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

碩士論文

Master's Thesis

可訊息回復之免憑證簽章機制之研究

Certificateless Signatures with Message Recovery

研 究 生：詹省三

指導教授：左瑞麟

中華民國一百年一月

January 2011

可訊息回復之免憑證簽章機制之研究

Certificateless Signatures with Message Recovery

研 究 生：詹省三　　Student：SHENG SAN CHAN

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

中華民國一百年一月

January 2011

可訊息回復之免憑證簽章機制之研究

摘要

在傳統的簽章機制中，我們需要一個具有公信力的第三方 (Trusted Third Party, TTP) 來核發數位憑證，以驗證公開金鑰確實屬於簽章者所擁有，為了減少 TTP 的負擔，於是就有學者提出了免憑證簽章 (Certificateless Signature) 機制。另一方面，具有訊息回復 (Message Recovery) 功能的數位簽章是指原始訊息不需要與簽章一起傳送給接收者以簡化訊息及簽章在傳送時的長度。

本論文中我們提出了一個具有訊息回復功能的免憑證簽章機制，和一般簽章方式相比，我們的方法不僅具有免憑證簽章的優點，訊息回復功能也減少了訊息和簽章的總長度，提昇了訊息的傳送效率 (Communication Cost)，在效能方面也有不錯的表現，因此非常適用於以頻寬為主要考量的公司組織以及對短訊息作簽章的應用，最後我們也有對我們的簽章方法做完整的安全性證明。

Certificateless Signatures with Message Recovery

Abstract

In traditional digital signature systems, a trusted third party (TTP) is required in order to issue a digital certificate. The certificate is to assure that the public key actually belongs to the person of the signature. In order to reduce the burden of TTP, some scholars proposed the Certificateless Signatures. On the other hand, a digital signature with message recovery is a signature that the message itself is not required to be transmitted together with the signature. It has the advantage of small data size of communication.

In this paper, a certificateless signature with message recovery is proposed. It inherits both the advantages of certificateless signatures and signatures providing message recovery. The performance of our scheme is compared with other schemes which shows that our scheme is quite efficient and the security of the scheme is finally proved in the random oracle model.

誌謝

轉眼間兩年多的時間一下就過去了，回首當初剛進政大時面對空蕩蕩的實驗室以及沒有學長姊的窘境還有著些許的不安，幸運的，這些做研究的日子過得還挺順利的，實驗室的氣氛也越來越熱鬧，而我也將離開這熟悉的校園以及熟悉的政大後山-貓空。

首先，我要誠摯的感謝我的指導教授左瑞麟老師，謝謝老師在這段時間悉心的教導，以及不時耐心的與我一起討論並指引我正確的方向，還有老師我一定要跟你說你實在是太陽光了，希望以後還有機會一起去參加比賽，最後也感謝張仁俊老師與陳恭老師，謝謝你們在口試時給了我很多關於論文修改的建議與意見，讓我的論文更加完整與嚴謹。

另外，我還要感謝在這段時間曾給予我鼓勵與打氣的所有人，實驗室的夥伴與學弟妹、資科所的師長與同學、政大單車社的朋友們、大學高中時期的朋友以及在家鄉的好朋友們，謝謝你們在這段時間的支持與包容，沒有你們也不會有現在的我，真的，衷心的謝謝你們。

最後要謝謝我的家人，這些年來雖然和你們相處的日子很少，但我明白你們都還是支持我的，讓我無後顧之憂，我也才能勇敢的繼續往前走。

最後的最後，僅以此篇論文獻給我天上的父親。

省三

目錄

第一章 緒論 1
1.1 研究背景 1
1.2 研究動機與目的 2
1.3 論文架構 4
第二章 背景知識 5
2.1 傳統簽章之簡介 5
2.2 基於身份認證之簽章 8
2.3 免憑證簽章 10
2.4 相關數論介紹 11
第三章 相關研究 15
3.1 Zhang等學者的基於身份認證之可訊息回復簽章 15
3.2 Al-Riyami等學者的免憑證簽章 19
第四章 可訊息回復之免憑證簽章機制 24
4.1 限制訊息長度之免憑證簽章 24
4.2 不限制訊息長度之免憑證簽章 29
第五章 安全性與效能分析 31
5.1 證明方法介紹 31
5.2 安全性模組 34
5.3 安全性證明 39
5.4 效能分析 52
第六章 結論與未來展望 53
參考文獻 54

圖目錄

圖 2.1.1 數位簽章基本架構圖。.....6

圖 2.1.2 憑證簽發流程圖。.....7

圖 2.2 基於身份認證之簽章示意圖。.....9

圖 2.3 免憑證簽章示意圖。.....10

圖 3.1 Zhang 的基於身份認證之可訊息回復簽章架構圖。.....18

圖 3.2 Al-Riyami 等學者的免憑證簽章架構圖。.....23

圖 4.1 提案方式的可訊息回復之免憑證簽章架構圖。.....27

表目錄

表 5.4 效能分析與比較表。.....52

第一章 緒論

1.1 研究背景

密碼學一直以來都是數位通訊、電腦網路、資訊安全等應用上的重要工具，密碼學也促進了電腦科學，特別是在於電腦與網路安全上所使用的技術，例如存取控制與資訊的機密性，日常生活中也隨處可見其相關的應用，例如 ATM 的晶片卡、電腦使用者的存取密碼、電子商務等。

隨著網際網路應用的發達與普及，許多消費者開始使用網路進行交易，於是各企業紛紛投入電子商務活動，在現今的電子商務環境中，商業交易行為必須依賴電子文件與數位簽章來確立其該有的權利與義務，因此數位簽章一直在電子商務中扮演著極為重要的角色，也是現在數位化環境中各種應用的重要基礎。

為了規範在電子商務環境下使用數位簽章所產生的風險，我國電子簽章法就有明文規定[27]，數位簽章須以一定之程序產生及驗證才能和實體之簽名蓋章具有同等之法律效力。根據資訊安全管理的國際標準ISO 17799 定義，數位簽章亦須符合資訊安全中的確認性、完整性以及不可否認性三種安全服務，其分別如下所述:

- 確認性：能確保簽章者即為簽署訊息之本人。
- 完整性：能確保訊息在網路傳送的過程中，不會遭偽造或竄改。
- 不可否認性：防止存心不良的使用者否認其所做過的事，即簽章者不可否認其所簽署過的訊息。

1.2 研究動機與目的

在傳統的簽章機制中，為了驗證公開金鑰確實屬於簽章者所擁有，我們需要一個具有公信力的第三方 (Trusted Third Party, TTP) 來核發數位憑證，藉由驗證此數位憑證來驗證使用者的公開金鑰。但憑證註銷 (Certificate Revocation) 等等的問題會造成 TTP 過多的負擔[29,31]，另外，對簽名的驗證者來說，憑證的驗證也增加了許多的計算成本(Computation Cost)。

在 1984 年的時候，Shamir 提出了第一個基於身份認證的簽章方法 (ID-based Signature) [18]，ID-based signature 的優點是允許簽名者以個人資訊來當作他的公開金鑰，例如 email address、姓名、電話號碼等，如此驗證者就不需要透過憑證去驗證公開金鑰的合法性，也大幅地減少了 TTP 的計算量與記憶體空間。在 ID-based 架構下，由於 TTP 負責提供使用者的私密金鑰，所以在這裡我們稱 TTP 為 PKG (Private Key Generator) ，然而，由於所有使用者的私密金鑰皆是由 PKG 所生成，所以 ID-based signature 可能會有金鑰託管的問題（Key escrow problem）。我們可以說 PKG 是私密金鑰的控管中心，由於私密金鑰皆是由 PKG 所生成， PKG 知道所有使用者的私密金鑰，這樣將使得 PKG 的權限過大，假如今天 PKG 遭受駭客入侵控制，惡意的 PKG 即可偽造所有使用者的簽名，因此使用者就可以否認之前所簽過的訊息，這並不符合數位簽章中的不可否認性，假如使用者人數一多，這會造成相當嚴重的安全性問題。

為了解決 ID-based signature 金鑰託管的問題，在 2003 年的時候，Al-Riyami 等學者提出了免憑證簽章的概念(Certificateless signature) [1]，它同時具有傳統簽章與 ID-based signature 的優點，既可解決金鑰控管的問題，也可保有 ID-based signature 免憑證的特點，其主要的差異在於簽名者的私密金鑰並不是完全由 PKG 所生成，所以 PKG 無法得知簽名者的私密金鑰。

一般的簽章演算法中明文是直接暴露的，當簽名者對訊息 m 簽名產生簽章 s，簽名者會將訊息 m 以及簽章 s 一起傳送給驗證者，這是因為驗證者必須要有訊息 m 才可以去驗證簽章 s 是否合法。因此，具有訊息回復功能的數位簽章是指原始訊息不需要與簽章一起傳送給驗證者，且驗證者可在驗證階段利用一些公開參數與簽章即可回復原始的訊息，這類型的簽章其目的就是為了能簡化簽章在傳送時的長度。

在 1993 年的時候，Nyberg 等學者提出了第一個具有訊息回復之數位簽章[16]，此方法是基於離散對數問題(Discrete Logarithm Problem)，自此以後，關於訊息回復這方面的簽章研究也陸續地被提出。在 2005 年的時候，Zhang 等學者提出了第一個具有訊息回復之 ID-based signature[25]，基於 Zhang 等學者以及 Barreto 等學者[5]的概念，在 2007 年的時候，Tso 等學者提出了更有效率的可訊息回復之 ID-based signature[21]，而我們的方法同樣也是基於 Zhang 等學者的概念下去做延伸。因此，本論文的研究動機就是為了解決 ID-based signature 架構下本身存在的問題，我們改良了 Zhang 等學者的方法，提出了基於免憑證架構下可訊息回復的簽章方法。

1.3 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與研究動機做一簡單、完整的說明。
- 第二章我們將對傳統數位簽章、基於身份認證之簽章、免憑證簽章做完整的說明，同時對於後續需使用到的雙線性配對函數、密碼學上的計算難題以及相關數學知識與定義做一簡單、完整的介紹。
- 第三章我們將對過去兩篇相關的文獻做簡要的介紹。
- 第四章為介紹我們所提出的可訊息回復之免憑證簽章協定。
- 第五章將介紹我們使用到的證明方法，並針對我們所提出的協定做安全性證明與效率分析，並與過去相關的方法做一比較。
- 第六章為結論與未來展望。

第二章 背景知識

2.1 傳統簽章之簡介

在 1976 年的時候，Diffie 和 Hellman 二位學者提出了公開金鑰密碼系統(Public-key cryptosystem)的概念[8]，他們的研究在密碼學發展上是一個重要的里程碑，之後各式各樣適合各種不同應用環境的簽章方法也先後的被發展出來。

數位簽章的架構如下圖 2.1.1 所示，在公開金鑰密碼系統架構下使用者擁有兩把金鑰，私密金鑰與公開金鑰，此兩把金鑰是相對應的，私密金鑰是用來簽署文件的，必須隱密的收藏著，只有使用者本人知道，而另一把公開金鑰顧名思義即是在系統內的每個使用者皆可得到，是用來驗證文件的，簽名過程如下圖，簽名者 Alice 是用其私密金鑰對訊息做簽章的動作，而驗證方的使用者則用簽名者 Alice 的公開金鑰去對訊息做驗證，即可驗證該訊息是否由簽名者 Alice 所簽署。

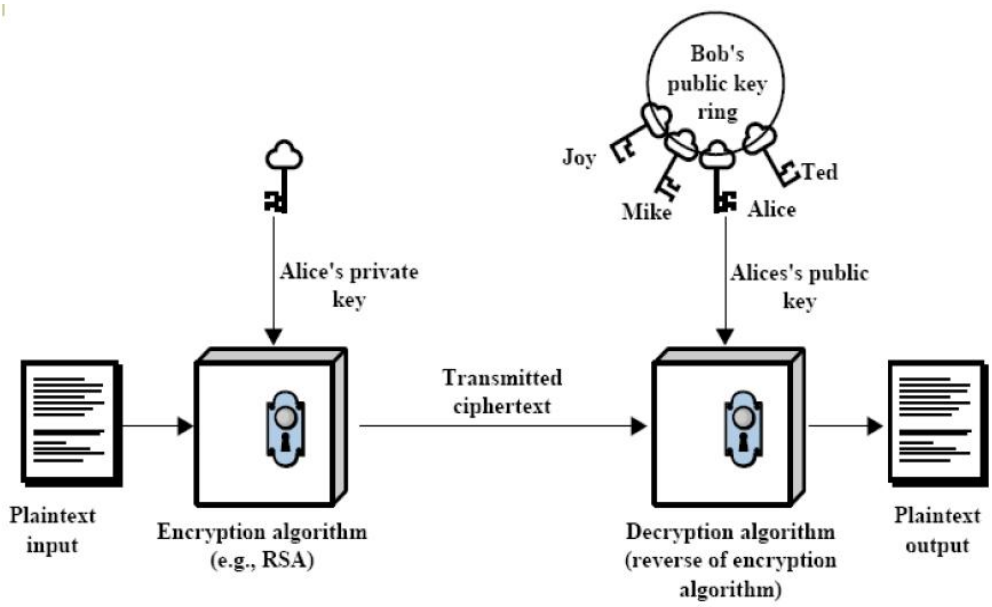

圖 2.1.1 數位簽章基本架構圖。

由於驗證方沒辦法證實公開金鑰的正確來源性，所以公開金鑰基礎架構(Public Key Infrastructure, PKI)就是為了解決這個問題應運而生，其主要功能就是為每位使用者的公開金鑰做背書，以證明該公開金鑰的持有者確實是由所宣稱的那個人所擁有，也因為有此特性，PKI 能有效的解決網路上進行交易或訊息傳遞時所衍生的身份辨識、資料的確認性、完整性以及不可否認性等安全問題。

如圖 2.1.2 所示，在 PKI 架構中所有公開金鑰都被集中保管於公正的第三方，也就是憑證機構(Certificate Authority, CA)，CA 驗證簽名者的身份後會核發一個憑證給簽名者，憑證內容就包括簽名者的公開金鑰與其身份識別，任何要驗證之使用者都可透過 CA 的公開金鑰向 CA 查詢交易相對人的公開金鑰及身份，因此可以確保公開金鑰的來源性。

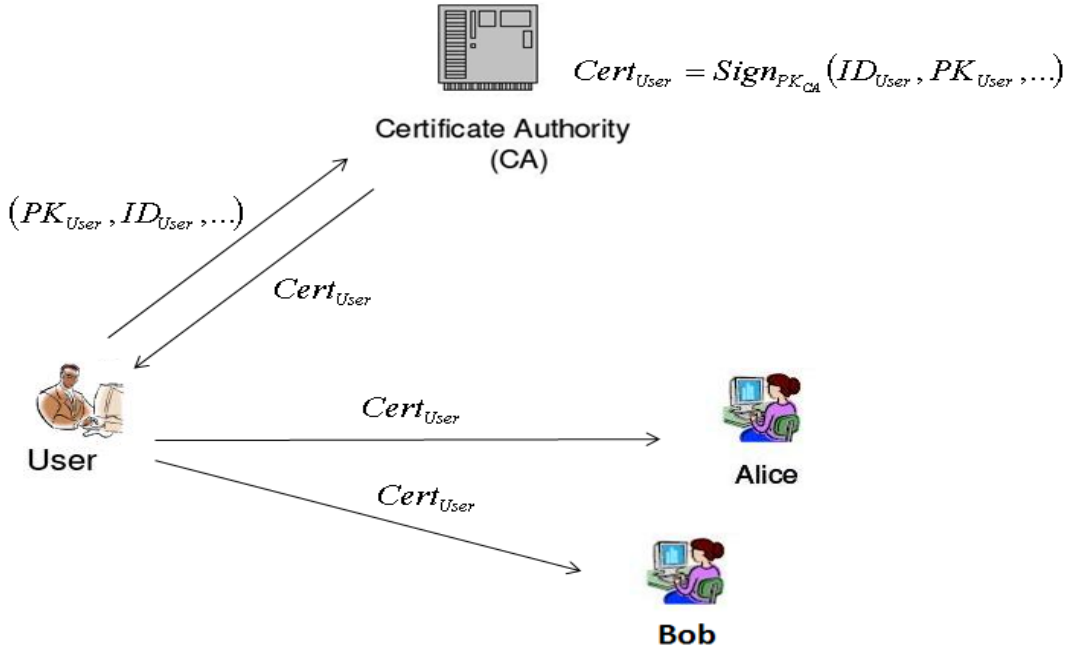

圖 2.1.2 憑證簽發流程圖。

由於憑證註銷(Certificate Revocation)等等的問題會造成 CA 過多的負擔，當使用者因某種原因更改其金鑰、使用者不再使用 CA 所提供的認證服務、使用者信用不好被 CA 列為拒絕往來戶、CA 自身遭遇危機或發生失誤時，就會需要註銷使用者的憑證，憑證註銷的目的就是為了避免數位憑證被盜用或非法使用，通常 CA 會利用憑證註銷清冊(Certificate Revocation List, CRL)來紀錄所有被註銷但尚未到期的憑證，CA 再將 CRL 傳送給所有驗證單位或使用者，當使用者要驗證憑證時只需檢查 CRL 即可知道該憑證是否合法，但也有可能會增加許多的計算成本(Computation Cost)，若是憑證註銷的使用者過多時，將會使 CRL 快速膨脹，因為註銷的即時性問題，驗證方每次要驗證憑證時都需要下載 CRL，會使整個驗證的過程效率降低，增加了 CA 與驗證單位的通訊量以及 CRL 的維護量，目前面對此問題大多以定期下載 CRL 使用為解決方法，但當前的 CRL 與最新的 CRL 間還是會有時間落差而不一致的問題，而為了解決時間差的問題，有很多學者也在這方面做了許多相關的研究[28,30]。

2.2 基於身份認證之簽章

在 1984 年的時候，Shamir 提出了第一個基於身份認證的簽章方法 (ID-based Signature) [18]，在傳統簽章下使用者不能自己決定持有的公開金鑰且公開金鑰都是系統隨機產生的亂數，此簽章與傳統簽章最大的不同就是使用者是以其「唯一」且「可識別」的身份識別子來當作其公開金鑰，例如電子信箱(e-mail),電話(phone number), 地址(address)等等資訊，因此不會有傳統簽章需要驗證公開金鑰來源性的問題，使用者也不需要保留公開金鑰的目錄，以及不需有公信力的第三方 (Trusted Third Party, TTP)的協助即可祕密地通訊和驗證對方的簽章。在基於身份認證的簽章架構下，每個使用者的私密金鑰皆是由 TTP 或也可稱作私密金鑰產生者(Private key generator, PKG)的公正第三方來生成，由於公開金鑰即為簽名者「唯一」且「可識別」的身份識別子，因此 PKG 不需要核發憑證給簽名者,每個接收簽章的驗證方皆可直接利用簽名者的公開金鑰來驗證簽章。在此架構下因為沒有憑證核發註銷等動作，因此也不會有在 PKI 架構下造成 TTP 過多負擔的問題，也省去了驗證憑證時所花費的通訊與計算成本。

如圖 2.2 所示，Alice 為此系統的簽名者，Bob 則是驗證簽名的接收者，首先 PKG 利用 Alice 傳送過來的 ID_A 去計算其私密金鑰 SK_A 並回傳給 Alice，Alice 再利用 SK_A 對訊息 M 做簽名來得到簽章 σ，接著把簽章 σ 與訊息 M 傳送給 Bob，Bob 再利用 Alice 的公開金鑰 ID_A 去做驗證的動作，這就是 ID-based Signature 的基本架構。然而，ID-based Signature 存在一個不安全的因子，由於 PKG 知道所有簽名者的私密金鑰，PKG 可以假冒所有的簽名者去對任意訊息做簽章的動作，因此簽名者可以否認之前所簽過的訊息，所以在此架構下 PKG 是必須完全被信賴的，這就是 ID-based Signature 本身存在的金鑰託管（Key escrow）問題。

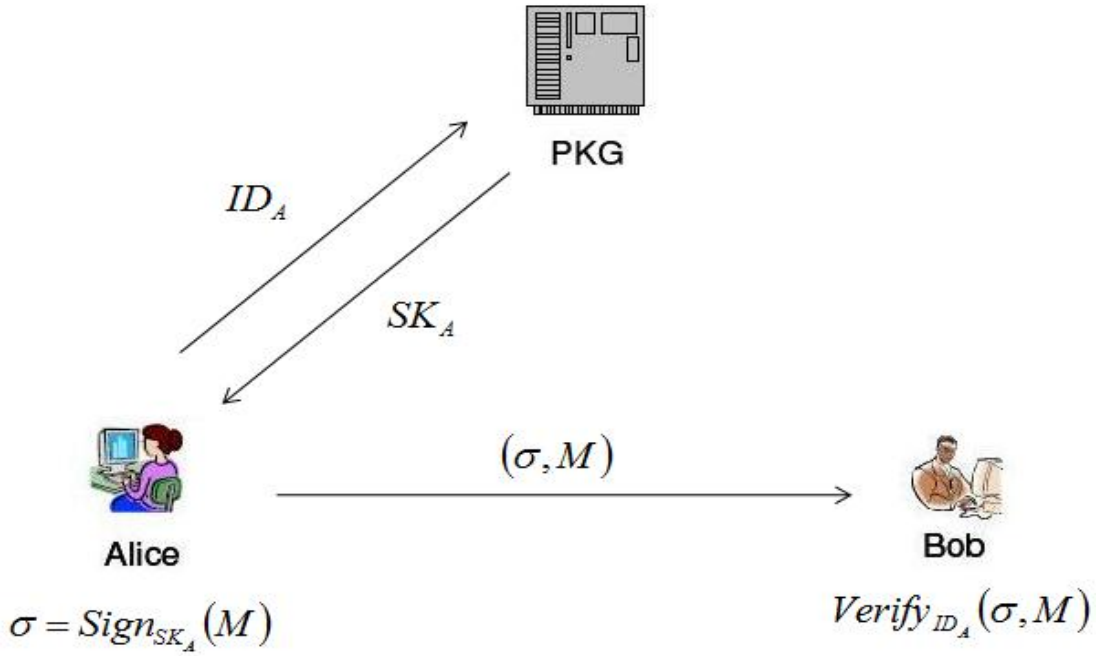

圖 2.2 基於身份認證之簽章示意圖。

2.3 免憑證簽章

在 2003 年的時候，Al-Riyami 等學者提出了免憑證簽章(Certificateless signature)的概念[1]，免憑證簽章與 ID-based Signature 最大的不同就是使用者有兩對金鑰，一對金鑰是使用者自己生成的私密金鑰與公開金鑰，另一對則是 PKG 生成的部份私密金鑰(Partial private key)與部份公開金鑰(Partial public key)，簽名的時候使用者是用到兩把私密金鑰，驗證的時候也是用到另外兩把公開金鑰才可驗證，這樣的好處就 PKG 沒辦法只利用部份私密金鑰去偽造簽章，因此免憑證簽章不會有金鑰控管的問題。

如圖 2.3 所示，Alice 同樣為此系統的簽名者，Bob 則是驗證簽名的接收者，首先 PKG 利用 Alice 傳送過來的 ID_A 去計算其部份私密金鑰 PSK_A 並回傳給 Alice，接著 Alice 隨機選取一亂數 s 當作自己的私密金鑰 SK_A，Alice 再利用 SK_A 與 PSK_A 對訊息 M 做簽名來得到簽章 σ，接著把簽章 σ 與訊息 M 傳送給 Bob，Bob 再利用 Alice 的公開金鑰 PK_A 與 ID_A 去做驗證的動作，這就是免憑證簽章的基本架構。

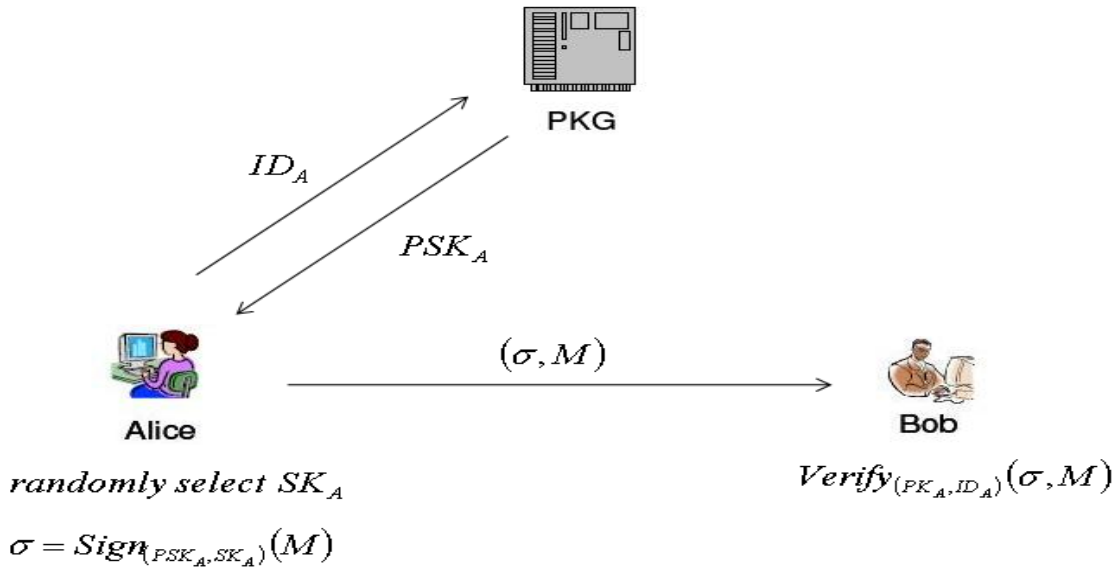

圖 2.3 免憑證簽章示意圖。

2.4 相關數論介紹

密碼學技術能有效保護資訊在網路上安全的傳送訊息，不管是金鑰協議、加解密機制或數位簽章機制，都需要用到密碼學的技術，以下我們將會對與本篇論文相關的密碼學理論與背景知識做相關介紹。

2.4.1 離散對數問題(Discrete Logarithm Problem)

密碼系統的安全性大多是基於計算複雜度的難題上，主要是基於數學理論上因數分解與離散對數問題，所謂的離散對數問題如下所定義[10]：

定理 1：

若 p 為很大之質數； g 為 p 之生成元 (generator)，方程式為 $y = g^x \pmod{p}$，今已知 y, g, p，需求出 x 滿足 $g^x = y \pmod{p}$，此一問題即為離散對數問題，一般相信是困難的。

2.4.2 橢圓曲線密碼學(Elliptic Curve Cryptography, ECC)

橢圓曲線原本是數學理論上的一個主題，但 1985 年及 1987 年有兩位學者 Miller[15] 和 Koblitz[14]分別將橢圓曲線應用到密碼學上，因為橢圓曲線具有較小金鑰長度即可達到與 RSA 相當的安全程度，以及橢圓曲線可以建構出雙線性配對等特點，所以近年來橢圓曲線在密碼學上相關的研究也越來越多。

橢圓曲線其實就是三次方的代數曲線，一般可表示成：

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

若 F_q 特徵值不為 2 或 3 時，我們就能將(1)轉換成較簡單的表示法

$$E: y^2 = x^3 + Ax + B \quad (2)$$

若滿足 $4A^3 + 27B^2 \neq 0$，則可依據切線與割線規則訂定點加法，使得 $E(F_q)$ 形成加法交換群，而詳細的加法運算規則可參閱[20]。

ECC 的主要優勢是在某些情況下金鑰長度比其他方法的要來的小，例如 RSA，目前所知破解 160 bit 的 ECC 與破解 1024 bit 的 RSA 所需要的時間大約相同，所以在相同的安全程度下 ECC 的金鑰長度比 RSA 小，而 ECC 的另一個優勢就是可以定義群之間的雙線性配對，例如 Weil pairing 或是 Tate pairing。直到 2000 年，Pairing 在密碼學上的正向應用才正式出現[13]，而更全面的應用方式則是 Boneh 等學者在 2001 年所提出，以 Weil pairing 建構出基於身份識別碼的加密系統[4]，自此 pairing 在密碼學上的地位就由此確立，也為密碼學指引了一個新的研究方向，此後各類密碼學應用的研究也都看得到 pairing 的蹤影[2, 6,12,19,22]。

2.4.3 雙線性配對 (Bilinear Pairing)

雙線性配對函數為一線性映射函數(Bilinear map)，由一個群(Group)對應到另一個群，以雙線性配對為基礎的密碼機制其安全性是建立在解一些難問題假設的困難度上，以下將會介紹雙線性配對函數的定義以及一些相關的難問題。

G_1 為一加法群 (Additive Group)，序(Order)為 q ，G_2 為一乘法群 (Multiplicative Group)，序也為 q。P 是 G_1 的生成元(Generator)，則一個雙線性配對表示為 $e: G_1 \times G_1 \rightarrow G_2$，具有以下三種性質[6,24]：

(1) 雙線性(Bilinear)：$P, Q \in G_1$ 及 $a, b \in Z_q^*$，$e(aP, bQ) = e(P, Q)^{ab}$。

(2) 非退化性(Non-degenerate)：若 P 是 G_1 的生成元，則 $e(P, P)$ 也會是 G_2 的生成元，即 $e(P, P) \neq 1$。

(3) 可計算性(Computable)：$P, Q \in G_1$，存在一有效率的演算法可計算 $e(P, Q)$。

在密碼學的研究領域，為符合系統安全的需求，通常會有許多計算難問題的假設，以下即是相關研究中常見的難問題定義[3,26]：

定理 2：

計算性 Diffie-Hellman 難問題(Computational Diffie-Hellman Problem, CDHP)，如果 $P \in G_1$ 且 a、$b \in Z_q^*$，在已知 P 、aP 、以及 bP 的條件下，要求解 abP 的問題即為 CDH 難問題。

定理 3：

決定性 Diffie-Hellman 難問題(Decisional Diffie-Hellman problem, DDHP)，如果 $P \in G_1$ 且 a、b、$c \in Z_q^*$，在給定 P 、aP 、bP 以及 cP 的條件下，要判斷是否 $c = ab$ 的問題即為 DDH 難問題。

定理 4：

間隙 Diffie-Hellman 難問題(Gap Diffie-Hellman problem, GDHP)，如果 DDH 問題為容易而 CDH 問題為困難者稱之為 GDH 難問題。

定理 5：

雙線性 Diffie-Hellman 問題(Bilinear Diffie-Hellman Problem,BDHP)，如果 a、b、$c \in Z_q^*$ 為未知數，給定 $P, aP, bP, cP \in G_1$，要求得 $e(P,Q)^{abc} \in G_2$ 是非常難解的。

2.4.4 符號標記

根據 Tso 等學者[21]提出的可訊息回復之 ID-based signature 的定義，本篇論文中同樣也會使用到相同的符號，其定義如下：

- $a \| b$：a 字串與 b 字串結合的連續字串。
- $\oplus$：二進位系統中的 X-OR 運算。
- $[x]_{10}$：x 的十進位表示法且 $x \in \{0,1\}^*$。
- $[y]_2$：y 的二進位表示法且 $y \in Z$。
- $|q|$：q 的位元長度。
- ${}_{l_1}|\beta|$：從 β 左側開始算起的 l_1 位元，亦即，最高有效 l_1 位元。
- $|\beta|_{l_2}$：從 β 右側開始算起的 l_2 位元，亦即，最低有效 l_2 位元。

第三章 相關研究

3.1 Zhang等學者的基於身份認證之可訊息回復簽章

此小節我們將介紹 Zhang 等學者提出的基於身份認證之可訊息回復簽章[25]，這是第一個具有訊息回復功能之 ID-based signature，由於我們的方法是基於此方法的延伸，故在此介紹他們的簽章方法。

3.1.1 簽章模組

根據 Zhang[25]的定義，Identity-based message recovery signatures 的簽章模組大致可以定義成四個演算法，分別如下所述：

(1) Setup

此階段 PKG 會生成一對自己的金鑰，且會公開一些系統參數。

(2) Extract

此階段 PKG 會根據使用者的身份識別 ID，生成使用者的私密金鑰並將該私密金鑰傳送給使用者。

(3) Sign

此階段使用者會利用私密金鑰與公開的系統參數去對訊息做簽章。

(4) Verify

此階段使用者可利用公開參數並透過一決定型的演算法去驗證簽章的合法性。

3.1.2 簽章方法

[Setup]

(1) 首先 PKG 隨機選取一亂數 $s \in Z_q^*$ 為 PKG 的私密金鑰，接著計算 $P_{pub}=sP$ 為 PKG 的公開金鑰。

(2) PKG 公開系統參數給使用者：

$<G_1, G_2, e, q, \lambda, P, H_0, H_1, F_1, F_2, k_1, k_2>$

且系統參數的定義分別如下：

- G_1, G_2 皆為相同序 q 的循環群，且 $|q| = l_1 + l_2$
- $e: G_1 \times G_1 \to G_2$ 為雙線性配對
- $H_0: \{0,1\}^* \to G_1^*$，輸入為 $\{0,1\}^*$ 的字串，輸出為 G_1^* 中的元素的單向雜湊函數 (one way hash function)
- $H_1: \{0,1\}^* \to Z_q^*$，輸入為 $\{0,1\}^*$ 的字串，輸出為 Z_q^* 中的元素的單向雜湊函數
- $F_1: \{0,1\}^{k_2} \to \{0,1\}^{k_1}$，輸入為 $\{0,1\}^{k_2}$ 的字串，輸出為 $\{0,1\}^{k_1}$ 的字串的單向雜湊函數

- F_2：$\{0,1\}^{k_1} \to \{0,1\}^{k_2}$，輸入為$\{0,1\}^{k_1}$的字串，輸出為$\{0,1\}^{k_2}$的字串的單向雜湊函數

[Extract]

(1) PKG 根據簽名者 A 的身份識別ID_A計算$Q_A = H_0(ID_A)$為簽名者的公開金鑰。

(2) PKG 計算$S_A = sQ_A$為簽名者的私密金鑰，並將私密金鑰回傳送給簽名者。

[Sign]

針對一欲簽名的訊息$m \in \{0,1\}^{k_2}$，簽名者 A 會執行以下步驟來對 m 做簽名：

(1) 隨機挑選一亂數$k \in Z_q^*$，接著計算$v = e(P,P)^k$。

(2) 計算$f = F_1(m) \| (F_2(F_1(m)) \oplus m)$。

(3) 定義$r = H_1(v) + f \bmod q$。

(4) 計算$U = kP - rS_A$。

最後得到對 m 的簽名為$\sigma = (r, U)$。

[Verify]

針對簽名σ，使用者可依以下步驟驗證σ是否合法：

(1) 使用者根據簽名$\sigma = (r,U)$及公開的參數可計算出：

$r - H_1(e(U,P)e(Q_A, P_{pub})^r) = f$。

(2) 回復訊息$m \leftarrow |f|_{k_2} \oplus F_2(_{k_1}|f|)$。

若 $_{k_1}|f| = F_1(m)$ 則表示此簽章 σ 為合法簽章。

關於此簽章的架構圖，如下圖 3.1 所示：

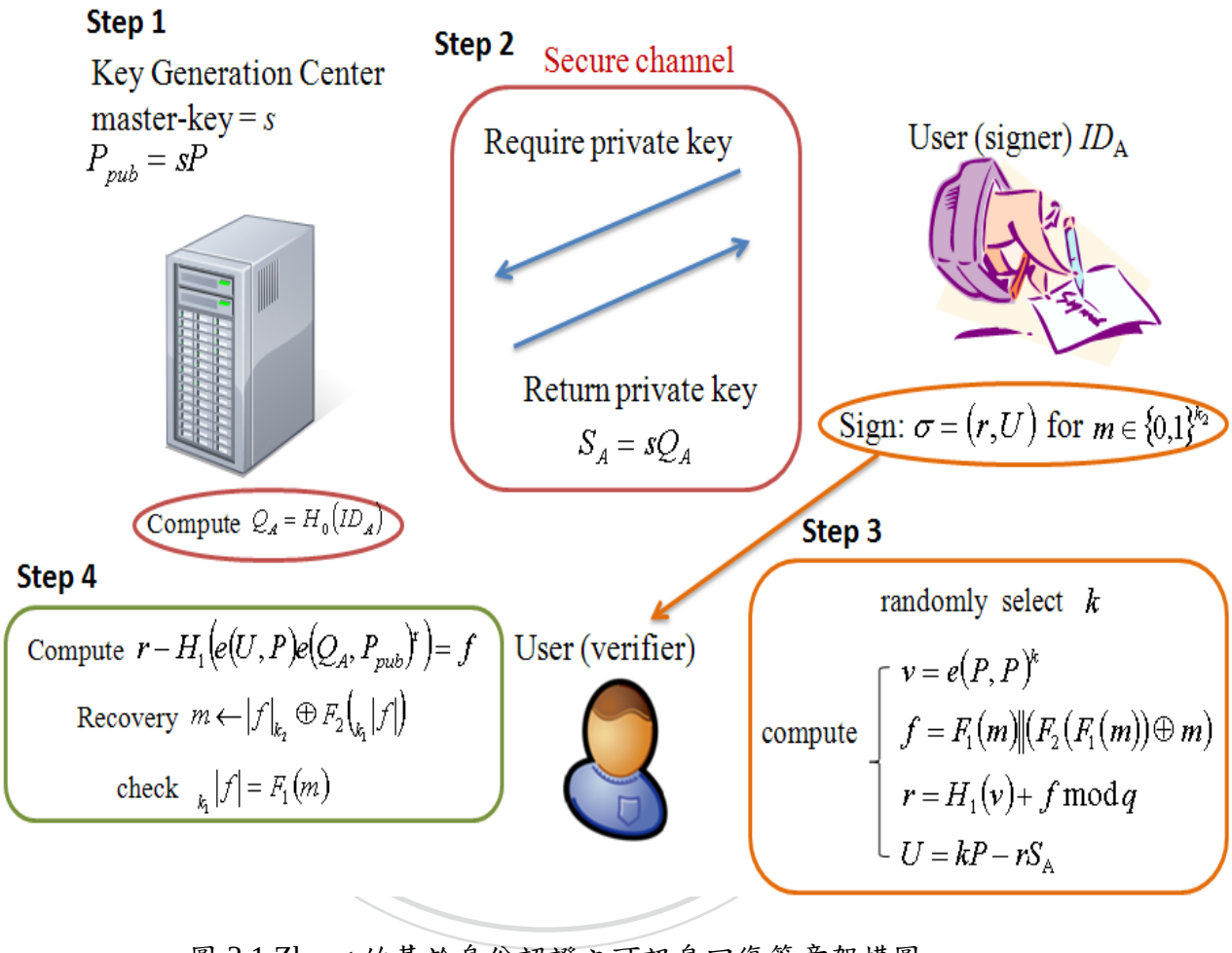

圖 3.1 Zhang 的基於身份認證之可訊息回復簽章架構圖。

由以上演算法可知，簽章者只需傳送 σ 給驗證者，驗證者可利用 σ 與系統公開參數來計算出 f，之後驗證者即可利用 f 來回復訊息 m 以及驗證簽章的合法性。此方法在 CDH 難問題假設上有被證明其安全性，但因為該簽章方法是建立在 ID-based 架構下，所以還是可能會發生金鑰控管的問題。

3.2 Al-Riyami等學者的免憑證簽章

此小節我們將介紹 Al-Riyami 等學者[1]提出的免憑證簽章演算法，這也是第一個免發送憑證的數位簽章，由於我們的免憑證簽章架構是基於此篇方法，故在此介紹他們的簽章方法。

3.2.1 簽章模組

根據 Al-Riyami 等學者[1]的定義，免憑證簽章大致可以定義成七個演算法，分別如下所述：

(1) Setup

此階段 PKG 會生成一對自己的金鑰，且會公開一些系統參數。

(2) Partial-Private-Key-Extract

此階段 PKG 會根據使用者的身份識別 ID 生成部份的私密金鑰與公開金鑰並將部份私密金鑰傳送給使用者。

(3) Set-Secret-Value

此階段使用者會隨機選取一秘密參數。

(4) Set-Private-Key

此階段使用者會利用秘密參數做一些運算，得到的結果為使用者的私密金鑰。

(5) Set-Public-Key

此階段使用者會利用秘密參數做一些運算，得到的結果為使用者的公開金鑰。

(6) Sign

此階段使用者會利用私密金鑰與公開的系統參數去對訊息做簽章。

(7) Verify

此階段使用者可透過一決定型的演算法去驗證簽章的合法性。

3.2.2 簽章方法

以下我們將介紹 Al-Riyami 等學者[1]提出的免憑證簽章演算法，如下所述：

[Setup]

(1) 首先 PKG 隨機選取一亂數 $s \in Z_q^*$ 為 PKG 的私密金鑰，接著計算 $P_0 = sP$ 為 PKG 的公開金鑰。

(2) PKG 公開系統參數給使用者：

$\langle G_1, G_2, n, e, P, P_0, H \rangle$

且系統參數的定義分別如下：

- G_1, G_2 皆為相同序 q 的循環群
- $e: G_1 \times G_1 \to G_2$ 為雙線性配對

- $H:\{0,1\}^* \times G_2 \to Z_q^*$，$H$ 為一單向雜湊函數，其輸入為 $\{0,1\}^*$ 的字串以及 G_2 中的元素，輸出為 Z_q^* 中的元素

[Partial-Private-Key-Extract]

(1) PKG 根據簽名者 A 的身份識別 ID_A 計算 $Q_A = H(ID_A) \in G_1^*$ 為簽名者 A 的部份公開金鑰(ID-based public key)。

(2) PKG 計算 $D_A = sQ_A \in G_1^*$ 為簽名者 A 的部份私密金鑰，並將部份私密金鑰傳送給簽名者 A。

[Set-Secret-Value]

簽名者 A 隨機選取一亂數 $x_A \in Z_q^*$ 為 Secret-Value。

[Set-Private-Key]

簽名者 A 計算 $S_A = x_A D_A = x_A s Q_A \in G_1^*$ 為自己的私密金鑰。

[Set-Public-Key]

簽名者 A 計算 $P_A = \langle X_A, Y_A \rangle$ 為自己的公開金鑰，其中 $X_A = x_A P, Y_A = x_A P_0 = x_A sP$。

[Sign]

針對一欲簽名的訊息 $M \in \{0,1\}^n$，簽名者 A 會執行以下步驟來對 M 做簽名：

(1) 隨機選取一亂數 $a \in Z_q^*$。

(2) 計算 $r = e(aP, P) \in G_2$。

(3) 設置 $v = H(M, r) \in Z_q^*$。

(4) 計算 $U = vS_A + aP \in G_1$。

最後得到對 M 的簽名為 (U, v)，並將 M 及 (U, v) 傳送給驗證方。

[Verify]

針對簽名 (U, v)，驗證方的使用者可依以下步驟驗證簽章是否合法：

(1) 使用者先確認 $e(X_A, P_0) = e(Y_A, P)$ 等式是否成立，如果不成立就輸出 $\perp$ 並終止驗證程序。

(2) 計算 $r = e(U, P) \cdot e(Q_A, -Y_A)^v$。

(3) 若 $v = H(M, r)$ 成立，則輸出 valid，表示此簽章是合法簽章，否則輸出 invalid。

關於此簽章的架構圖，如下圖 3.2 所示：

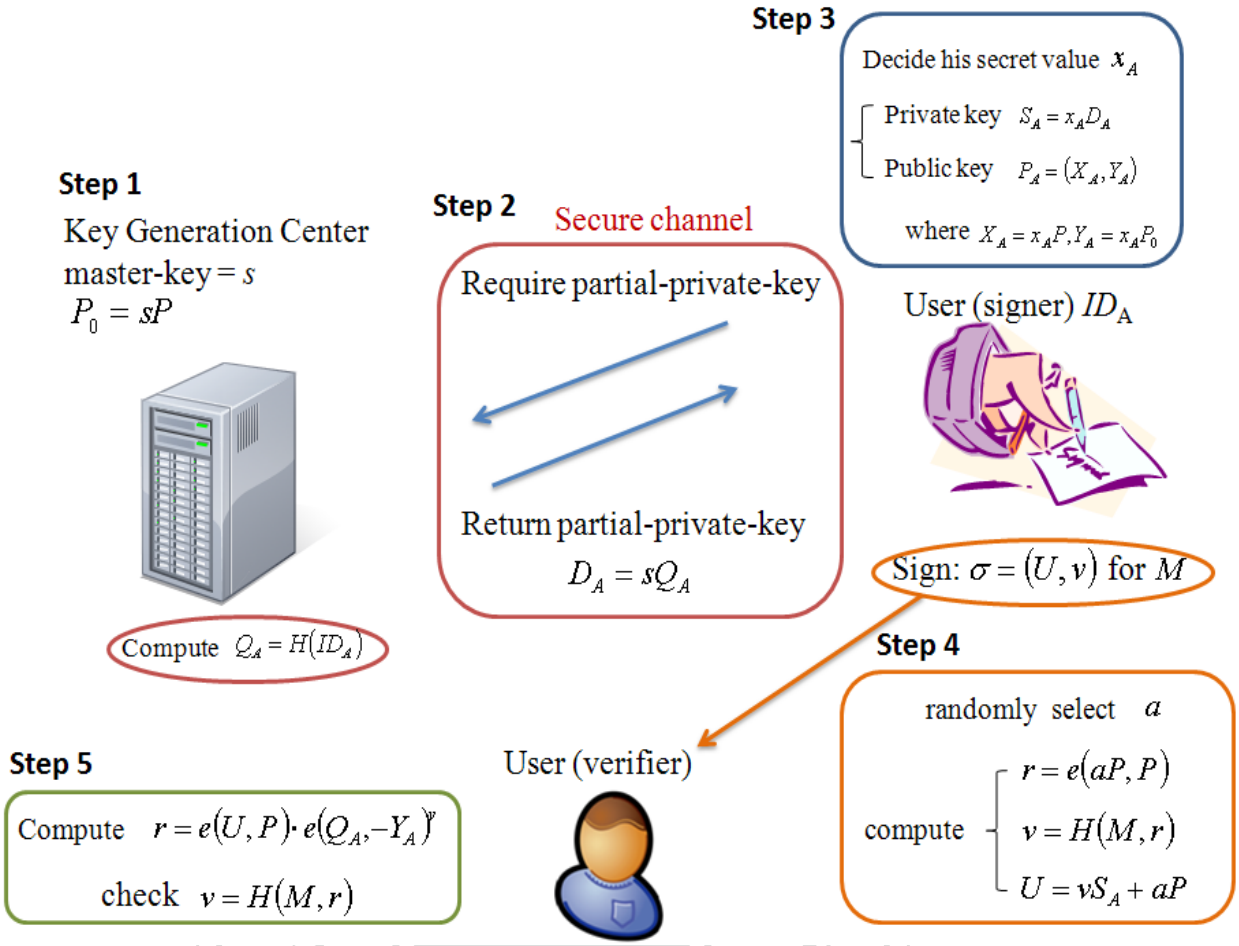

圖 3.2 Al-Riyami 等學者的免憑證簽章架構圖。

第四章 可訊息回復之免憑證簽章機制

根據 Zhang 等學者提出的基於身份認證之可訊息回復簽章[25]，為了解決金鑰控管問題，我們將會以 3.2 小節所提到之免憑證簽章架構去改良 Zhang 等學者提出的方法。在此章節我們將會提出可訊息回復之免憑證簽章的方法，關於訊息回復在我們的方法中我們分兩點討論，第一點就是訊息長度有限制，但可以回復完整的訊息，例如，$m \in \{0,1\}^{l_1}$ 即 m 的長度限制為 l_1，第二點則是訊息長度不限制，但只能回復部份的訊息，以下我們會個別討論。

4.1 限制訊息長度之免憑證簽章

與 Al-Riyami 等學者[1]的免憑證簽章架構一樣，我們需要七個演算法來達到本簽章系統的設計及目的，其分述如下：

[Setup]

(1) 首先 PKG 隨機選取一亂數 $s \in Z_q^*$ 為 PKG 的私密金鑰，接著計算 $P_{pub} = sP$ 為 PKG 的公開金鑰。

(2) PKG 公開系統參數給使用者：

$\langle G_1, G_2, q, e, P, P_{pub}, \mu, H_1, H_2, F_1, F_2, l_1, l_2 \rangle$

且系統參數的定義分別如下：

- G_1, G_2 皆為相同序 q 的循環群，且 $|q| = l_1 + l_2$
- $e: G_1 \times G_1 \to G_2$ 為雙線性配對
- $\mu = e(P, \mathrm{P})$
- $H_1: \{0,1\}^* \to G_1$，輸入為 $\{0,1\}^*$ 的字串，輸出為 G_1 中的元素的單向雜湊函數 (one way hash function)
- $H_2: \{0,1\}^* \to Z_q^*$，輸入為 $\{0,1\}^*$ 的字串，輸出為 Z_q^* 中的元素的單向雜湊函數
- $F_1: \{0,1\}^{l_1} \to \{0,1\}^{l_2}$，輸入為 $\{0,1\}^{l_1}$ 的字串，輸出為 $\{0,1\}^{l_2}$ 的字串的單向雜湊函數
- $F_2: \{0,1\}^{l_2} \to \{0,1\}^{l_1}$，輸入為 $\{0,1\}^{l_2}$ 的字串，輸出為 $\{0,1\}^{l_1}$ 的字串的單向雜湊函數

[Partial-Private-Key-Extract]

(1) PKG 根據簽名者 A 的身份識別 ID_A 計算 $\mathrm{Q}_A = H_1(ID_A)$ 為簽名者 A 的部份公開金鑰。

(2) PKG 計算 $D_A = sQ_A$ 為簽名者 A 的部份私密金鑰，並將部份私密金鑰傳送給簽名者 A。

[Set-Secret-Value]

簽名者 A 隨機選取一亂數 $x_A \in Z_q^*$ 為 Secret-Value。

[Set-Private-Key]

簽名者 A 計算 $S_A = x_A Q_A$ 為自己的私密金鑰。

[Set-Public-Key]

簽名者 A 計算 $PK_A = (PK_1, PK_2) = (x_A P, x_A P_{pub})$ 為自己的公開金鑰。

[Sign]

針對一欲簽名的訊息 $m \in \{0,1\}^{l_1}$，簽名者 A 會執行以下步驟來對 m 做簽名：

(1) 計算 $\beta = F_1(m) \| (F_2(F_1(m)) \oplus m)$，接著定義 $\alpha = [\beta]_{10}$。

(2) 隨機挑選一亂數 $r \in Z_q^*$，接著計算 $V = H_2(\mu^r \| PK_2) + \alpha$。

(3) 計算 $U = rP + V(D_A + S_A)$。

最後得到對 m 的簽名為 $\sigma = (U, V)$。

[Verify]

當驗證方的使用者收到系統公開參數、σ、以及簽名者的 ID 與 PK_A 後，需先根據 $e(PK_1, P_{pub}) = e(P, PK_2)$ 等式來驗證 PK_A 的合法性。

若 PK_A 是合法的，使用者再依以下步驟驗證 σ 是否合法：

(1) 使用者根據簽名 $\sigma = (U, V)$ 及公開的參數可計算出：

$\alpha = V - H_2(e(U, P) \cdot e(Q_A, PK_1 + P_{pub})^{-v} \| PK_2)$。

(2) 接著可推算出 $\beta = [\alpha]_2$。

(3) 回復訊息 $m' = F_2(_{l_2}|\beta|) \oplus |\beta|_{l_1}$。

(4) 若 $_{l_2}|\beta| = F_1(m')$ 則表示此簽章 σ 為合法簽章。

關於此簽章的架構圖，如下圖 4.1 所示：

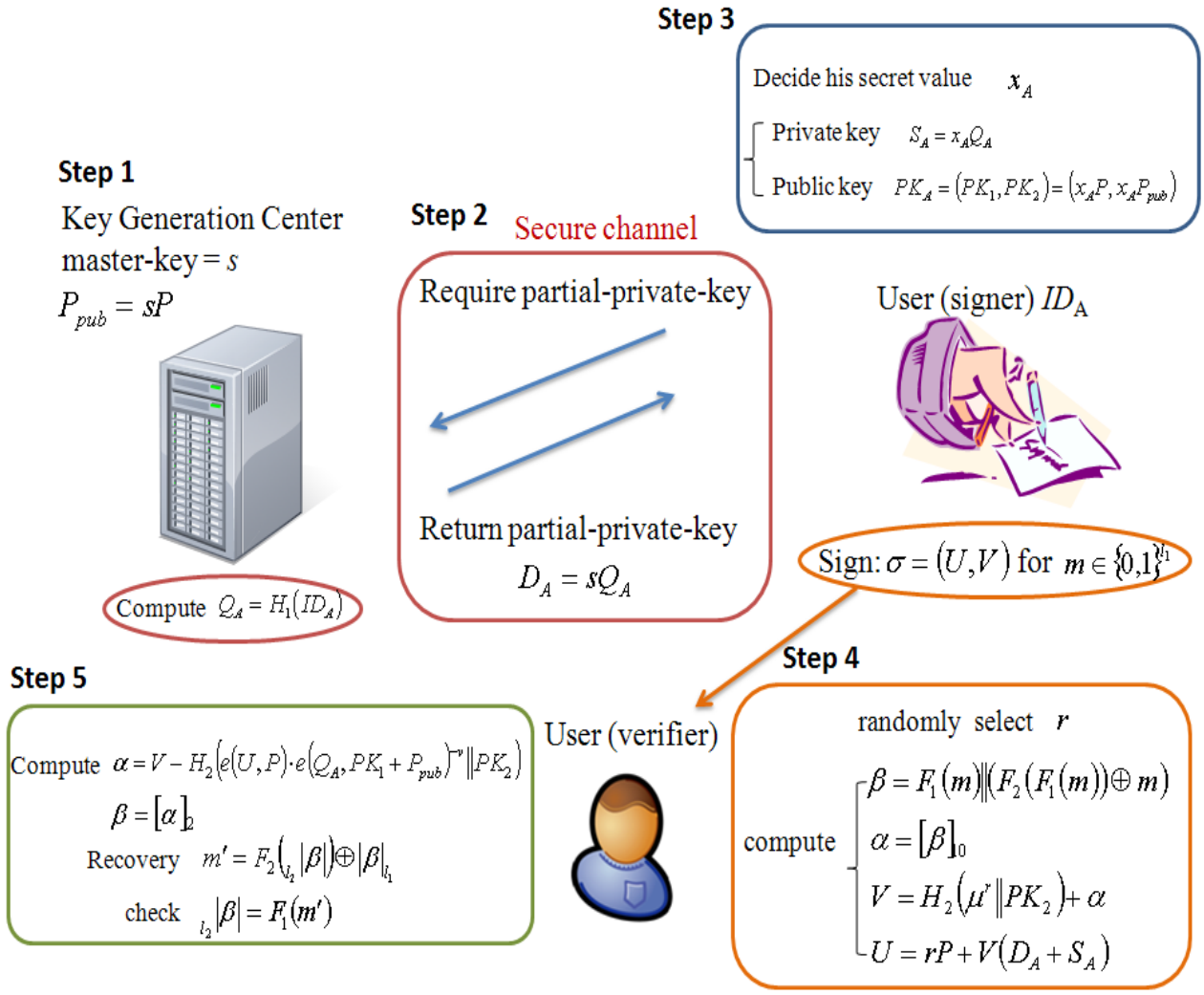

圖 4.1 提案方式的可訊息回復之免憑證簽章架構圖。

此方法的正確性可被證明如下：

因為 $\alpha = V - H_2\left(e(U,P)\cdot e\left(Q_A, PK_1 + P_{pub}\right)^{-v} \| PK_2\right)$

$= H_2\left(\mu^r \| PK_2\right) + \alpha - H_2\left(e(U,P)\cdot e\left(Q_A, PK_1 + P_{pub}\right)^{-v} \| PK_2\right)$

所以我們只需證明 $\mu^r = e(U,P)\cdot e\left(Q_A, PK_1 + P_{pub}\right)^{-v}$ 的等式是否成立即可：

$e(U,P)\cdot e\left(Q_A, PK_1 + P_{pub}\right)^{-v}$

$= e\left(rP + V(D_A + S_A), P\right)\cdot e\left(Q_A, PK_1 + P_{pub}\right)^{-v}$

$= e(rP, P)\cdot e(D_A + S_A, P)^V \cdot e\left(Q_A, x_A P + P_{pub}\right)^{-v}$

$= e(P,P)^r \cdot e(D_A, P)^V \cdot e(S_A, P)^V \cdot e(Q_A, x_A P)^{-V} \cdot e(Q_A, sP)^{-V}$

$= e(P,P)^r \cdot e(D_A, P)^V \cdot e(S_A, P)^V \cdot e(x_A Q_A, P)^{-V} \cdot e(sQ_A, P)^{-V}$

$= e(P,P)^r \cdot e(D_A, P)^V \cdot e(S_A, P)^V \cdot e(S_A, P)^{-V} \cdot e(D_A, P)^{-V} = e(P,P)^r = \mu^r$

若 σ 是合法的簽名，則 $[\beta]_{10} = \alpha$ 且 $\beta = F_1(m) \| (F_2(F_1(m)) \oplus m) = \beta = [\alpha]_2$。

因此，我們可以得到：

$F_2\left(_{l_2}|\beta|\right) \oplus |\beta|_{l_1}$

$= F_2(F_1(m)) \oplus (F_2(F_1(m)) \oplus m)$

$= m$

最後，若 $_{l_2}|\beta| = F_1(m)$，則即可驗證訊息的正確性。

4.2 不限制訊息長度之免憑證簽章

針對沒有限制長度的訊息，簽名的步驟與方法大致上跟前面所提的一樣，因此這裡我們只針對有修改的部份做討論：

[Setup]

在這個步驟我們只修改了 F_1，新的定義如下：

- $F_1 : \{0,1\}^* \rightarrow \{0,1\}^{l_2}$

[Sign]

此階段針對一欲簽名的訊息 $m \in \{0,1\}^*$，簽名者 A 會執行以下步驟來對 m 做簽名：

(1) 計算 $\beta = F_1(m) \| (F_2(F_1(m)) \oplus m)$，接著定義 $\alpha = [\beta]_{10}$。

(2) 將訊息 m 劃分為 $m_2 \| m_1$ 且 $m_1 \in \{0,1\}^{l_1}$。

(3) 隨機挑選一亂數 $r \in Z_q^*$，接著計算 $V = H_2(\mu^r \| PK_2) + \alpha$。

(4) 計算 $U = rP + V(D_A + S_A)$。

最後我們將 m 的簽名 $\sigma = (U, V)$ 及部份訊息 m_2 送給驗證方。

[Verify]

當驗證方的使用者收到系統公開參數、m_2、σ、以及簽名者的 ID 與 PK_A 後，需先根據 $e(PK_1, P_{pub}) = e(P, PK_2)$ 等式來驗證 PK_A 的合法性。

若 PK_A 是合法的，使用者再依以下步驟驗證 σ 是否合法：

(1) 使用者根據簽名 $\sigma=(U,V)$ 及公開的參數可計算出：

$$\alpha = V - H_2\left(e(U,P)\cdot e\left(Q_A, PK_1 + P_{pub}\right)^{-v} \| PK_2\right)$$

(2) 接著可推算出 $\beta=[\alpha]_2$。

(3) 回復部份訊息 $m_1' = F_2\left({}_{l_2}|\beta|\right)\oplus|\beta|_{l_1}$。

(4) 若 ${}_{l_2}|\beta| = F_1(m_2\|m_1')$ 則表示此簽章 σ 為合法簽章。

若簽章 σ 為合法簽章，我們可以回復訊息 $m = m_2\|m_1'$。

關於此改良方法的正確性，其證明與前面所述差不多，故這裡就不多做介紹。

第五章安全性與效能分析

此章節我們將會針對上一章節所提出的簽章系統做安全性的證明，我們會利用到 Random Oracle Model[7]證明工具設計出一套安全性模組，然後針對免憑證簽章中的 Type I 與 Type II 兩種攻擊者去進行安全性假設，接著我們會利用 Forking Lemma 輔助定理[17]來幫助我們將安全性假設導入到**定理 2** 之 CDH 難問題，然後再利用矛盾證明法來證明我們的簽章方法是安全的，最後我們也會與其他相關的簽章方法做簡單的效能分析與比較。

5.1 證明方法介紹

此小節我們將介紹 Random Oracle Model 的概念以及通常密碼學者如何利用它來證明簽章演算法的安全性，接著我們再介紹一個證明簽章演算法時所用到的輔助定理：Forking Lemma 的定義。

5.1.1 Random Oracle Model

1993 年，Bellare 及 Rogway[7]指出雜湊函數可被視為隨機函數，使用雜湊函數的簽章機制可以在 Random Oracle Model 下證明其安全性。為了證明一個數位簽章演算法的安全性，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(truly random)的情況下，任何一個機率式多項式演算法都沒辦法自行計算出此雜湊函數的輸出值，因此我們必須向外在的 Random Oracle 詢問此雜湊函數的輸出值，而 Random Oracle Model 的精神就是我們可

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，因此藉由反證法，我們可以證明我們欲證明的簽章演算法是安全的。

通常，我們會利用 Random Oracle Model 將我們的簽章系統模組化，我們欲利用此安全性模組來證明簽章演算法的安全性，首先我們會以外在獨立的 Random Oracle 來取代簽章演算法中的雜湊函數,如此所證明的簽章安全性才與雜湊函數無相依關係,在此模組中攻擊者可以向 Random Oracle 詢問相對應某一輸入的輸出值，攻擊者再藉由這些輸出值來破解簽章演算法,另一方面,此模組中存在著另一挑戰者(Challenger)，挑戰者會控制著 Random Oracle 的輸出值，並隨時紀錄 Random Oracle 的輸入與輸出值，挑戰者的目的就是利用攻擊者成功破解的結果來攻破公認的難問題，只要攻擊者有不可忽略的機率可以攻破簽章演算法，挑戰者同樣也可以攻破難問題，最後利用反證法即可證明簽章演算法的安全程度。

5.1.2 Forking Lemma

Forking Lemma 是密碼學者在證明簽章安全性時常使用到的輔助定理，根據 Pointcheval 及 Stern 對 Forking Lemma 的定義[17]，Forking Lemma 只適用在 Generic Signature 的格式下，滿足 Generic Signature 條件的簽章格式為(r,h,s)(r 代表著亂數，h 代表著輸入為(m,r)的雜湊值，s 代表著只和(r,m,h)相依的簽章。

在我們的簽章演算法中，我們可將$V=H_2(\mu^r \| PK_2)+\alpha$ 中的亂數 r 視作 r，且由於 α 中包含著訊息 m，因此我們可將 $H_2(\mu^r \| PK_2)+\alpha$ 當作另個 $H_2'(m,r)$ 並視為 h，而 $\sigma=(U,V)$ 即可視作只與 (r,m,h) 相依的 s，因此我們可以得知我們的簽章方法滿足 Generic Signature 的簽章格式。

定理 6：

(Forking Lemma)，在 Generic Signature 演算法下，令 A 為一個只能輸入公開資料機率式多項式時間之狀態機(Probabilistic Polynomial Time Turning Machine)。令 Q 為 A 向 Random Oracle 要求的次數。假設在時間 T，A 可以在機率 $\varepsilon > 7Q/2^k$ (k 為系統安全參數)下產生一組合法的簽章 (m,r,h,s);則 A 亦能夠在時間 $T' \le 84480TQ/\varepsilon$ 內產生二組簽章 (m,r,h,s) 及 (m,r,h',s')，其中 $h \ne h'$ (h' 為 A 換了 Random Oracle 後對於 (m,r) 的回傳值)。

5.2 安全性模組

此小節我們將針對免憑證簽章機制的安全性定義做討論，並根據攻擊者能力的不同，利用 Random Oracle Model 設計出兩種 Game。

5.2.1 Security Definition

根據 Al-Riyami 等學者的定義[1]，免憑證簽章的安全性通常會根據攻擊者的能力不同分成兩種情況討論，如下所述：

- Type I 攻擊者

此種情況是假設 PKG 是公正的 TTP，這類型的攻擊者 A_I 無法從 PKG 那裡得到 PKG 所生成的部份私密金鑰，但攻擊者 A_I 能夠偽造簽名者的私密金鑰以及有能力替換簽名者的公開金鑰，此種攻擊者我們歸為 Type I 攻擊者。

- Type II 攻擊者

此種情況是假設 PKG 不是公正的 TTP，這類型的攻擊者 A_{II} 可輕易地從 PKG 那裡得到 PKG 所生成的部份私密金鑰，但無法得到與簽名者相對應的私密金鑰以及替換簽名者的公開金鑰，基於這種假設的攻擊者我們歸為 Type II 攻擊者。

若免憑證簽章系統可以抵擋此兩種攻擊者的攻擊，則我們可以說此系統是安全的。此外，我們會根據攻擊者能力的不同設計出兩種 Game，每個 Game 將有五個 Random Oracle 會被攻擊者詢問，以下是我們對這五個 Oracle 的定義：

1. CreateUser：

輸入為使用者的身份識別 $ID \in \{0,1\}^*$，若 *ID* 已被建立在 CreateUser List 中，則不會發生任何事。否則，此 Oraclec 會執行免憑證簽章的 Partial-Private-Key-Extract、Set-Secret-Value、Set-Public-Key 三個演算法來得到 partial private key D_{ID}、secret value x_{ID} 以及 public key PK_{ID}，最後再把 *ID* 的資訊紀錄在 CreateUser List。此兩種情況皆會輸出 PK_{ID} 來回應。

2. PartialKeyExtract：

輸入為使用者的身份識別 *ID*，若 *ID* 已被建立在 CreateUser List 中，此 Oracle 會回覆相對應的 partial private key D_{ID}。否則，回覆一個錯誤信號"⊥"。

3. SecretValueExtract：

輸入為使用者的身份識別 *ID*，若 *ID* 已被建立在 CreateUser List 中，此 Oracle 會回覆相對應的 secret value x_{ID}。否則，回覆一個錯誤信號"⊥"。

4. PublicKeyReplace：

輸入為使用者的身份識別 *ID* 以及一隨機的 public key PK'_{ID}，若 *ID* 已被建立在 CreateUser List 中，此 Oracle 會將使用者 *ID* 原本的 PK_{ID} 替換成 PK'_{ID} 並紀錄於 CreateUser List 中。否則，不會發生任何事。

5. Sign：

輸入為使用者的身份識別 *ID* 以及一訊息 $m \in \{0,1\}^*$，若 *ID* 已被建立在 CreateUser List 中，此 Oracle 會回覆一合法的簽章 σ 並紀錄於資料表中，合法的簽章需滿足 $true \leftarrow Verify(params, m, \sigma, ID, PK_{ID})$ 的條件(在這裡的 PK_{ID} 是 CreateUser Oracle 所給的)。否則，回覆一個錯誤信號"⊥"。

5.2.2 Security Model

在 1988 年的時候，Goldwasser、Micali and Revist 提出了一個關於數位簽章安全性的概念：Existential Unforgeability Against Adaptive Chosen Message Attacks (EUF-ACMA)[11]，若我們能夠在攻擊者任意選擇訊息攻擊模式(Adaptively chosen message attack)下證明簽章的安全性，我們就稱該簽章方法滿足 EUF-ACMA。因此，為了證明免憑證簽章機制能夠滿足 EUF-ACMA 的安全概念，我們針對 Type I 攻擊者 A_I 以及 Type II 攻擊者 A_{II} 分別設計了兩個 Game： Game 1 及 Game 2，其定義如下：

Game 1：

這個 Game 是由一挑戰者 C 以及進行 Adaptively chosen message attack 的 Type I 攻擊者 A_I 所參與，其 Game 的流程如下：

- Setup： 挑戰者 C 先執行免憑證簽章的 Setup 演算法來得到系統公開參數 *params* 以及 PKG 的私密金鑰 *msk* ，接著挑戰者 C 把系統公開參數 *params* 傳送給攻擊者 A_I ，但 *msk* 只有挑戰者 C 知道。
- Queries： 在一有限的多項式時間內，攻擊者 A_I 會向挑戰者 C 詢問 5.2.1 小節中提及的五個 Oracle。
- Forgery： 若滿足以下條件，攻擊者 A_I 成功偽造一合法簽章 $\left(ID^*, PK_{ID^*}, m^*, \sigma^*\right)$ 並在此 Game 中獲得勝利。

  1. $true \leftarrow Verify\left(params, ID^*, PK_{ID^*}, m^*, \sigma^*\right)$。

  2. A_I 從未向 Sign Oracle 詢問過 $\left(ID^*, m^*\right)$。

3. A_I 從未向 PartialKeyExtract Oracle 及 SecretValueExtract Oracle 詢問過 ID^*。

定義 1：若在機率式多項式演算法時間內，攻擊者 A_I 有微乎其微的機率可以在 Game 1 中獲得勝利，我們則可以說此免憑證簽章面對 Type I 攻擊者是安全的。

Game 2：

這個 Game 是由一挑戰者 C 以及進行 Adaptively chosen message attack 的 Type II 攻擊者 A_{II} 所參與，其 Game 的流程如下：

- Setup： 挑戰者 C 先執行免憑證簽章的 Setup 演算法來得到系統公開參數 *params* 以及 PKG 的私密金鑰 *msk* ，接著挑戰者 C 把系統公開參數 *params* 及 *msk* 皆傳送給攻擊者 A_{II}。
- Queries： 在一有限的多項式時間內，攻擊者 A_{II} 會向挑戰者 C 詢問 5.2.1 小節中提及的五個 Oracle。
- Forgery： 若滿足以下條件，攻擊者 A_{II} 成功偽造一合法簽章 $\left(ID^*, PK_{ID^*}, m^*, \sigma^*\right)$ 並在此 Game 中獲得勝利。

  1. $true \leftarrow Verify\left(params, ID^*, PK_{ID^*}, m^*, \sigma^*\right)$。

  2. A_{II} 從未向 Sign Oracle 詢問過 $\left(ID^*, m^*\right)$。

  3. A_{II} 從未向 SecretValueExtract Oracle 詢問過 ID^*。

定義 2：若在機率式多項式演算法時間內，攻擊者 A_{II} 有微乎其微的機率可以在 Game 2 中獲得勝利，我們則可以說此免憑證簽章面對 Type II 攻擊者是安全的。

定義 3：若在機率式多項式演算法時間內，此免憑證簽章在面對上述 Type I 攻擊者及 Type II 攻擊者的情況下是安全的，我們則可以說此免憑證簽章機制滿足 Existential Unforgeability Against Adaptive Chosen Message Attacks (EUF-ACMA)的安全概念。

5.3 安全性證明

此小節我們將針對我們所提出的可訊息回覆之免憑證簽章機制做完整的安全性證明，由於有限制訊息長度方案與無限制訊息長度方案的證明方法類似，故在此我們只針對有限制訊息長度之方案做證明。

基於 CDH 難問題的假設，我們會證明我們的簽章方法在 Random Oracle Model 下面對 Type I 攻擊者及 Type II 攻擊者能夠符合***定義 3***並滿足 EUF-ACMA 的安全概念。以下我們將會利用本文章中 5.2.2 小節所提到的安全性模組來證明面對 Type I 攻擊者及 Type II 攻擊者時簽章的安全性。

5.3.1 Unforgeability against Type I Adversary

若存在一 Type I 攻擊者 A_I 可以攻破我們的簽章演算法，那我們可以假設存在另一個挑戰者 C，挑戰者 C 可以模擬我們的簽章演算法，並利用 A_I 為一黑盒子(black box)來達到解決 CDH 難問題的目的。為了讓簽章方法與 CDH 難問題有關連性，所以我們希望針對一特定 ID^*，挑戰者 C 會將 $Q_{ID^*} \leftarrow aP$ 及 $P_{pub} \leftarrow bP$，而挑戰者 C 的目的就是要利用 A_I 攻擊成功的結果來得到 $D_{ID^*} \leftarrow abP$ 這個值，關於證明過程如下所述：

- Setup： 為了能讓攻擊者 A_I 能順利執行，挑戰者 C 會先模擬整個簽章環境來得到系統參數，接著再模擬 Oracle 來回應攻擊者 A_I 的詢問，此證明中我們也考慮模擬雜湊函數 H_1、H_2、F_1、F_2 的 Random Oracle。此階段挑戰者 C 會執行以下步驟：

1. C 設置 $P_{pub} \leftarrow bP$ 。

2. C 將系統參數 $\langle G_1, G_2, q, e, P, P_{pub}, \mu, H_1, H_2, F_1, F_2, l_1, l_2 \rangle$ 傳送給 A_I ，並允許 A_I 可以執行 C 所模擬的簽章環境。

- Query： 任意多項式時間內，攻擊者 A_I 被允許向以下的 Oracle 詢問，這些 Oracle 皆是由挑戰者 C 所模擬，且為了保持一致性(consistency)與避免衝突(collision)，我們假定挑戰者 C 會紀錄每一個 Oracle 的輸入與輸出：

1. CreateUser：輸入為使用者的身份識別 ID_i ，若 ID_i 已被建立在 CreateUser-List 中，則不會發生任何事。否則，C 先選取 $x_i \in_R Z_q^*$ 、 $y_i \in_R Z_q^*$ ，接著設置 $PK_{ID_i} = \left(PK_{(ID_i,1)}, PK_{(ID_i,2)}\right) = \left(x_i P, x_i P_{pub}\right)$ ，因此：

   - 若 $i \neq t$，C 會設置 $Q_{ID_i} = H\left(ID_i\right) = y_i P$ 、 $D_{ID_i} = bQ_{ID_i} = by_i P = y_i bP = y_i P_{pub}$ 、以及 $S_{ID_i} = x_i Q_{ID_i} = x_i y_i P$ 。

   - 若 $i = t$ ，C 會設置 $Q_{ID_t} = H\left(ID_t\right) = aP$ 以及 $S_{ID_t} = x_t Q_{ID_t} = x_t aP$ ，接著設置 D_{ID_t} ="⊥"代表著無法計算出 Partial Private Key。

   最後，此兩種情況 C 皆會回應 $\left(Q_{ID_i}, PK_{ID_i}\right)$ 給 A_I 。

2. PartialKeyExtract：輸入為使用者的身份識別 ID_i，若 ID_i 已被 CreateUser Oracle 建立且 $i \neq t$，則 C 會回覆 $D_{ID_i} = y_i P_{pub}$ 給 A_I 。否則，C 則回覆"⊥"表示此模擬終止。

3. PublicKeyReplace：輸入為使用者的身份識別 ID_i 以及 A_I 所選的 PK'_{ID_i} ，是為了用來取代原本 ID_i 的 PK_{ID_i} ，若 ID_i 已被 CreateUser Oracle 建立，C 會將 ID_i 原本的 PK_{ID_i} 替換成 PK'_{ID_i} 。否則，C 則回覆"⊥"表示此模擬終止。

4. SecretValueExtract：輸入為使用者的身份識別 ID_i ，若 ID_i 已被 CreateUser Oracle 建立且 $i \neq t$ ，則 C 會回覆 x_i 給 A_I 。否則，C 則回覆"⊥"表示此模擬終止。(在這裡 A_I 所詢問的 secret value 與原本 CreateUser Oracle 設置的 public key 是相對應的)

5. H_1 query：輸入為使用者的身份識別 ID_i，若 ID_i 已被 CreateUser Oracle 建立，則 C 會回覆 Q_{ID_i} $(= y_i P)$ 給 A_I 。否則，C 則回覆"⊥"表示此模擬終止(C 會紀錄所有的 input/output 於 H_1-List，儲存格式為 $\left(ID_i, Q_{ID_i}\right)$)。

6. H_2 query：輸入為一 $w_i \in \{0,1\}^{|G_1|+|G_2|}$ (因為 $H_2\left(\mu^r \| PK_2\right)$ 中的 r 亦是隨機亂數，故我們將 $\mu^r \| PK_2$ 視為 w)，C 會先確認 H_2-List，若 w_i 已經在 H_2-List 中，則 C 會回覆 H_2-List 中相對應的 v_i 給 A_I ，否則，C 會回覆一隨機選取的 $v_i \in Z_q^*$ 給 A_I (C 會紀錄所有的 input/output 於 H_2-List，儲存格式為 $\left(w_i, v_i\right)$)。

7. F_1 and F_2 queries：若 A_I 是向 F_1 Oracle 詢問，則輸入為 l_1 bit 的值，C 會回覆一隨機 l_2 bit 的值給 A_I 。若 A_I 是向 F_2 Oracle 詢問，則輸入為 l_2 bit 的值，C 會回覆一隨機 l_1 bit 的值給 A_I (C 會紀錄所有的 input/output 於 F_1-List 與 F_2 -List)。

8. Sign：輸入為使用者的身份識別 ID_i 以及一訊息 $m \in \{0,1\}^{l_1}$，若 ID_i 尚未被 CreateUser Oracle 建立，則 C 會回覆"⊥"表示此模擬終止。若 ID_i 已被 CreateUser Oracle 建立，則分兩種情況討論，第一種情況是當 $ID_i \neq ID_t$ 時，C 會利用 CreateUser-List 中的 (S_{ID_i}, D_{ID_i}) 來產生一合法的簽章 σ，另一種情況是當 $ID_i = ID_t$ 時，C 會偽造一合法的簽章，其過程如下:

(1) 隨機選取 $U \in G_1$、$V \in Z$。

(2) 計算 $\beta = F_1(m) \| (F_2(F_1(m)) \oplus m)$。

(3) 計算 $\alpha = [\beta]_{10}$。

(4) 設置 $H_2\left(e(U,P)\cdot e\left(Q_{ID_i}, PK_{(ID_i,1)} + P_{pub}\right)^{-v} \| PK_{(ID_i,2)}\right) = V - \alpha$（我們可將 $e(U,P)\cdot e\left(Q_{ID_i}, PK_{(ID_i,1)} + P_{pub}\right)^{-v} \| PK_{(ID_i,2)}$ 視為 H_2-List 中的 w_i^*）。

(5) 將 $H_2\left(e(U,P)\cdot e\left(Q_{ID_i}, PK_{(ID_i,1)} + P_{pub}\right)^{-v} \| PK_{(ID_i,2)}\right) = V - \alpha$ 儲存到 H_2-List (儲存格式為 $\left(w_i^*, (V-\alpha)^*\right)$)，此時要考慮兩種情況:

- 若 w_i^* 已經在 H_2-List 中，但 output $(V-\alpha)^* \neq v_i$。
- 若 output $(V-\alpha)^* = v_i$，但 input $w_i^* \neq w_i$。

若上述任一種情況發生，C 則會立刻終止此 Query 且重新再進行一次 Sign Query。

(6) 得到一合法的簽章 $\sigma = (U,V)$ 並回覆給 A_I。

- Forgery： 除非攻擊者 A_I 異常終止，否則挑戰者 C 會一直與攻擊者 A_I 保持互動。經過所有 Oracle 的詢問後，若攻擊者 A_I 沒有異常終止，攻擊者 A_I 會成功偽造一合法簽章 $\left(ID^*, PK_{ID^*}\left(=PK_{(ID^*,1)}, PK_{(ID^*,2)}\right), m^*, \sigma^*\left(=U^*, V^*\right)\right)$，並在此 Game 中獲得勝利。

現在我們就可以利用上述的結果來破解 CDH 難問題，若 $ID^* \neq ID_t$，挑戰者 C 則回覆"⊥"表示此模擬終止，否則，若 $ID^* = ID_t$，表示攻擊者 A_I 可在挑戰者 C 所模擬的環境下成功偽造一合法簽章 $\sigma^*\left(=U^*, V^*\right)$。由 5.1.2 小節我們可以得知我們的簽章方法滿足 Generic Signature 簽章格式 (r, h, s)，因此根據 Forking Lemma 的定義[4]，若挑戰者 C 在利用自己所定義之 Random Oracle 的情況下，攻擊者 A_I 有不可忽略的機率偽造出一組簽章，那麼當挑戰者 C 在第二次模擬的中途換了一個 Random Oracle，攻擊者 A_I 也能對相同的 (r, m) 偽造出另一組簽章，因此，挑戰者 C 可以在相同的 ID^*、PK_{ID^*}、m^* 情況下得到另一個合法的簽章 $\sigma' = (U', V')$。由於 σ^* 與 σ' 皆是與 ID^*、PK_{ID^*}、m^* 相對應的合法簽章，所以我們可以得到 $U^* = rP + V^*\left(D_{ID^*} + S_{ID^*}\right)$ 以及 $U' = rP + V'\left(D_{ID^*} + S_{ID^*}\right)$，接著挑戰者 C 可以利用 $\left(U^*, U'\right)$ 來計算出 D_{ID^*}，其計算過程如下:

$$\left(U^* - U'\right) = \left(V^* - V'\right) \cdot \left(D_{ID^*} + S_{ID^*}\right)$$

$$\Rightarrow \left(D_{ID^*} + S_{ID^*}\right) = \left(V^* - V'\right)^{-1}\left(U^* - U'\right)$$

最後，挑戰者C可以得到$D_{ID^*}=\left(V^*-V'\right)^{-1}\left(U^*-U'\right)-S_{ID^*}$(根據 Type I 攻擊者的定義攻擊者$A_I$知道$S_{ID^*}$)，而$D_{ID^*}$也是挑戰者$C$欲破解之 CDH 難問題的結果(在給予$P$、$Q_{ID^*}\leftarrow aP$、$P_{pub}\leftarrow bP$情況下，可以得到$D_{ID^*}\leftarrow abP$)。因此，我們可以說挑戰者$C$破解了 CDH 難問題，但事實上 CDH 難問題為無法解決，因此根據矛盾證明法，我們可以得到我們的簽章方法在面對 Type I 攻擊者A_I時是安全的且符合***定義 1***。

在這裡我們要注意，在$PK_{ID^*}=\left(PK_{(ID^*,1)},PK_{(ID^*,2)}\right)=\left(PK_{(ID_t,1)},PK_{(ID_t,2)}\right)$的情況下，若$PK_{ID^*}$是一開始在 CreatUser Oracle 原始生成的公開金鑰，則我們即可直接由上述證明過程來解決 CDH 難問題，否則，根據 Type I 攻擊者的定義，攻擊者A_I具有替換公開金鑰的能力，若攻擊者A_I在 PublicKeyReplace Oracle 替換了PK'_{ID^*}為新的公開金鑰，此時PK'_{ID^*}亦可通過公開金鑰的驗證式$e\left(PK'_{(ID^*,1)},P_{pub}\right)=e\left(P,PK'_{(ID^*,2)}\right)$，因此這裡要考慮兩種攻擊者$A_I$替換$PK'_{ID^*}$的方法:

(1) 攻擊者A_I會直接選取一PK'_{ID^*}替換掉原本的PK_{ID^*}，此種情況挑戰者C無法得知與PK'_{ID^*}相對應的x_{ID^*}，因此也就無法得知$S_{ID^*}\left(=x_{ID^*}Q_{ID^*}\right)$並利用其來計算$D_{ID^*}=\left(V^*-V'\right)^{-1}\left(U^*-U'\right)-S_{ID^*}$。

(2) 攻擊者A_I會隨機選取一亂數r，計算$PK'_{ID^*}=rPK_{ID^*}=\left(rPK_{(ID^*,1)},rPK_{(ID^*,2)}\right)=\left(rx_{ID^*}P,rx_{ID^*}P_{pub}\right)$並替換掉原本的$PK_{ID^*}$，此種情況由於挑戰者$C$無法得知$r$，因此也就無法得知與$PK'_{ID^*}$相對應的$S_{ID^*}\left(=rx_{ID^*}Q_{ID^*}\right)$並利用其來計算$D_{ID^*}=\left(V^*-V'\right)^{-1}\left(U^*-U'\right)-S_{ID^*}$。

根據上述兩種替換方法，若挑戰者C能求得與PK'_{ID^*}相對應的x_{ID^*}或者 r，D_{ID^*}即可被計算出來且CDH難問題亦可順利破解。

在攻擊者A_I可替換公開金鑰的情況下，我們會利用到 The Knowledge of Exponent Assumption (KEA)，其定義如下[9,23]：

定理 7：

(*The Knowledge of Exponent Assumption*)，令g為一質數有序群(prime order group)$\langle g\rangle$的生成元。令$k=\log|\langle g\rangle|$為一秘密參數。針對任意的機率多項式時間(probabilistic polynomial time, PPT)之演算法A，若A的輸入為g和g^a(a是此範圍$[0,|\langle g\rangle|-1]$中隨機挑選出來的)，其輸出為(x,y)且$x\in\langle g\rangle$，則亦會存在另一 PPT 之提取者(extractor)E，E會在相同的輸入情況下輸出(x^r,y^r)(r為任意值)，使得k於足夠大的情況下滿足:

$$\Pr[y=x^a \text{ and } g^r\neq x]<\frac{1}{Q(k)}\quad (Q\text{為任意之多項式})$$

根據**定理 7** 我們可以知道：

- 於乘法群中已知$(A,B)=(g,g^b)$的情況下，若能計算出$(C,D)=(g^x,g^{bx})$並滿足$\log_A^C=\log_B^D$的條件，我們即可假設x的值可以被求出來。
- 於加法群中已知$(A,B)=(P,bP)$的情況下，若能計算出$(C,D)=(xP,xbP)$，我們即可假設x的值可以被求出來。

因此根據 KEA，若攻擊者 A_I 能從 $(P, P_{pub}) = (P, bP)$ 計算出 $(x_{ID^*}P, x_{ID^*}bP) = PK'_{ID^*}$ 或從 $PK_{ID^*} = (PK_{(ID^*,1)}, PK_{(ID^*,2)})$ 計算出 $(rPK_{(ID^*,1)}, rPK_{(ID^*,2)}) = PK'_{ID^*}$，我們即可假設攻擊者 A_I 能夠求出 x_{ID^*} 或 r，因此挑戰者 C 即有辦法從而得知，則挑戰者 C 即可計算出 D_{ID^*} 並解決 CDH 難問題。

5.3.2 Unforgeability against Type II Adversary

若存在一 Type II 攻擊者 A_{II} 可以攻破我們的簽章演算法，那我們可以假設存在另一個挑戰者 C'，挑戰者 C' 可以模擬我們的簽章演算法，並利用 A_{II} 為一黑盒子(black box)來達到解決 CDH 難問題的目的。為了讓簽章方法與 CDH 難問題有關連性，所以我們希望針對一特定 ID^*，挑戰者 C' 會將 $Q_{ID^*} \leftarrow aP$ 及 $PK_{(ID^*,1)} \leftarrow bP$，而挑戰者 C' 的目的就是要利用 A_{II} 攻擊成功的結果來得到 $S_{ID^*} \leftarrow abP$ 這個值，關於證明過程如下所述：

- Setup： 為了能讓攻擊者 A_{II} 能順利執行，挑戰者 C' 會先模擬整個簽章環境來得到系統參數，接著再模擬 Oracle 來回應攻擊者 A_{II} 的詢問，此證明中我們也考慮模擬雜湊函數 H_1、H_2、F_1、F_2 的 Random Oracle。此階段挑戰者 C' 會執行以下步驟：

  1. C' 設置 $P_{pub} \leftarrow sP$ 。
  2. C' 將系統參數 $\langle G_1, G_2, q, e, P, P_{pub}, \mu, H_1, H_2, F_1, F_2, l_1, l_2 \rangle$ 及系統私密金鑰 s 傳送給 A_{II}，並允許 A_{II} 可以執行 C' 所模擬的簽章環境。

- Query： 任意多項式時間內，攻擊者 A_{II} 被允許向以下的 Oracle 詢問，這些 Oracle 皆是由挑戰者 C' 所模擬，由於攻擊者 A_{II} 已知系統私密金鑰 s 以及所有的部份私密金鑰，所以 PartialKeyExtract Oracle 不會被詢問，且為了保持一致性(consistency)與避免衝突(collision)，我們假定挑戰者 C' 會紀錄每一個 Oracle 的輸入與輸出：

  1. CreateUser：輸入為使用者的身份識別 ID_i，若 ID_i 已被建立在 CreateUser-List 中，則不會發生任何事。否則，C' 先選取 $x_i \in_R Z_q^*$、$y_i \in_R Z_q^*$，因此：

- 若 $i \neq t$，C' 會設置 $Q_{ID_i} = H(ID_i) = y_i P$、$D_{ID_i} = sQ_{ID_i} = sy_i P = y_i sP = y_i P_{pub}$、以及 $S_{ID_i} = x_i Q_{ID_i} = x_i y_i P$、$PK_{ID_i} = (PK_{(ID_i,1)}, PK_{(ID_i,2)}) = (x_i P, x_i P_{pub})$。

- 若 $i = t$，C' 會設置 $Q_{ID_t} = H(ID_t) = aP$ 以及 $PK_{ID_t} = (PK_{(ID_t,1)}, PK_{(ID_t,2)}) = (bP, sbP)$，接著設置 $D_{ID_t} = y_t P_{pub}$ 以及 S_{ID_t} ="⊥" 代表著無法計算出使用者 ID_t 的 Private Key。

最後，此兩種情況 C' 皆會回應 (Q_{ID_i}, PK_{ID_i}) 給 A_{II}。

2. SecretValueExtract：輸入為使用者的身份識別 ID_i，若 ID_i 已被 CreateUser Oracle 建立且 $i \neq t$，則 C' 會回覆 x_i 給 A_{II}。否則，C' 則回覆"⊥"表示此模擬終止。(在這裡 A_{II} 所詢問的 secret value 與原本 CreateUser Oracle $i \neq t$ 時所設置的 public key 是相對應的)

3. H_1 query：輸入為使用者的身份識別 ID_i，若 ID_i 已被 CreateUser Oracle 建立，則 C' 會回覆 Q_{ID_i} $(= y_i P)$ 給 A_{II}。否則，C' 則回覆"⊥"表示此模擬終止(C' 會紀錄所有的 input/output 於 H_1-List，儲存格式為 (ID_i, Q_{ID_i}))。

4. H_2 query：輸入為一 $w_i \in \{0,1\}^{|G_1|+|G_2|}$ (因為 $H_2(\mu^r \| PK_2)$ 中的 r 亦是隨機亂數，故我們將 $\mu^r \| PK_2$ 視為 w)，C' 會先確認 H_2-List，若 w_i 已經在 H_2-List 中，則 C' 會回覆 H_2-List 中相對應的 v_i 給 A_{II}，否則，C' 會回覆一隨機選取的 $v_i \in Z_q^*$ 給 A_{II} (C' 會紀錄所有的 input/output 於 H_2-List，儲存格式為 (w_i, v_i))。

5. F_1 and F_2 queries：若 A_{II} 是向 F_1 Oracle 詢問，則輸入為 l_1 bit 的值，C' 會回覆一隨機 l_2 bit 的值給 A_{II}。若 A_{II} 是向 F_2 Oracle 詢問，則輸入為 l_2 bit 的值，C' 會回覆一隨機 l_1 bit 的值給 A_{II} (C' 會紀錄所有的 input/output 於 F_1-List 與 F_2-List)。

6. Sign：輸入為使用者的身份識別 ID_i 以及一訊息 $m \in \{0,1\}^{l_1}$，若 ID_i 尚未被 CreateUser Oracle 建立，則 C' 會回覆 "⊥" 表示此模擬終止。若 ID_i 已被 CreateUser Oracle 建立，則分兩種情況討論，第一種情況是當 $ID_i \neq ID_t$ 時，C' 會利用 CreateUser-List 中的 (S_{ID_i}, D_{ID_i}) 來產生一合法的簽章 σ，另一種情況是當 $ID_i = ID_t$ 時，C' 會偽造一合法的簽章，其過程如下:

   (1) 隨機選取 $U \in G_1$、$V \in Z$。

   (2) 計算 $\beta = F_1(m) \| (F_2(F_1(m)) \oplus m)$。

   (3) 計算 $\alpha = [\beta]_{l_0}$。

   (4) 設置 $H_2\left(e(U,P)\cdot e\left(Q_{ID_i}, PK_{(ID_i,1)} + P_{pub}\right)^{-v} \| PK_{(ID_i,2)}\right) = V - \alpha$ (我們可將 $e(U,P)\cdot e\left(Q_{ID_i}, PK_{(ID_i,1)} + P_{pub}\right)^{-v} \| PK_{(ID_i,2)}$ 視為 H_2-List 中的 w_i^*)。

   (5) 將 $H_2\left(e(U,P)\cdot e\left(Q_{ID_i}, PK_{(ID_i,1)} + P_{pub}\right)^{-v} \| PK_{(ID_i,2)}\right) = V - \alpha$ 儲存到 H_2-List (儲存格式為 $\left(w_i^*, (V-\alpha)^*\right)$)，此時要考慮兩種情況:

   - 若 w_i^* 已經在 H_2-List 中，但 output $(V-\alpha)^* \neq v_i$。
   - 若 output $(V-\alpha)^* = v_i$，但 input $w_i^* \neq w_i$。

若上述任一種情況發生，C'則會立刻終止此 Query 且重新再進行一次 Sign Query。

(6) 得到一合法的簽章$\sigma=(U,V)$並回覆給A_{II}。

- Forgery：除非攻擊者A_{II}異常終止，否則挑戰者C'會一直與攻擊者A_{II}保持互動。經過所有 Oracle 的詢問後，若攻擊者A_{II}沒有異常終止，攻擊者A_{II}會成功偽造一合法簽章$\left(ID^*, PK_{ID^*}\left(=PK_{(ID^*,1)}, PK_{(ID^*,2)}\right), m^*, \sigma^*\left(=U^*, V^*\right)\right)$，並在此 Game 中獲得勝利。

現在我們就可以利用上述的結果來破解 CDH 難問題，若$ID^* \neq ID_t$，挑戰者C'則回覆"⊥"表示此模擬終止，否則，若$ID^* = ID_t$，表示攻擊者A_{II}可在挑戰者C'所模擬的環境下成功偽造一合法簽章$\sigma^*\left(=U^*, V^*\right)$。由 5.1.2 小節我們可以得知我們的簽章方法滿足 Generic Signature 簽章格式(r,h,s)，因此根據 Forking Lemma 的定義[4]，若挑戰者C'在利用自己所定義之 Random Oracle 的情況下，攻擊者A_{II}有不可忽略的機率偽造出一組簽章，那麼當挑戰者C'在第二次模擬的中途換了一個 Random Oracle，攻擊者A_{II}也能對相同的(r,M)偽造出另一組簽章，因此，挑戰者C'可以在相同的ID^*、PK_{ID^*}、m^*情況下得到另一個合法的簽章$\sigma'=(U',V')$。由於σ^*與σ'皆是與ID^*、PK_{ID^*}、m^*相對應的合法簽章，所以我們可以得到$U^* = rP + V^*\left(D_{ID^*} + S_{ID^*}\right)$以及$U' = rP + V'\left(D_{ID^*} + S_{ID^*}\right)$，接著挑戰者$C'$可以利用$\left(U^*, U'\right)$來計算出$S_{ID^*}$，其計算過程如下:

$$\left(U^{*}-U'\right)=\left(V^{*}-V'\right)\cdot\left(D_{ID^{*}}+S_{ID^{*}}\right)$$

$$\Rightarrow\left(D_{ID^{*}}+S_{ID^{*}}\right)=\left(V^{*}-V'\right)^{-1}\left(U^{*}-U'\right)$$

最後，挑戰者C'可以得到$S_{ID^{*}}=\left(V^{*}-V'\right)^{-1}\left(U^{*}-U'\right)-D_{ID^{*}}$(根據 Type II 攻擊者的定義攻擊者$A_{II}$知道$D_{ID^{*}}$)，而$S_{ID^{*}}$也是挑戰者$C'$欲破解之 CDH 難問題的結果(在給予$Q_{ID^{*}}\leftarrow aP$及$PK_{\left(ID^{*},1\right)}\leftarrow bP$情況下，可以得到$S_{ID^{*}}\leftarrow abP$)。因此，我們可以說挑戰者$C'$破解了 CDH 難問題，但事實上 CDH 難問題為無法解決，因此根據矛盾證明法，我們可以得到我們的簽章方法在面對 Type II 攻擊者A_{II}時是安全的且符合***定義 2***。

5.4 效能分析

本小節我們將會把我們的簽章方法與 Zhang[25]及 Tso[21]的簽章方法做效能上的分析與比較，其比較結果如下所述：

根據表 5.4，由於 Zhang[25]與 Tso[21]的方法皆是 ID-based signature，所以都會有金鑰控管的問題，然而，我們提出的方法是基於免憑證簽章的概念，所以我們不會有此方面的問題。在效能運算方面，1Exp 代表一次的指數運算，1EC 代表一次的橢圓曲線運算，1e 代表一次的雙線性配對運算，所以由表 5.4 我們可以發現我們的方法在簽章跟驗證階段的效能運算都不會太差，甚至還要更好，但我們最主要的優勢就是我們在可訊息回覆的情況下沒有金鑰控管的問題。(註：*為有限制訊息長度之簽章方法)

表 5.4 效能分析與比較表。

	Key escrow problem	Sign	Verify
Our Scheme1*	N	1Exp+2EC	1e+1Exp
Our Scheme2	N	1Exp+2EC	1e+1Exp
Zhang's Scheme [25]1*	Y	1Exp+2EC	2e+1Exp+1EC
Zhang's Scheme [25]2	Y	1Exp+2EC	2e+1Exp+1EC
Tso's Scheme [21]1*	Y	1Exp+1EC	1e+1Exp+1EC
Tso's Scheme [21]2	Y	1Exp+1EC	1e+1Exp+1EC

第六章結論與未來展望

本論文中我們提出了第一個可訊息回復的免憑證簽章，並證明其安全性滿足 EUF-ACMA 的安全概念，在效能上與其他相關研究比較亦有不錯的表現。

由於我們的方法是基於免憑證簽章的架構，因此不只改善了傳統數位簽章在憑證管理上的問題亦解決了基於身份認證之簽章所產生的金鑰控管問題，同時由於具有訊息回復之特性，簡化了訊息及簽章在傳送時的總長度，因此我們的簽章方法相當適用於以頻寬為主要考量的行動網路服務以及需要大量對短訊息做簽章的相關應用。由於智慧型手機已漸漸普及，行動服務勢必為未來發展的重點，因此未來我們希望能將我們提出的可訊息回復之免憑證簽章實作於行動通訊網路上的電子交易、訊息認證等方面的應用上。

參考文獻

[1] S. Al-Riyami, K. Paterson, "*Certificateless public key cryptography*", Advances in Cryptology-Asiacrypt'03, Springer-Verlag, LNCS 2894, pp.452-473, 2003.

[2] G. Bertoni, L. Breveglieri, L. Chen, P. Fragneto, K. Harrison, and G. Pelosi, "*A pairing SW implementation for smart-cards*", Advances in Journal of Systems and Software, Vol. 81(7), pp.12401247, 2008.

[3] F. Bao, R. Deng, and H. Zhu, "*Variations of Diffie-Hellman Problem*", In Proceedings of ICICS 2003, Springer-Verlag, LNCS 2836, pp.301-312,2003.

[4] D. Boneh and M. Franklin,"*Identity-base encryption from Weil pairing*", Advances in Cryptology- CRYPTO 2001, Springer-Verlag, LNCS 2139, pp.213-239, 2001.

[5] P. S. L. M. Barreto, B. Libert, N. McCullagh, and J. Quisquater, "*Efficient and provably-secure identity-based signatures and signcryption from bilinear maps*", Advances in Cryptology -ASIACRYPT'05, LNCS 3778, pp.515–532, 2005.

[6] D. Boneh, B. Lynn, and H. Shacham,"*Short signature from Weil pairing*", Advances in Cryptology- ASIACRYPT 2001, Springer-Verlag, LNCS 2248, pp.514-532, 2001.

[7] M. Bellare and P. Rogaway, "*Random Oracles are Practical: A Paradigm for Designing Efficient Protocols*", Advances in 1st Conference on Communications Security, ACM, pp62–73, 1993 .

[8] W. Diffie, and M. Hellman, "*New directions in cryptography*", IEEE Transactions on Information Theory 22, pp.644-654, 1976.

[9] I. Damgard,"*Towards practical public key systems secure against chosen ciphertext attacks*", Advances in Cryptology-CRYPTO'91, pp.445–456,1991.

[10] T. ElGamal, "*A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*", Advances in Cryptology-CRYPTO'84, Springer-Verlag, LNCS 196, pp.10-18, 1985.

[11] S. Goldwasser, S. Micali, and R. L. Rivest, "*A digital signature scheme secure against adaptive chosen-message attacks*", Advances in SIAM Journal of Computing Vol.17(2), pp. 281-308, 1988.

[12] F. Hess. "*Efficient Identity Based Signature Schemes Based on Pairings*", In Proceedings of SAC 2002, LNCS 2595, Springer-Verlag, pp. 310-324, 2002.

[13] A. Joux, "*A one round protocol for tripartite Diffie-Hellman*," In proceedings of ANTS 4, Springer-Verlag, LNCS 1838, pp.385-393, 2000.

[14] N. Koblitz, "*Elliptic curve cryptosystems*", Advances in Mathematics of Computation, vol. 48, pp.203-209, 1987.

[15] V. Miller, "*Use of elliptic curves in cryptosystems*", Advances in CRYPTO 85, Springer-Verlag, LNCS 218,pp.417-426, 1985.

[16] K. Nyberg and R. A. Ruepple, "*Message recovery for signature schemes based on the discrete logarithm problem*", Advance in Cryptology-Eurocrypt'94, Springer-Verlag, LNCS 0950, pp.182-193, 1995.

[17] D. Pointcheval and J. Stern, "*Security proofs for signature schemes*", Advances in Cryptology- Eurocrypt'96, Springer-Verlag, LNCS 1070, pp. 387-398,1996.

[18] A. Shamir, "*Identity-based cryptosystems and signature schemes*", Advances in Cryptology- CRYPTO'84, LNCS 0196, pp.47–53, 1984.

[19] K. Shim, "*Efficient ID-based authenticated key agreement protocol based on the Weil pairing*", Advance in Electronics Letters 39 (8), pp. 653-654, 2003.

[20] W. Stallings,"*Cryptography and Network Security: Principles and Practice*," 3rd ed., Prentice Hall,2003.

[21] R. Tso, C. Gu, T. Okamoto, and E. Okamoto, "*Efficient ID-based digital signatures with message recovery*", Proceedings of the 6th International Conference on Cryptology and Network Security (CANS2007), Springer-Verlag, LNCS 4856, pp. 47-59, 2007.

[22] R. Tso, X. Yi, and X. Huang, "*Efficient and short certificateless signatures*", In proceedings of CANS'08, LNCS 5339, pp. 64-79, 2008.

[23] J. Wu, and D.R. Stinson "*An efficient identification protocol and the knowledge-of-exponent assumption*", Advances in Cryptology ePrint Archive: Report 2007/479.

[24] F. Zhang, and K. Kim, "*Efficient ID-based Blind Signature and Proxy Signature from Bilinear Pairings*", Proceedings of ACISP '03, Springer-Verlag, LNCS 2727, pp.312–323, 2003.

[25] F. Zhang, W. Susilo, and Y. Mu, "*Identity-based partial message recovery signatures (or How to shorten ID-based signatures)*", Advances in FC'05, Springer-Verlag, LNCS 3570, pp.45–56, 2005.

[26] IEEE Standard 1363-2000, "*Standard Specifications for Public Key Cryptography*", Available from http://grouper.ieee.org/groups/1363, 2000.

[27] 全國法規資料庫 - 電子簽章法 Available at "*http://law.moj.gov.tw/LawClass/LawContent.aspx?pcode=J0080037*"(2010.11).

[28] 林滔天，公開金鑰憑證註銷之研究，國立成功大學資訊工程系碩士論文，2006 年。

[29] 陳冠穎，公開金鑰基礎建設之探討與實務研究，世新大學資訊管理系碩士論文，2002 年。

[30] 陳坤男，使用霍夫曼樹建立具實用性的憑證廢止機制，國立東華大學資訊工程系碩士論文，2004 年。

[31] 羅建民，無需撤銷公開金鑰之密碼系統，國立成功大學資訊工程系碩士論文，2004 年。