一個有效率的免憑證代理簽章機制

紀汶承　陳昱圻　洪國寶
中興大學資訊科學與工程學系
{s9756020,s9756034, gbhorng}@cs.nchu.edu.tw

劉兆樑
亞洲大學資訊多媒體應用學系
jliu@asia.edu.tw

摘要

在數位化的時代，代理簽章為一種新的數位簽章機制。在代理簽章中，有一個原始簽章者，可以授權本身的簽章能力給其他的代理人，這些被授權的代理人我們稱為代理簽章者。

近年來，為了排除在傳統公開金鑰系統下公鑰憑證，以及在基於 ID 加密系統下金鑰託管的問題。免憑證公開金鑰加密系統被提出來處理這些問題。

本文我們提出一個有效率的免憑證代理簽章機制，將免憑證公開金鑰技術引進代理簽章中，原始簽章者除了可以將自己的簽章能力授權給代理簽章者，暫代自己的職務，更可以解決基於身分公開金鑰密碼系統所面臨的金鑰託管問題。我們提出的方法在假設 CDH 問題艱難於 random oracle model 下可以抵擋惡意的 KGC 與外部公鑰取代攻擊。

關鍵詞: 代理簽章(proxy signature)、數位簽章(digital signature)、免憑證公開金鑰加密系統(certificateless public key cryptography)

Abstract

Proxy signature is a new signature technique, such that an original signer is able to delegate his ability of signing to a proxy known as a “proxy signer”. The public key certificates incur costs many computations and communications in traditional public key cryptosystem. Identity public key cryptosystem (ID-PKC) is proposed to achieve certificateless. Unfortunately, key escrow is a problem in ID-PKC where key generation center can own all users’ private keys. Certificateless public key cryptosystem (CL-PKC) is used to solve this problem.

In this paper, we propose a new certificateless proxy signature scheme. Our scheme is provably secure in the random oracle model against the malicious KGC and outer public key replacement attack under the hardness assumption of the CDH problem.

Keyword: proxy signature, digital signature, certificateless public key cryptography

1. 前言

在傳統的公開金鑰加密系統，一個使用者的公鑰必須經過可信賴的授權憑證中心(CA)憑證，才能取得公鑰。不可避免的，就需要用到大量的儲存空間存放公鑰以及花更多的時間去計算管理這些憑證[5]。為了簡化憑證的過，Shamir 首先提出了基於身分加密系統(ID-PKC)[9]，不在需要原本公鑰憑證，取而代之的是公開的、已知的資訊來當作使用者的公鑰，例如 e-mail 地址或是 ID。但是卻產生了一個內在的問題，因為 ID-PKC 必須經由公鑰託管中心(KGC)的 master key 去產生每個使用者的密鑰。顯然地，一個惡意的 KGC 就有能力去偽造任何使用者的合法簽章。這就是所謂的”公鑰託管問題”。因此在 2003 年 Al-Riyami 和 Paterson 提出了一個免憑證公鑰加密法(CL-PKC)去取代原先的 PKC 中的公鑰憑證的使用，並解決了金鑰託管的問題[1]。而 CL-PKC 基本的想法是藉由使用者隨機選取的一個秘

密值以及 KGC 的 master key 去產生一組公密鑰。而 Boneh 等人提出短簽章機制[2,3]，更適合用於行動通訊上。

在 1996 年由 Mambo, Usuda 和 Okamoto 提出了代理簽章機制[7,8]。一個原始簽章者可以將自己的簽署能力授權給代理簽章者，而代理簽章者就能代表原始簽章者簽署文件。而驗證者可以驗證其正確性以及分辨這是否為一個代理簽章。代理簽章可以應用在許多方面，例如:電子商務、行動裝置…等。代理簽章的機制，是由原始簽章者 Alice 產生一個簽章，其中包含一些授權資訊，並將此授權傳送給代理簽章者 Bob。而 Bob 使用這些資訊去產生一個代理簽章的密鑰，並使用這把密鑰去簽署文件。當驗證者收到文件以及簽章時，會使用 Alice 以及 Bob 的公鑰去驗證是否為合法的代理簽章。

我們提出免憑證代理簽章機制(CL-PS)，可以更實際地來應用。本文第二章描述一些基礎知識以及 CL-PS 的安全性 model 等，第三章簡述 Chen 等人的方法，第四章我們提出一個有效率的免憑證代理簽章機制，第五章為安全性分析，最後為本文結論。

2. 基礎知識

本章節簡短介紹 Bilinear pairing、CL-PS 機制及特性、CL-PS 安全性 model 以及困難問題的假設。

2.1. Bilinear pairing

Bilinear pairing 是在橢圓曲線上的一種特性，存在著一個 pairing 同構的值。令 G_1 是在橢圓曲線上點的加法循環群，G_2 是在一個有限體中的乘法循環群。G_1 和 G_2 有相同的 order q。Bilinear pairing 就像是一個 mapping 函式 $e: G_1 \times G_1 \to G_2$，並有以下的特性:

1. 可計算性:給定 $P, Q \in G_1$，存在一個多項式時間的演算法去計算出 $e(P,Q) \in G_2$
2. 雙線性:對任意整數 $x, y \in Z_q^*$，存在 $e(xP, yP) = e(P,P)^{xy}$， $P \in G_1$。
3. 非退化性:假設 P 是 G_1 的生成元，則 $e(P,P)$ 是 G_2 的生成元。

2.2. 困難問題的假設

Computational Diffie-Hellman (CDH)問題: 給定 (P, aP, bP) 去計算出 abP，其中 P 是 G_1 中的生成元，G_1 是乘法群，$a, b \in Z_q^*$ 是未知的。

2.3. 免憑證代理簽章機制

2.3.1. 免憑證代理簽章的性質:

- 可驗證性:驗證者可以驗證簽章的正確性，並確信為原始簽章者的認可。
- 不可偽造性:包含 KGC 在內，任何人都不可以偽造原始簽章者的授權簽章以及代理簽章者的代理簽章。
- 可區分性:任何人都可以區分代理簽章與一般簽章的不同，以及原始簽章者與代理簽章者之間的關係。
- 不可否認性:當代理簽章者產生有效的代理簽章，原始簽章者不可否認這個行為。
- 防止濫用性:代理簽章者不可以簽署未授權的資訊也不可以將簽署能力非法轉移他人。

2.3.2. 免憑證代理簽章機制

CL-PS 擁有的成員為 KGC、原始簽章者、代理簽章者、驗證者，以及下列幾個演算法組成:

- **Setup** (k)**:** 由 KGC 執行的演算法，給一個安全參數 k，KGC 會產生 master-key s 以及系統參數。
- **Partial-Private-Key** $(params, s, ID_i)$**:** KGC 執行此演算法，輸入系統參數、master-key s、使用者的 ID_i。此演算法會產生使用者的 Partial-Private-Key D_i。
- **UserKeyGen** (ID_i)**:** 此演算法由使用者執行，輸入使用者的 ID_i，然後輸出使用

者的公/密鑰 X_i / x_i。

- **Partial-Proxy-Key** (ID_i, x_i, D_i, m_w)**:** 此演算法由使用者執行，輸入私鑰 x_i、partital-private-key D_i 以及授權資訊 m_w，然後輸出使用者的 Partial-Proxy-Key S_i。

- **Proxy Sign** $(M, ID_{i_1}, ID_{i_2}, x_i, D_i, S_{i_1}, S_{i_2}, m_w)$**:** 此演算法由代理簽章者執行，輸入文件 M、原始簽章者/代理簽章者的身分 ID_{i_1} / ID_{i_2}、代理簽章者密鑰/部分密鑰 x_i / D_i、原始簽章者/代理簽章者的部份簽章 S_{i_1} / S_{i_2} 以及授權資訊 m_w。輸出一個免憑證代理簽章 σ。

- **Verify:** 驗證者藉由原始簽章者 ID_A，代理簽章者 ID_B，A、B 的公鑰 X_A、X_B 文件 M、授權資訊 m_w 去驗證代理簽章的正確性，若是有效的則輸出 true，否則輸出 false。

2.3.3. CL-PS 的安全 model

在 CL-PKC 中有兩種型態的攻擊模式[4]: 一種 type I 是攻擊者 A_1 其代表的是外部攻擊者的角色，不能存取 master-key，但可以取代公鑰；另一種 type II 是攻擊者 A_2 其代表 KGC 的角色，可以存取 master-key 但沒有取代公鑰的能力。在很多的文章中[4,6]使用 game I 來模擬 type I 的攻擊者，利用另一個 game II 來模擬 type II 的攻擊者。利用此 game 來證明 CL-PS 可以滿足可驗證性以及不可偽造性證明。

3. 相關研究

Chen 等人提出可證明安全性的免憑證代理簽章機制[4]，由下列幾個演算法組成:

Setup: G 是一個 Bilinear map 的生成元，其演算法如下:

1. 輸入安全參數 l 去產生 (G_1, G_2, e)，其中 $e: G_1 \times G_1 \to G_2$ 是一個 Bilinear map。
2. 選擇一個生成元 $P \in G_1$。
3. 選擇一個 master-key $s \in Z_q^*$ 並設定 $P_0 = sP$。
4. 選擇雜湊函數 $H_1, H_2, H_3, H_4 : \{0,1\}^* \to G_1^*$。

並且輸出公開的系統參數 param= $(G_1, G_2, e, q, P, P_0, H_1, H_2, H_3, H_4)$，訊息空間 $\overline{M} \in \{0,1\}^*$。

Partial-private-key-extract: 輸入使用者 $ID_i \in \{0,1\}^*$ 並且計算 $Q_i = H_1(ID_i)$，回應 partial-private-key $D_i = sQ_i$。

Set-secret-value: 輸入 param 以及使用者 ID_i，並選擇一個 $x_i \in Z_q^*$，回應 x_i 為使用者的 secret value。

Set-public-key: 輸入 param 以及使用者的 x_i，產生公鑰 $P_i = x_i P$。

Set-private-key: 輸入 param 以及使用者的 D_i、x_i、P_i 以及 ID_i，回應 private-key 為 $S_i = D_i + x_i T_i$，其中 $T_i = H_2(ID_i \| P_i)$。

Partial-proxy-key-generate: 輸入 param、S_A 和 warrant m_w，其中原始簽章者 A 的 ID 為 ID_A、公鑰為 P_A，然後去計算 Partial-proxy-key 給代理簽章者 B，過程如下:

1. 隨機選取 $r_A \in Z_q^*$ 並計算 $R_A = r_A P$。
2. 計算 $U_A = H_3(m_w \| ID_A \| P_A \| R_A)$ 和 $K_A = S_A + r_A U_A$。
3. 輸出 (m_w, R_A, K_A) 給 B，然後令 $\theta_A = (R_A, K_A)$ 為 partial proxy key。

Partial-proxy-key-verify: 當收到 (m_w, R_A, K_A) 時，代理簽章者 B 會確認 $e(K_A, P) = e(Q_A, P_0) e(T_A, P_A) e(U_A, R_A)$ 是否成立，假使成立，則接受 (m_w, R_A, K_A)，否則拒絕。

Set-proxy-key:假使代理簽章者 B 接受 (m_w, R_A, K_A)，則 B 會設定他的 proxy key 為 (R_A, K_A, S_B)。

Proxy-sign:代理簽章者 B 去簽署一份文件 M，需要 ID_B、公鑰 P_B 以及 proxy key (R_A, K_A, S_B)，並執行以下演算法:

1. 隨機選取 $r_B \in Z_q^*$ 並計算 $R_B = r_B P$。
2. 計算 $U_B = H_4(m \| m_w \| ID_B \| P_B \| R_B)$ 和 $V = K_A + S_B + r_B U_B$。
3. 輸出代理簽章 $\sigma = (R_A, R_B, V)$。

Proxy-verify:透過原始簽章者的 ID_A、P_A，代理簽章者的 ID_B、P_B，去驗證這份文件其中包含 (m, m_w, σ)，並執行以下演算法:

1. 檢查文件 M 是否符合授權資訊 m_w。假使不符合則拒絕 σ，反之則執行以下演算法:
2. 計算 $Q_A = H_1(ID_A)$、$Q_B = H_1(ID_B)$、$T_A = H_2(ID_A \| P_A)$、$T_B = H_2(ID_B \| P_B)$、$U_A = H_3(m_w \| ID_A \| P_A \| R_A)$ 和 $U_B = H_4(m \| m_w \| ID_B \| P_B \| R_B)$。
3. 檢查 $e(V,P) = e(Q_A + Q_B, P_0) e(T_A, P_A) e(T_B, P_B) \cdot e(U_A, R_A) e(U_B, R_B)$ 是否成立，若成立則接受代理簽章 σ，反之拒絕。

4. 提出的免憑證代理簽章機制

提出有效率的 CL-PS 機制有幾個角色分別為 KGC、原始簽章者 A、代理簽章者 B、驗證者，以及下列幾個演算法組成:

- **Setup** (k)**:** 給一個安全參數 k，KGC 會選定一個 prime order 為 q 的加法循環群 G_1，P 為 G_1 之生成元，和另一個相同 order 乘法循環群 G_2，以及雙線性映射函數 $e: G_1 \times G_1 \to G_2$。KGC 接著隨機選一個 $s \in Z_q^*$ 當作 master-key 並計算 $P_0 = sP$，選 3 個 MapToPoint hash function 分別為 $H_1: \{0,1\}^* \to G_1$、$H_2: \{0,1\}^* \to G_1$、$H_3: \{0,1\}^* \to G_1$。此演算法最後輸出系統參數 $params = (G_1, G_2, e, P, P_0, H_1, H_2, H_3)$、以及 message space 為 $M \in \{0,1\}^*$。
- **Partial-Private-Key** $(params, s, ID_i)$**:** KGC 執行此演算法，輸入系統參數 params、master-key s、使用者的 $ID_i \in \{0,1\}^*$，計算 $Q_i = H_1(ID_i)$ 進而得到 partial-private-key $D_i = sQ_i$。
- **UserKeyGen** (ID_i)**:** 使用者輸入使用者的 ID_i，此演算法隨機選一個 $x_i \in Z_q^*$ 然後設定其公鑰 $X_i = x_i P$、私鑰 x_i。
- **Partial-Proxy-Key** (ID_i, x_i, D_i, m_w)**:** 計算 $S_i = x_i H_2(m_w) + D_i$，回應 S_i。
- **Proxy Sign** $(M, ID_{i_1}, ID_{i_2}, x_i, D_i, S_{i_1}, S_{i_2}, m_w)$**:** 產生一個免憑證的代理簽章動作如下，A 為原始簽章者、B 為代理簽章者、M 為文件、m_w 是授權資訊。
  1. A 將 S_A、m_w 傳送給 B。
  2. B 輸出 M 的免憑證代理簽章 $\sigma = x_B H_3(M \| m_w \| ID_A \| ID_B) + (S_A + S_B)$
- **Verify:** 驗證者藉由原始簽章者 ID_A，代理簽章者 ID_B，公鑰 X_A、X_B 文件 M、授權資訊 m_w 去驗證，驗證式子如下。

$$e(\sigma, P) = ? e(H_1(ID_A) + H_1(ID_B), P_0) \cdot e(H_3(M \| m_w \| ID_A \| ID_B), X_B) \cdot e(X_A + X_B, H_2(m_w))$$

若等式成立則輸出 true，否則輸出 false。

5. 安全性分析

5.1 可驗證性以及不可偽造性證明

提出的機制是可被證明的安全在假設 CDH 問題是難解的情況下。

Theorem 1. 若存在一個 type I 的攻擊者 A 有 ε 的機率依照 Game I 的型式去攻擊提出的機制以達到偽造簽章的目的，則存在一個演算法 B 有機率 $\varepsilon' \geq \frac{1}{q_{H_1} e^2}\varepsilon$ 可以破解 CDH 問題，q_{H_1} 為至多次的 H_1 詢問。

Proof. B 給定一個在 G_1 中 CDH 問題的 (P, aP, bP)，攻擊者 A 與 B 透過以下詢問做互動。最後 B 可以靠 A 的偽造簽章來解 CDH 的問題。

Setup: B 設定 $P_0 = aP$，參數 params= $(G_1, G_2, e, P, P_0, H_1, H_2, H_3)$。然後將 P_0 以及參數 params 傳送給 A。A 設定原始簽章者公密鑰對 X_A / x_A 以及代理簽章者公密鑰對 X_B / x_B。

Attack: A 在多項式時間以內依調整方式可以執行下列詢問。

- H_1 **query:** B 會維持一個 H_1 list，內有幾個欄位 (ID_j, α_j, Q_j)，此 list 初始狀態是空的。A 可以對於 ID_i 做詢問。
  1. 若 ID_i 在之前有被詢問過，則 B 按照 H_1 list 回應 Q_i 給 A。
  2. 若 ID_i 先前未被詢問，B 會隨機產生 $\alpha_i \in Z_q^*$，以及一個隨機位元 $c_i \in \{0,1\}$ 其中 $\Pr[c_i = 0] = 1/q_{H_1}$。q_{H_1} 為至多 H_1 的詢問次數。
  3. 若 $c_i = 0$，則 B 回應 $Q_i = \alpha_i(bP)$，並將 (ID_j, α_j, Q_j) 加入 H_1 list。
  4. 若 $c_i = 1$，則 B 回應 $Q_i = \alpha_i P$ 給 A，並將 (ID_j, α_j, Q_j) 加入 H_1 list。
- H_2 **query:** B 會維持一個 H_2 list，內有幾個欄位 $(M, m_{w,} ID_{i_1}, ID_{i_2}, \beta_i, W_i)$，此 list 初始狀態是空的。A 可以對於 ID_{i_2} 做詢問。
  1. 若 ID_{i_2} 在之前有被詢問過，則 B 按照 H_2 list 回應 W_i 給 A。
  2. 若 ID_{i_2} 先前未被詢問，B 會隨機產生 $\beta_i \in Z_q^*$，回應 $W_i = \beta_i P$ 給 A，並將 $(M, m_{w,} ID_{i_1}, ID_{i_2}, \beta_i, W_i)$ 加入 H_2 list。
- H_3 **query:** B 會維持一個 H_3 list，內有幾個欄位 (ID_i, m_w, r_i, R_i)，此 list 初始狀態是空的。A 可以對於 ID_i 做詢問。
  1. 若 ID_i 在之前有被詢問過，則 B 按照 H_3 list 回應 R_i 給 A。
  2. 若 ID_i 先前未被詢問，B 會隨機產生 $r_i \in Z_q^*$，回應 $R_i = r_i P$ 給 A，並將 (ID_i, m_w, r_i, R_i) 加入 H_3 list。
- **Partial-Private-Key query:** B 會維持一個 K list，內有幾個欄位 (ID_j, x_j, D_j, X_j)，A 依然可以詢問其 $PPK(ID_i)$，若之前被詢問過 B 會回應先前的結果，否則會回去執行 H_1 query 再回應 PPK query。B 的回應會如下:
  1. 若 $c_i = 0$，則 B 失敗且終止。
  2. 若 $c_i = 1$ 且 (ID_i, x_i, D_i, X_i) 存在 K list，則回應 D_i。
  3. 若 $c_i = 1$ 且 (ID_i, x_i, D_i, X_i) 中的 D_i 不存在，設定 $D_i = \alpha_i P_0$ 並將加入 K list。
- **Proxy sign query:** 當收到代理簽章的詢問 $PS(M, ID_{i_1}, ID_{i_2}, x_i, D_i, S_{i_1}, S_{i_2}, m_w)$，B

會根據 H_1 list、H_2 list 和 H_3 list 執行如下:

1. 若對應的 $c_i = 0$，B 中止且宣告失敗。
2. 否則回應 $\sigma = r_i X_B + \beta_i (X_A + X_B) + \alpha_i P_0$ 給 A。

- **Forgery:** A 輸出一個原始簽章者與代理簽章者 $U^* = \{U_A^*, U_B^*\}$ 其 ID 為 $L_{ID}^* = \{ID_A^*, ID_B^*\}$、對應公鑰 $L_{PK}^* = \{X_A^*, X_B^*\}$、訊息為 $L_M^* = \{M^*\}$ 以及授權資訊 m_w^*。在不失一般性的情況下，我們訂定 $i = B$ 時，ID_B^* 在先前未被詢問 Proxy sign。A 輸出一個偽造簽章 σ^*，此偽造的 CLAS 必須滿足以下式子:

$$e(\sigma, P) = ?\, e(H_1(ID_A) + H_1(ID_B), P_0) \cdot e(H_3(M \| m_w \| ID_A \| ID_B), X_B) \cdot e(X_A + X_B, H_2(m_w))$$

若 ID_B^* 對應的 $c_B \neq 0$ 則 B 會失敗。當 A 能偽造一個簽章 σ^* 相當於 B 可以得到以下式子

$$abP = \alpha_i^{-1}\left(\sigma^* - \left(r_i X_B + \beta_i \left(X_A + X_B\right)\right)\right)$$

經過以上正規的證明，假設 B 能夠解得 CDH 問題在機率 $\varepsilon' \geq \frac{1}{q_{H_1} e^2}\varepsilon$ 下。我們定義以下幾個事件

E1: B 在任何 query 中皆不失敗。

E2: B 在 Forgery 中不失敗。

當 E1、E2 成立時，B 成功。

Claim 1. B 在 A 的任何詢問下都不會失敗的機率至少為 $1/e^2$。

Pf: 因為 $\Pr[c_i = 0] = 1/q_{H_1}$，在兩個詢問下 Proxy sign、Partial-Private-Key 不會失敗的機率為 $(1-1/q_{H1})^{q_{H1}} \geq 1/e$，所以 $\Pr[E1] \geq 1/e^2$。

Claim 2. A 輸出合法的偽造代理簽章後 B 不失敗的機率為 $1/q_{H_1}$。

Pf: 當 A 產生的偽造代理簽章其中 $c_B = 0$ 其他 $c_i = 1$。因為 $\Pr[c_B = 0] = 1/q_{H1}$，可以推得 $\Pr[E2] = 1/q_{H_1}$。

最後我們可以得到 B 不失敗的機率為 $\varepsilon' = \Pr[E1 \cap E2] \geq \frac{1}{q_{H_1} e^3}\varepsilon$。所以我們推得若 A 要得到一個合法的偽造代理簽章之機率等價於 B 要破解 CDH 問題的機率。顯而易見地我們的 CL-PS 機制在相同的假設下也能抵擋 type II 攻擊。證明過程類似 Theorem 1，我們省略。因此我們提出的機制滿足了**可驗證性**以及**不可偽造**的能力。

5.2 可區分性證明

原始簽章者與代理簽章者的公鑰皆出現在驗證的式子當中，且授權訊息 m_w 也包含在其中，故任何人都可以從授權訊息中決定代理簽章者的身分，滿足**可區分性**。

5.3 不可否認性證明

在 m_w 中必須明確指出原始簽章者與代理簽章者之間的關係、必要資訊(包含原始簽章者的公鑰及參數)以及授權的內容與能力，且 m_w 須在安全的通道下傳送。在代理簽章中，簽署的密鑰包含了原始以及代理簽章者的私鑰，其中還包含了 $Q_i = H(ID_i)$，若 A 想偽造一個自己的私鑰來否認授權簽名，等價於求 hash 函數的單向性問題。一旦替原始簽章者產生一個有效的代理簽章，而代理簽章的密鑰只有代理簽章者本身能產生，所以代理者無法否認自己的簽章，故滿足**不可否認性**。

5.4 防止濫用證明

由於有授權資訊 m_w 的存在，以及 m_w 出現在代理簽章的驗證等式中，使得代理簽章者不可合法簽署未經授權的文件(已明確規定其能力以及身分)，當然也無法將

簽署能力非法轉移他人，滿足了**防濫用性**。

6. 討論與比較

在表 1.中我們提出的機制與 CHEN Hu 的機制做比較，我們列舉出一些 operations，pairing 以 pr 表示，scalar multiplication 以 S 表示， L 表示在 G_1 元素的長度。並忽略了一些可以預先計算的花費。

由此可見，PPKG 表 Partial-Proxy-Key-Generate 之計算成本、PS 表示 Proxy-Sign 之計算成本、PV 表示 Proxy-Verify 之計算成本、以及|PS|表 proxy signature 之通訊成本。顯而易見，我們比 Chen 等人在|PS|還省空間，在 PS 也更有效率。而在 PV 與 PPKG 上是差不多的。

表 1. 我們的方法與 Chen 等人之比較

Scheme	PPKG	PS	PV	|PS|
Chen et al.	1S	2S	2pr	3L
Our scheme	1S	1S	2pr	1L

7. 結論

數位簽章的使用提供了資料完整性、可驗證性、不可否認性，這讓現今數位環境的資訊更具保障。在代理簽章的實際應用上，一個公司的主管可賦予其職務代理人簽章的能力，讓主管不在公司的期間，代理人能確實執行其工作。本文提出了一個有效率的代理簽章機制，也滿足了代理簽章所需要的各個性質。我們的方法在 random oracle model 下假設 CDH 問題是棘手情況下可以抵擋惡意金鑰產生中心以及公鑰取代攻擊。

致謝

本研究由國科會補助完成，計畫編號：NSC96- 2628-E-005-076-MY3 與 NSC98-2221-E-468-012。

參考文獻

[1] S. Al-Riyami and K. Paterson, *Certificateless public key cryptography*, Asiacrypt2003, LNCS 2894, pp. 452-473, Springer-Verlag, 2003.

[2] D. Boneh and X. Boyen, *Short Signatures Without Random Oracles*, Eurocrypt2004, LNCS 3027, pp. 56-73, 2004.

[3] D. Boneh, H. Shacham, and B. Lynn, *Short signatures from the Weil pairing*, Journal of Cryptology, Vol. 17, No. 4, pp. 297-319, 2004.

[4] H. Chen, F. Zhang, R. Song, *Certificateless proxy signature scheme with provable security.* Journal of Software, 2009,20(3):692-701.

[5] P. Gutmann, *PKI: It's not dead, just resting*, IEEE Computer, 35(8), pp. 41-49,2002.

[6] B. C. Hu, D. S. Wong, Z. Zhang, and X. Deng, *Key Replacement Attack Againsta Generic Construction of Certificateless Signature*, ACISP 2006, LNCS 4058, pp.235-246, Springer-Verlag, 2006.

[7] M. Mambo, K. Usuda and E. Okamoto. "*Proxy Signatures:Delegation of the Power to Sign Message*", IEICE Trans.Fundamentals, Vol. E79 A, No. 9, 1996

[8] M. Mambo, K. Usuda, E. Okamoto. *Proxy signatures fordelegating signing operation*. In: 3rd ACM Conference onComputer and Communications Security(CCS'96), pp.48-57. New York: ACM Press, 1996

[9] A.Shamir, *Identity based cryptosystems and signature schemes*, Crypto 1984,LNCS, Vol.196, pp. 47-53, Springer-Verlag, 1984.