_1$，然後判斷 $e(X_A, P_{pub})$的結果與 $e(Y_A, P)$是否相等。
2. 計算 $Q_A = H_1(ID_A\|\text{level}\|\text{Timestamp})\in G_1$。其中，$ID_A$ 為使用者 A 的 email 帳號，level 可輸入學校名稱或是公司名稱，Timestamp 為輸入使用者 A 可取得解密金鑰的日期。
3. 選擇 $v\in_R Z_q^*$來做為提早解密的金鑰。
4. 計算並輸出密文 $C=<vP, M\oplus H_2(e(Q_A, Y_A)^v)>$

產生提早解密金鑰：若傳送方想要提早給接收方解密金鑰，也就是傳送方想要在 KGC 給使用者 i 產生 i 的私鑰 D_i 之前給接收方解密，則需計算出 $V_t=v P_{pub}$，並且將此結果提供給需要解密的使用者或是公開這項資訊。

解密步驟：在正常情況下，解密需由伺服器根據時間到之後將 D_A 利用之前建立好的會議金鑰 ksP 將 D_A 加密後提供給解密者，如圖七，解密者得到 D_A 後依照下面的方式將文件解密。

我們假設密文 $C = < vP, M \oplus H_2(e(Q_A, Y_A)^v)>=<U,V> \in \mathcal{C}$，A 將 $x_A D_A$ 組合成 S_A，並使用 A 的私鑰 S_A 對密文 C 解密。

計算 $V \oplus H_2(e(S_A,U)) = V \oplus H_2(e(x_A sQ_A, vP))$

$= V \oplus H_2(e(Q_A, x_A sP)^v)$

$= V \oplus H_2(e(Q_A, Y_A)^v)$

$= M \oplus H_2(e(Q_A, Y_A)^v \oplus H_2(e(Q_A, Y_A)^v)$

$=M$

若傳送者提前公開解密金鑰 V_t，則密文解密方式為下列方式：

計算 $V \oplus H_2(e(x_A Q_A, V_t)) = V \oplus H_2(e(x_A Q_A, vsP))$

$= V \oplus H_2(e(Q_A, x_A sP)^v)$

$= V \oplus H_2(e(Q_A, Y_A)^v)$

$= M \oplus H_2(e(Q_A, Y_A)^v \oplus H_2(e(Q_A, Y_A)^v)$

$=M$

下面為使用者 A(傳送方)、使用者 B(接收方)使用此協定的例子：

初始設定：

使用者 B 於網頁上輸入基本資料，假設電子郵件為 test@nccu.edu.tw，服務單位為 nccu，則使用者 B 的 ID_B 為 test@nccu.edu.tw，階級範圍為 nccu。

使用者 A 加密：

使用者 A 於網頁上輸入基本資料，並且取得使用者 B 的 ID_B 及 X_B。選擇一亂數 $v \in Z_q^*$。計算出 $Q_B = H_1$(test@nccu.edu.tw||nccu||07-12-2012||$x_B P$)，其中的解密時間則為 2012 年 7 月 12 日。加密者 A 使用 $P_B=<X_B,Y_B>=<x_B P, x_B P_{pub}>$ 加密明文 M，產生密文 C，加密公式如下：

$C=<vP, M \oplus H_2(e(Q_B, Y_B)^v)>$

並將密文 C 及可提取部分私鑰時間通知使用者 B。

使用者 B 解密：

使用者 B 接收到密文 C，並且登入網站取得 KGC 所提供的部分私鑰，KGC 使用當天的時間戳記及使用者 B 的階級產生出對應於 Q_B 的 $D_B = sQ_B$，並將 KGC 所計算出的 D_B 藉由會議金鑰將 D_B 加密傳至使用者 B，如圖七，使用者 B 計算出最後解密私鑰 $S_B = x_B D_B$，使用 S_B 將密文 $C=(vP, V)$ 解密，產生明文 M，解密公式如下，如圖八：

$$V \oplus H_2(e(S_B, vP)) = V \oplus H_2(e(x_B sQ_B, vP))$$
$$= V \oplus H_2(e(Q_B, x_B sP)^v)$$
$$= V \oplus H_2(e(Q_B, Y_B)^v)$$
$$= M \oplus H_2(e(Q_B, Y_B)^v \oplus H_2(e(Q_B, Y_B)^v)$$
$$= M$$

KGC　　產生 D_B 部分私鑰給接收方　　使用者 B

$$V \oplus H_2(e(S_B, vP))$$
$$= V \oplus H_2(e(x_B sQ_B, vP))$$
$$= V \oplus H_2(e(Q_B, x_B sP)^v)$$
$$= V \oplus H_2(e(Q_B, Y_B)^v)$$
$$= M \oplus H_2(e(Q_B, Y_B)^v \oplus H_2(e(Q_B, Y_B)^v)$$
$$= M$$

圖 八 接收方向 KGC 取得部分私鑰解密

若傳送者提前公開解密金鑰 V_t，則使用者 B 須計算出 $Q_B = H_1(\text{test@nccu.edu.tw}||\text{nccu}||\text{07-12-2012}||x_B P)$，並且依照下列的公式解密，如圖九：

$$V \oplus H_2(e(x_B Q_B, V_t)) = V \oplus H_2(e(x_B Q_B, vsP))$$
$$= V \oplus H_2(e(Q_B, x_B sP)^v)$$
$$= V \oplus H_2(e(Q_B, Y_B)^v)$$
$$= M \oplus H_2(e(Q_B, Y_B)^v \oplus H_2(e(Q_B, Y_B)^v)$$

$=M$

使用者 A　　公布 V_t 資訊給接收方　　公開於網站(使用者 B)

計算出 Q_B 值，並以下列方式解密

$$V \oplus H_2(e(x_B Q_B, V_t))$$
$$= V \oplus H_2(e(x_B Q_B, vsP))$$
$$= V \oplus H_2(e(Q_B, x_B sP)^v)$$
$$= V \oplus H_2(e(Q_B, Y_B)^v)$$
$$= M \oplus H_2(e(Q_B, Y_B)^v \oplus H_2(e(Q_B, Y_B)^v)$$
$$= M$$

圖 九 傳送方提早公開解密金鑰給接收方解密

第五章 安全性分析與系統實作

本章節將會針對上一章節所提出的免憑證系統做安全性分析，並且利用到 Random Oracle Model 證明工具設計出一套安全性模組，然後針對免憑證加密系統中的 TypeI 和 TypeII 兩種攻擊者去進行安全性假設，並將其假設導入至 BDH 難問題上，使用矛盾證明的方式來證明我們的加密方式為安全的。

5.1 證明方法介紹

本小節介紹在免憑證加密系統中，攻擊者可能企圖去取得使用者的部分私鑰或者是攻擊使用者，而根據 Al-Riyami 和 K.Paterson 的文獻[3]將可能的攻擊者分為兩種，我們在分析演算法的時候，將會分兩種攻擊者來做個別討論。本小節除了介紹兩種攻擊者外，也介紹關於 Random Oracle Model 的概念以及如何利用這個模組來證明演算法的安全性。

5.1.1 攻擊者介紹

TypeI 攻擊者：

通常假設這種攻擊者為 A_1，在本研究中，我們定義這種攻擊者有以下幾種特點：

●A_1 沒有能力取得 KGC 的主要密鑰

●A_1 不能得到目標 ID 的部分私密金鑰和使用者的祕密數值

●A_1 可替換目標 ID 的公鑰，$P_A' = x_A' P$，但替換時須同時提供相對應的秘密數

值 x_A'

TypeII 攻擊者：

通常假設這種攻擊者為 A_2，在本研究中，我們定義 A_2 通常有以下幾種特點：

●A_2 可以取得 KGC 的主要密鑰

●A_2 沒有辦法去替換使用者的公鑰部分

●A_2 不需要提取目標 ID 的部分私密金鑰，因 A_2 已知所有使用者的私鑰

5.1.2 Random Oracle Model

1993 年，Bellare 及 Rogway[14]指出雜湊函數可被視為隨機函數，使用雜湊函數的加密機制可以在 Random Oracle Model 下證明其安全性。為了證明一個加密演算法的安全性，我們必須導入 random oracle 的概念，random oracle 可以被視為完美的雜湊函數，在假設雜湊函數皆為完全隨機的情況下，任何一個機率多項式都沒辦法自行計算出此雜湊函數的輸出值，因此必須向外在的 random oracle 詢問此雜湊函數的輸出值。而 Random Oracle Model 的精神就是以可以藉由控制此外在的 random oracle 的輸出值以及一個假設可以破解目標系統的演算法，使其來破解某個公認的難問題，因為目前的難問題尚無解決方法，因此藉由反證法，可以證明我們欲證明的加密演算法式安全的。

5.2 安全性證明

此小節針對上個章節提出的加密演算法設置 Random Oracle Model 的模型，用以證明此演算法的安全性。並且根據上個小節所提出的攻擊者的能力不同，利用 Random Oracle Model 設計出兩個 Game。

在下面的 Game 中，我們簡稱上個章節所提出的系統為 CTRE，並且假設攻擊者 A 為 TypeI 攻擊者，攻擊者 B 則為 TypeII 攻擊者，而攻擊者 F 則為 BDH 的攻擊者進行下面 Game 的流程。Game1 是針對 TypeI 攻擊者做安全性證明，而 Game2 則是針對 TypeII 攻擊者做安全性證明。

Game1：攻擊者 A 為提案方式的攻擊者，目標則是破解選定密文攻擊(IND-CCA)的攻擊。所謂的選定密文攻擊是指攻擊者在開始攻擊之前可以選擇一些密文並從系統中獲得相對應的明文。攻擊者 F 提供提案方式的系統參數給攻擊者 A，並且將方案推倒至 BDH 難問題上。

設定：挑戰者 C 給攻擊者 F 一個 BDH 的難問題，其中 BDH 參數為$< q, G_1, G_2, e >$，這個參數對應到 BDH 難問題中的$< P, aP, bP, cP >$問題，因此攻擊者 F 會收到參數$< q, G_1, G_2, e, P, aP, bP, cP >$。攻擊者 F 選擇一個機密數值 $x \in Z_q^*$，然後計算出公鑰 $X=xP$。接著攻擊者 F 會給攻擊者 A 一串 CTRE 的系統參數$< q, n, G_1, G_2, P, P_{pub}, e, H_1, H_2 >$，其中的 n 為 H_2 中所對應到的明文長度，而在參數中的 $P_{pub}=aP$。攻擊者 F 依照下列的方式模擬 random oracle H_1 及 H_2，目標則是輸出 $e(P, P)^{abc}$ 的結果。

***H_1*-queries**：攻擊者 A query H_1 關於$< ID_i, l_i, t_i, P_i >$的資訊，則攻擊者 F 必須在列

表中找尋是否有參數值符合< ID_i, l_i, t_i, P_i, k_i, Q_i>，我們將這串資訊稱為 H_1^{list}，在 H_1^{list} 中一開始的資訊為空的。攻擊者 F 在 H_1-queries 回答前會先產生一個亂數 j，且 $1 \leq j \leq q_{h1}$，其中，q_{h1} 為可 query 的最多次數，j 為 query 的次數到第 j 次。

1.如果 query < ID_i, l_i, t_i, P_i>並發現其< ID_i, l_i, t_i, P_i>的資訊已存在於 H_1^{list} 中，並且具有下列參數< ID_i, l_i, t_i, P_i, k_i, Q_i>，則會回傳 H_1^{list} 中的 $Q_i = H_1 (ID_i \| l_i \| t_i \| P_i)$。

2.否則，攻擊者 F 會選擇一個隨機的亂數 $k_i \in Z_q^*$，然後計算 $Q_i = k_i P$。並且將產生新的參數< ID_i, l_i, t_i, P_i, k_i, Q_i>加入到 H_1^{list} 中。

3.注意:如果 query < ID_j, l_j, t_j, P_j>則攻擊者 F 會回答 $Q_j = cP$ 而不是 $k_j P$。

***H_2*-queries**：在任何時間下，攻擊者 A 都可以任意的 query random oracle H_2。攻擊者 F 必須在列表中找尋是否有參數值符合< g_i, h_i >，我們將這串資訊稱為 H_2^{list}。

1.若於 query g_i 時發現 g_i 的資訊已在 H_2 中被 query 過，則攻擊者 F 會回傳在 H_2^{list} 中的 $h_i = H_2 (g_i)$。

2.否則，攻擊者 F 會選擇一個亂數 $h_i \in \{0,1\}^n$，然後將參數< g_i, h_i >加入到 H_2^{list} 之中，並會回傳 h_i。

接著攻擊者 A 做 extraction oracle 和 decryption oracle 的 queries，以及 key replacement，攻擊者 F 將會模擬下面的 random oracle。

Extraction-queries：攻擊者 A query 關於 Q_i 的資訊，包含 Q_i 的部分金鑰 D_i。攻擊者 F 必須在列表中找尋是否有參數值符合< ID_i, Q_i, D_i>，我們將這串資訊稱為 Ex^{list}。

1.如果 $Q_i = Q_j$，攻擊者 F 會回傳錯誤，並且會停止執行。

2.如果 Q_i 的資訊已被 query 過，則攻擊者 F 會在中 Ex^{list} 找到 D_i 的資訊，並且將 D_i 回傳。

3.否則，攻擊者 F 會從 H_1-queries 之中取得< ID_i, l_i, t_i, P_i, k_i, Q_i>的資訊，並且計算出 $D_i = k_i P_{pub}$ ($=k_i aP=aQ_i$) $\in G_1$，並且將參數< ID_i, l_i, t_i, P_i, k_i, Q_i, D_i>

紀錄至 Ex^{list} 之中，然後將結果回傳給攻擊者 A。

Decryption-queries：攻擊者 A 要求密文 C_i=<U, V>對應的明文 M_i，因此攻擊者 A query 關於 Q_i 及 Pk_i=<X_i,Y_i>的資訊，其中，$Y_i = x_i P_{pub}$，Y_i 可以是原始的公鑰或是替換之後的公鑰。

A.攻擊者 F 得到 query 的要求後，會先檢查 Pk_i 是否為正確的資訊，若 Pk_i 的資訊為正確的，則會做下列的步驟：

1.如果 $Q_i = Q_j$，攻擊者 F 會回傳錯誤，並且會停止執行。

2.如果 Q_i 在 H_1^{list} 中，則 $Q_i = k_i P$，而 k_i 存在於 H_1^{list} 之中，所以攻擊者 F 可以查到 k_i 的資料。

3.否則，攻擊者 F 會先設定 $Q_i = k_i P$，再將 Q_i 的資訊存入 H_1^{list} 之中。

B.因為 $e(Q_i, Y_A)^v = e(k_i P, Y_A)^v = e(vP, Y_A)^{k_i}$，所以攻擊者 F 會先確認 $e(vP, Y_A)^{k_i}$ 是否存在於 H_2^{list} 之中。

1.如果 $e(vP, Y_A)^{k_i}$ 的結果存在於 H_2^{list} 之中，攻擊者 F 會在 H_2^{list} 之中找到 $H_2(e(vP, Y_A)^{k_i})$的結果，並將 $H_2(e(vP, Y_A)^{k_i})$的結果設為 T。明文結果即為 $M_i = V \oplus T = V \oplus H_2(e(vP, Y_A)^{k_i})$，攻擊者 F 最後會將 M_i 的結果輸出。

2.否則，攻擊者 F 會設一亂數 $T' = H_2(e(vP, Y_A)^{k_i})$，並且將 T'的結果存入 H_2^{list} 之中，然後輸出明文 $M_i = V \oplus T'$的結果。

Key-replacement：若攻擊者 A 要替換公鑰的部分，則攻擊者 A 須提供<X_i,Y_i,Sk_i>

的資訊，攻擊者 F 將替換的結果紀錄在 Kr^{list} 中。

Challenge：攻擊者 A 產生兩個相等長度的訊息$< M_0, M_1 >$，而且挑戰的目標為 Q_i，須提供$< M_0, M_1 >$，Q_i，Pk_i 的資訊給攻擊者 F。

1.如果 $Q_i \neq Q_j$，攻擊者 F 會回傳錯誤，並且會停止執行。

2.否則，攻擊者 F 會選擇一串字串 $R \in \{0,1\}^n$，然後回傳 $C=<bP, R>$給攻擊者 A。

依照系統方案，若要解密 C，則必須計算 $R \oplus H_2(e(S_i, bP)) = R \oplus H_2(e(x_i a Q_i, bP))$。

Guess：當攻擊者 A 成功猜出 bit b′的結果時，則 $R \oplus M_b = R \oplus H_2(e(x_j a Q_j, bP))$，而基於 random oracle 的性質，$H_2(e(x_i a Q_i, bP))$有很大的機率在 H_2-queries 的步驟之中被問過，所以攻擊者 F 可在 H_2^{list} 之中找到相對應的內容，其為 $e(x_j a Q_j, bP)$。因為 $Q_j = cP$，所以 $e(S_j, bP) = e(x_j a Q_j, bP) = e(x_j acP, bP) = e(P, P)^{abcx_j}$。又$<Pk_j, Sk_j> = <X_j, Y_j, Sk_j> = <x_j P, x_j P_{pub}, x_j>$，如果 Pk_j 是原始的 Pk_j 值，則 x_j 為攻擊者 F 的自選值，若 Pk_j 為攻擊者 A 產生的，則 x_j 可由 Kr^{list} 中找到。所以$(e(P, P)^{abcx_j})^{x_j^{-1}} = e(P, P)^{abc}$ 即為 BDH 問題的解答。

Game2：攻擊者 B 為提案方式的攻擊者，目標則是破解選定密文攻擊(IND-CCA)的攻擊。攻擊者 F 提供提案方式的系統參數給攻擊者 B，並且將方案推倒至 BDH 難問題上。

設定：挑戰者 C 給攻擊者 F 一個 BDH 的難問題，其中 BDH 參數為< q, G_1, G_1, e >，這個參數對應到 BDH 難問題中的< P, aP, bP, cP >問題，因此攻擊者 F 會收到參數< $q, G_1, G_2, e, P, aP, bP, cP$ >。攻擊者 F 選擇一個亂數 $s \in Z_q^*$，然後計算出公鑰 $P_{pub} = sP$。接著攻擊者 F 會給攻擊者 B 一串 CTRE 的系統參數< $q, n, G_1, G_2, P, P_{pub}, e, H_1, H_2$>，並將 s 值傳送給攻擊者 B，其中的 n 為 H_2 中所對應到的明文長度，以及給攻擊者 B 公鑰 $Y=aP$。而攻擊者 F 依照下列的方式模擬 random oracle H_1 及 H_2，目標則是輸出 $e(P, P)^{abc}$ 的結果。

***H_1*-queries**：攻擊者 B query H_1 關於< ID_i, l_i, t_i, P_i>的資訊，則攻擊者 F 必須在列表中找尋是否有參數值符合< $ID_i, l_i, t_i, P_i, k_i, Q_i$>，我們將這串資訊稱為 H_1^{list}，在 H_1^{list} 中一開始的資訊為空的。攻擊者 F 在 H_1-queries 回答前會先產生一個亂數 j，且 $1 \le j \le q_{h1}$，其中，q_{h1} 為可 query 的最多次數，j 為 query 的次數到第 j 次。

1. 如果 query< ID_i, l_i, t_i, P_i>並發現其< ID_i, l_i, t_i, P_i>的資訊已存在於 H_1^{list} 中，並且具有下列參數< ID_i, l_i, t_i, P_i, Q_i>，則會回傳 H_1^{list} 中的 $Q_i = H_1 (ID_i \| l_i \| t_i \| P_i)$。
2. 否則，攻擊者 F 會選擇一個隨機 $k_i \in Z_q^*$，然後計算 $Q_i = k_i P$。並且將產生新的參數< ID_i, l_i, t_i, P_i, Q_i>加入到 H_1^{list} 中。

3.注意:如果 query< ID_j, l_j, t_j, P_j>則攻擊者 F 會回答 $Q_j = cP$ 而不是 $k_j P$。

***H_2*-queries**：在任何時間下，攻擊者 B 都可以任意的 query random oracle H_2。攻擊者 F 必須在列表中找尋是否有參數值符合< g_i, h_i >，我們將這串資訊稱為

H_2^{list}。

1.若於 query g_i 時發現 g_i 的資訊已在 H_2 中被 query 過，則攻擊者 F 會回傳在 H_2^{list} 中的 $h_i = H_2(g_i)$。

2.否則，攻擊者 B 會選擇一個亂數 $h_i \in \{0,1\}^n$，然後將參數$< g_i, h_i >$加入到 H_2^{list} 之中，並會回傳 h_i。

Decryption-queries：攻擊者 B 要求密文 $C_i=<U, V>$對應的明文 M_i，因此攻擊者 B query 關於 Q_i 及 $Pk_i=<X_i,Y_i>$的資訊，其中，當 $Q_i \neq Q_j$ 時，$X_i= x_iP$，$Y_i= x_iP_{pub}$。當 $Q_i = Q_j$ 時，$X_i= cP$，$Y_i=cP_{pub}$。另外 Pk_i 必須是原始的公鑰。

A.攻擊者 F 得到 query 的要求後，會先檢查 Pk_i 是否為正確的資訊且為原始而非攻擊者 B 自選之公鑰。若 Pk_i 的資訊為正確的，則會做下列的步驟：

1.如果 $Q_i = Q_j$，攻擊者 F 會回傳錯誤，並且會停止執行。

2.如果 Q_i 在 H_1^{list} 中，則 $Q_i= k_iP$，而 k_i 存在於 H_1^{list} 之中，所以攻擊者 F 可以查到 k_i 的資料。

3.否則，攻擊者 F 會先設定 $Q_i= k_iP$，再將 Q_i 的資訊存入 H_1^{list} 之中。

B.因為 $e(Q_i, Y_A)^v= e(k_iP, Y_A)^v= e(vP, Y_A)^{k_i}$，所以攻擊者 F 會先確認 $e(vP, Y_A)^{k_i}$ 是否存在於 H_2^{list} 之中。

1.如果 $e(vP, Y_A)^{k_i}$ 的結果存在於 H_2^{list} 之中，攻擊者 F 會在 H_2^{list} 之中找到 $H_2(e(vP, Y_A)^{k_i})$的結果，並將 $H_2(e(vP, Y_A)^{k_i})$的結果設為 T。明文結果即為 $M_i=V\oplus T= V\oplus H_2(e(vP, Y_A)^{k_i})$，攻擊者 F 最後會將 M_i 的結果輸出。

2.否則，攻擊者 F 會設一亂數 $T'=H_2(e(vP, Y_A)^{k_i})$，並且將 T'的結果存入 H_2^{list} 之中，然後輸出明文 $M_i=V\oplus T'$的結果。

Challenge：攻擊者 B 產生兩個相等長度的訊息$< M_0, M_1>$，而且挑戰的目標為 Q_i，

須提供< M_0, M_1>，Q_i，Pk_i 的資訊給攻擊者 F。

1.如果 $Q_i \neq Q_j$，攻擊者 F 會回傳錯誤，並且會停止執行。

2.否則，攻擊者 F 會選擇一串字串 $R \in \{0,1\}^n$，然後回傳 $C=<bP, R>$給攻擊者 B。

依照系統方案，若要解密 C，則必須計算 $R \oplus H_2(e(S_i, bP)) = R \oplus H_2(e(asQ_i, bP))$。

Guess：當攻擊者 B 成功猜出 bit b′的結果時，則 $R \oplus M_b = R \oplus H_2(e(asQ_j, bP))$，而基於 random oracle 的性質，$H_2(e(asQ_i, bP))$有很大的機率在 H_2-queries 的步驟之中被問過，所以攻擊者 F 可在 H_2^{list} 之中找到相對應的內容，其為 $e(asQ_j, bP)$。因為 $Q_j=cP$，所以 $e(S_j, bP)= e(asQ_j, bP)= e(ascP, bP) = e(P, P)^{abcs}$。又 s 為攻擊者 F 的自選值，所以$(e(P, P)^{abcs})^{s^{-1}}=e(P, P)^{abc}$ 即為 BDH 問題的解答。

5.4 系統實作

此小節將介紹本論文所提出的協定實作的結果。

5.4.1 實作環境

實作部分主要分為兩個部分，一為使用者網站登入環境及資料庫的存取，二為主要協定的部分。使用者網站登入環境及資料庫的存取部分，使用 PHP 以及 MySQL 的語言撰寫，執行環境是利用個人電腦(2.50 GHz Pentium(R) Dual-Core, 4GB 記憶體)執行，使用開發環境為 phpMyAdmin 2.10.3 的版本存取資料庫的資訊。主要協定的部分使用 JAVA 語言撰寫程式碼，執行環境是利用個人電腦(2.50 GHz Pentium(R) Dual-Core, 4GB 記憶體)執行，並且使用開發環境 NetBeans IDE 7.1.2 版本來執行程式碼。

5.4.2 系統流程

本論文將實作系統分為使用者使用介面及協定運算兩個部分。

使用者使用介面

使用者進入系統初始網頁，並可在網頁上進行新增個人資訊，登入個人系統，索取部分私鑰，以及更新系統使用者資訊列表等流程，如下圖十。

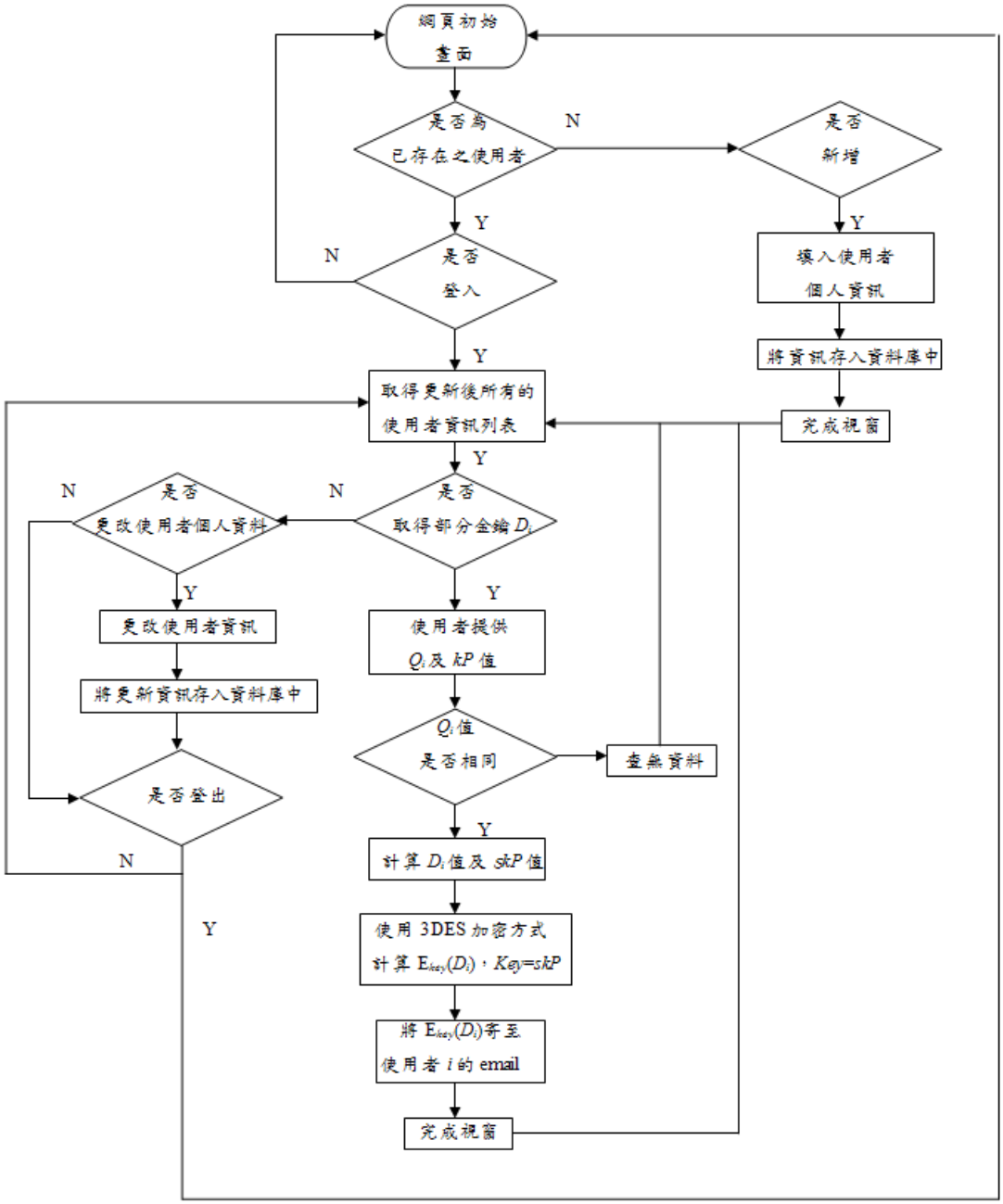

圖 十 使用者使用系統流程

協定運算

使用者索取金鑰時，系統會執行此運算，並將運算的結果送至資料庫中，流程如下圖十一。

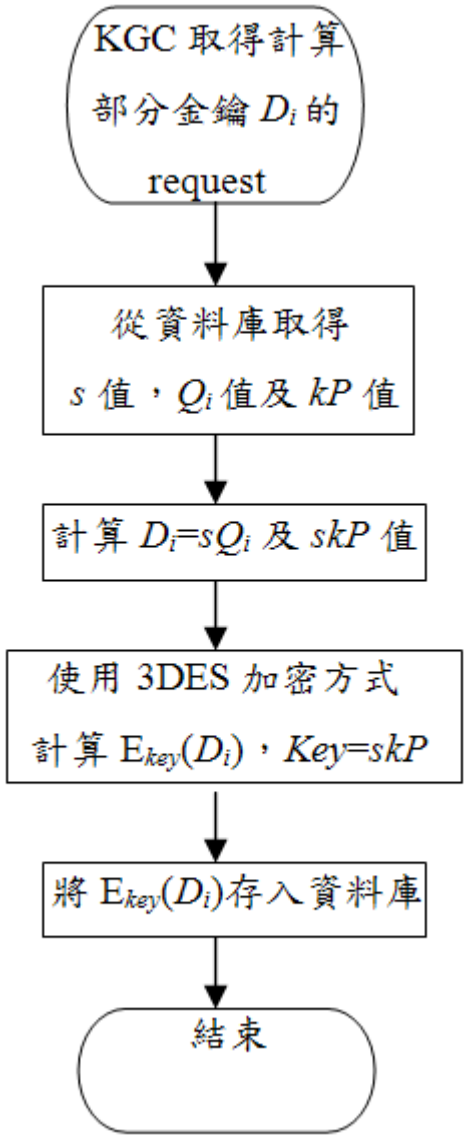

圖 十一 KGC 計算使用者部分私鑰 D_i 流程

實作流程畫面

下列為執行此實作系統的畫面，首先使用者於新增畫面輸入使用者資訊，如圖十二。

圖 十二 使用者新增畫面

接著，使用者的資訊會出現在使用者列表之中，如圖十三。

姓名	服務單位	電子郵件	部分公鑰
王大明	nccu	123@nccu.edu.tw	123123123
蔡小華	nccu	456@nccu.edu.tw	456456456
林欣瑤	nccu	99753009@nccu.edu.tw	99753009

圖 十三 使用者列表

若使用者需索取部分解密金鑰D_i，則需向伺服器提供從傳送方取得的Q_i值，以及自行選取一亂數 k 並提供伺服器 kP 值，如下圖十四。

圖 十四 索取部分金鑰 D_i 值

接收方取得使用 3DES 加密後的私鑰 D_i，如圖十五。

```
MFy57AkHPDm8fE5qFEKfjSNNiS5OfFUF57Cosver4p4ZBTKB7B9XssLsRTPf3eqZDOki0tbM0oD79DQ/8OdGjOC+Zbg1b4cCiol3CifIKDFjj60xF2H1NQ==
CEk6cBw/RhD4cdDZXg3GldX1oSFtd26cHcfixLmH/GAKBKEOIVIVA6pYRNYIhIgVkbHHUjI/s+uCFCs5x8BvrRw5EuwwiwVffv0I5wAN5+RFHtib6hXfJo2ndEsM1Y1
baM5s+/uQALzpkGegzl7SWGUMNGXNHTRzbGLpOXt3Dsc0jDFnod5FyOgg+E/Hpudr+Rtff51YYIdgBm+S7ysmYQzqGJrhZQ59VNPjWwUpC9O7JfgxgCbHA==
9STu59T6wu9vPIQQ4fNt4Edhdqc0+80vc4m26rUZq1b1ODv+Yi7D+xpN71kYLuCw2ptc0n1Bay3nixq8X2IX1bUy8ACAW51VD6E3iJvS/2ZgmnW/yGuVdg==
cbZwsaEcGsZ9FCeHC09ZWH0jMwMAqv0U7Dms0pM5y6YRXA+DMU1oM1Zys1CEUgWzBM5p+FE2Q1aA+A0ITRFbfvyJfSTU7FKI2d3N7Gotoc5PbqDrKLaHkw==
4x1pOrbO1Fr8zd1RjPstChSiHbqPsmbb+9qz8CZhbt85xzmdNcXi+bFHO1Pxnwe1qVu4EKkTNrJxdc3WiXoY3cujLjEwxcvaFxImYkO5MDTbeUX998U8fw==
XoESzXpTNjwlkcmdp9G152f++TmaMEmq+H5e6fqH5+v+qgqFe7+MjEpk0E0JVIrHU1Bhv3mO8YiBU6b6BZ1g7ykkuiBokQoHvWQJRFcirR+HQm9cSJuXgg==
OP5zxc/T1bB+BGCesOs1U7g1vwSADIVhNCsngNm+oXfUscON00UDi+QJ2/qxBfI5qRrccIDAnt7kNKrY9F+sedU8MgqxfHpN
z2N42r6YQ6AI9RikF2ZyQ21fFG/ntvO5VwDDQ6PD6ewrVEs2z1FbYUmRgCKWiDoI9JR5PRaH57nIsyy/bUYDTm6xqCvePR3kibo3KP5bAHg=
SNpoj5f/LH/TYNR6t+jDEbY+4ZcpJXDOKudj+DA10T7gRnDgPRCScI291/B04aD2R9IxGoUjGOhNNIArzRNNY9eeYtho/45DSgAaPBCAKwdmrq2JGZd0xw==
oHEe0ZBObPJd86G4SGBBBvjcH/PJP4qqjWrGnlJrHNL5a8NlvHu44OiX8ntDD+kPoVtXtoCsk7eQhRDiKXK9DM+/FWQ1LgCc18hX50cAgriWA2E53KNApA==
aJSeO4AQ7bQJOiEFRSm2IUv41xCLgvmEndnw02aZBrSLo4dzc4bPEoYQ10De6pbqQZ6OU35eMN1N4av0tGT43DF+50azXSbxIgnoq6fOLW2RgmVOhETqrQr8h2cb5k05
JKxjTPWmsKBOo040U27upjkd1QegqnWD+2w4JZEhaTwUiwTMu8G6wK7h4WE3H704aaIDIlOzm/CGgs2PAr5QXkszSXU/s4leLZ11NYduuCsuU+8c7vRhR+k+AwSXrlPW
K0T4U1NWATjWkx1FtOEX4fAv2/cM6NR0y0RAqLYPLg599FX878wXRwq2Uq+kUcZg2WwOJZrQpLWH97+Mih/uVehwRIooO6GYbzDFYnI2Z6NpYLWKs8CwqCJYqQOCGQ2q
DUtvSRg4QVuIxfcR9ngBw4NAMdZl6qXkzAzOJ+ex5rDlGAFoWd4VZuFTUoOZ4wB17cMj1vP/7e8RSW6PRX2TyyUQPjpqz/88r7/gwg5q0dKKLcLB/ghdsA==
61gp6HsHBXW0foMpQrL7fHOCCKTciOj/opsAV2MYEzLaUhBusOkvjhwNg99Dv0qWt2gCFv8/qKV80BMgbhuC/0B+5xKQ2ggAe4INtFh7C6md65yqa18jk86kKBCFTbm+
cUgnEv1awpuY+spJZwVrV3K0Bcx3WhzS
```

圖 十五 使用 3DES 加密的私鑰 D_i

接收方解密後取得私鑰 D_i 的內容，如圖十六。

```
EFqzGc1YjTe0oVXu8OjYvP+JFUTxLDwDM2bLiEN3v1SoGi/Tmxi5cKCGTAFoIwgkESXa8UTy4Z378UqDbZwJjn4tWK7+FEedE6Wv6VKP6yG2xAnXBJMtvqbU6Mq+1P5GgPHgjU4CzoBpehFihxtGafGdqrXB
Efi2xb3cyg0qeGxUhHB76HLz6OkzFXYqe0waJuKY1bZktgu5j3AdUcN0MF7u086QO6GnIU/U+BvJ78tsz8coIxyUpd2+C91+qbrwQpQQdOFFCGC1eoWeueFxBEddeNKu6gic3pI1VzOhyFxyRL+hJyWEcHel
noxtAi91A0V2gYuu06bsxM6cRcKiveM00MIL/LTn1SEL0YAeFrVP18YBr7bNFmvA1ESXoc24Iaz/dr/u1vkUPZo60H9cjMlghJbVgY4QuIjCi0J2+6NWt3ap1THtfjXTO8UnI12X0tGkrSF+v0/2PN+gj0Qn
01vGbkhUEzeg77mCFEQx237mqsP0wRMv2RKNRgIQh9umx2qQJCcJ1ZYQXY4Uc1xunNtIaZnm0LAm1SsIPaf4+jPAyiQRfbDYOZWFY6S2oCvQ0Wgnq4gotvvGe7nqXFo1wvR3nR70jcUjU1ijfU30ODy/9fGC
vSfwF2esWUom4lg6k315Uhz1r4fq+99zEBUIOPK09E3bVKiRkx1RG8BnIB8y6E+nB5hHgaltViPH/q5tjmO02FGqIRYBGbNEsakmGp/1EyUsMIwp/vha1nnhAvmZh6xWjAlZuVrUupi8Tk1om8MsDVRS6nop
xjTTChrkWbx8v6Jn9PAoke5AFeJ4KF0JS4M7pUKvzY8HR+QMAuF2k1VtS8Lo1WJmZPB5M1m5PJyr0hg4Z8hz+Ev+i9TgTG9TEJS7Np2iw+sq0+J6Zxz6VxpVI86XSb1BpqxOoTjcyikU8bmdIJnmdRkL3QhA
8tuloJ5aQxefTXmku2bRYY1FLktKQkQjOUNo8+mvFOq7Y6n0GNwskmFZIU6vRq88x9YRh1wdRoLECQzwIUPHXRrEnhNnkwHrFChSsIyDQMxGijqqbUFVbEKrFk8n5K30oq5I300NoTOhOJxaqRnHfsoofqK
11Yc2fAgLe67Hn1xpCQyUGwq9VQoJ33UOI6IxaY0azI6AYJwPlba4Sbb4/K1URY4uYo38f2Abt4y0101AhcKdama0IBuFPrRU8EWzXrzUmPJyH4RgNFHQyFCye2ljXLl10OrbKXFRRyeMnAtXABcveSvhQP
k3ApTkTrFBYDpoH8zPMonGpkK3U0b1uaQ3dFB3c3CyxCV/Kw3ONtRaUnq/2Liv+dNDgo4+F1/Wn1fs6Y3gGRv9n0U3U1r0MIoXsA3XZ6yJUZU5BgQoXFSpkb0MfKLLd2YZSUWYG5Rwkub05hs/bzSUP9e0+1
KXWBOKKkQALq0w9Y+wEcsUhweAwC93Q+vjUUNwxBy2ILTCaBqac0LITmuJwW62rS0KLH65rUCSFjXoXGV/ICUb5nU9RgRcNUMBSXUa9tl8oB4u1gVoviNIOvjjUcUpww9YXFOWlgHKEBZwbNEreidFz6baUZ
+2SuplTcjqySGhIgpvFXg2wDgYP8kXgAYzzPZg9aNrYNCtvmBWsYMTDXqcdSy9c9bHWI0+X1USnAdIKJ6weKxSj004BolGDp60JX91aEPEPU9/fAqqW6lHdOI17HES2qS+FA/Ur+sDIKhnmZsP5fja6sb+T4
K/WQIhs6RHD2RSZQpulJBekgmTw+QKtw0dVSoyGcxjyWqf0DeXpvXL25stQRDJzIdmXJyJVbuZDyKgVUMKUw9PAFHLnyiVWu1RyVvna838Pb+4YVEFB9ZUKGF2Uv1vvimSwAUDw/YEaK7AqHugEBAq9ORqko
tUJk7RvjgtiS16NtU+AO2H6jUjVWt9GwvQA0BDdIL2FO+aFZp0ZfbDuFfo1t5DBPIQ6dCIS3GEuIDoYdfaQKuwYQtde59WN401eYuHJSqG9T0J4U21sLUdvUD5maCy/B5GJ9etzHIdw70noJKyUvexzhQ2+z5
KxTPuM1vw2W+F35ufbJ25D2Q/FmAZcfW5X1sguKmlgp1FUJNargmoXZ6IgJTwFkUZiIP1ELzXWsfs63gnJ3hzIsyjgPWN6FB39wBg7xM9R1lI5Z2Sygq/7tmcFFLQNtL5MFoodwf/HBQN36HKBNxkv7JcmVH
NXU3Y2FXbqmUH03nkSh0jmngLS7fuBL+vuwTHc2Nj6F5UBvb6TGNbfKpHu4H5uOYYX8DAo1BSq4l4IziZxVugfZSvF1p7U91R2uQTAh0q2/WS6laon6sLFYZmnQpsVxer5xvUHnlM9jPsSGo9mR/zr1o5mGo
ndMytEFEgSqQaX2yxWT8pLDEipRcxbeZdU210KFqjsIQkuVk9kY4st4XO5x1D7G6ZuKtc0NuYiR/123aZkcrw96Der4JLU8us+rTyWx3GULHq478sdPN5H0z0h78qvHaXmX0MD6QX1ylHH3EvqVA6B75aRhF
tzJK0H5Iu6REmgn2m9rIivP4Gt+ktDRk0uuYn8UJTCFUDiTlN6FMJVqk8UtH3L5YiWviN7rI7gvbjprM6UX1Ag==
```

圖 十六 部分私鑰 D_i 的內容

接收方使用最後的解密私鑰 $S_i=x_iD_i$ 將取得的文件解密，如圖十七。

This paper introduces the concept of certicateless public key cryptography (CL-PKC). In contrast to traditional public key cryptographic systems, CL-PKC does not require the use of certicates to guarantee the authenticity of public keys. It does rely on the use of a trusted third party (TTP) who is in possession of a master key. In these respects, CL-PKC is similar to identity-based public key cryptography (ID-PKC). On the other hand, CL-PKC does not suer from the key escrow property that seems to be inherent in ID-PKC. Thus CL-PKC can be seen as a model for the use of public key cryptography that is intermediate between traditional certicated PKC and ID-PKC.
We make concrete the concept of CL-PKC by introducing certicateless public key encryption (CL-PKE), signature and key exchange schemes. We also demonstrate how hierarchical CL-PKC can be supported. The schemes are all derived from pairings on elliptic curves. The lack of certicates and the desire to prove the schemes secure in the presence of an adversary who has access to the master key requires the careful development of new security models. For reasons of brevity, the focus in this paper is on the security of CL-PKE. We prove that our CL-PKE scheme is secure in a fully adaptive adversarial model, provided that an underlying problem closely related to the Bilinear Diffie-Hellman Problem is hard.

圖 十七 使用私鑰 S_i 將文件解密

第六章 結論及未來展望

本論文提出了一個免憑證加密系統的方案，並且將此方案實作出來，使得此方案更具實用性。此方案主要架構為免憑證加密系統，利用此系統的特性消除傳統公開金鑰密碼系統中需要公開金鑰憑證認證的麻煩，也不會產生基於身分認證加密系統的私鑰託管問題，有效的結合了兩項系統的優點，並且提高了這兩種系統的安全性及方便性。本論文的協定中，在基於身分認證加密系統的公鑰部分還加入了階級以及時間戳記的概念，用以限制接收方取得部份私鑰的能力，並且也將接收方的部分公鑰加入其中，來增加部份私鑰的安全性。另外此協定也加入了提早解密金鑰的部分，可讓傳送方在傳出密文後更改解密時間，而不需要重新使用新的公鑰加密資訊，便可提早讓接收方取得相對應的明文資訊。

此系統目前主要針對協定加解密的部分實作，網頁設置的完整性及網頁的美觀上則是我們未來的目標。另外，此協定在未來的應用上，因為此協定加入了階級的概念，因此也可以使用在學校、公司、或是政府機關的人員轉帳部分，或者是醫院的醫護人員取得病例的權限，又或者是公司機密文件的讀取權限上。希望可以在未來將此協定運用在這些部分，讓個人的隱私能安全有效的被保護，減少個人資料外洩的危險。

第七章 參考文獻

[1] W. Diffie, M. E. Hellman, "*New directions in cryptography*", Information Theory 22(6), IEEE Transactions on, pp.644-654 , 1976.

[2] D. Boneh, M. K. Franklin, "*Identity-based encryption from the weil pairing*", CRYPTO 2001, LNCS 2139, pp.213-229, 2001.

[3] S. S. Al-Riyami, K. G. Paterson,"*Certificateless public key cryptography*", ASIACRYPT 2003, Springer-Verlag, LNCS 2894, pp. 452-473,2003.

[4] A. Shamir, "*Identity-based cryptosystems and signature schemes*", CRYPTO 1985, LNCS 196 ,pp.47-53, 1985.

[5] M. Hou, Q. Xu, "*Secure and efficient two-party authenticated key agreement protocol from certificateless public key encryption scheme*", NCM 2009, pp.894-897,2009.

[6] A. Kihidis, K. Chalkias, G. Stephanides, "*Practical implementation of identity based encryption for secure e-mail communication*" Panhellenic Conference on Informatics 2010, pp.101-106,2010.

[7] A. Shamir, "*On the security of DES*". CRYPTO 1985,LNCS 218, pp.280-281,1985.

[8] National Institute of standards and Technology, "*The advanced encryption standard*", http://csrc.nist.gov/aes/ , 2000.

[9] R. L. Rivest, A. Shamir, L. M. Adleman, "*A method for obtaining digital signatures and public-key cryptosystems*", Communications, ACM 21(2), pp.120-126, 1978.

[10] T. ElGamal," *A public-key cryptosystem and a signature scheme based on discrete logarithms*", CRYPTO 1985, LNCS 196, pp.10-18,1985.

[11] Y. H. Hwang, D. H. Yum, P. J. Lee, "*Timed-release encryption with pre-open capability and its application to certified e-mail system*", ISC 2005, LNCS 3650, pp.344-358, 2005.

[12] M. Geng, F. Zhang, M. Gao, "*A secure certificateless authenticated group key agreement protocol*", Multimedia Information Networking and Security 2009, International Conference on , pp. 342−346, 2009.

[13] C. Wang, D. Long, Y. Tang, "*An efficient certificateless signature from pairings*", Data, Privacy, and E-Commerce, 2007, The First International Symposium on, pp.236-238 ,2007.

[14] M. Bellare, P. Rogaway, "*Random oracles are practical: a paradigm for designing efficient protocols*", Computer and Communications Security 1993, ACM Conference on, pp.62-73, 1993.

[15] 詹省三，可訊息回覆之免憑證簽章機制之研究，國立政治大學資訊科學系碩士論文，2011年

[16] 林欣瑤，左瑞麟，關於免憑證密鑰交換機制的一些安全性分析，全國計算機會議，2011年