植基於智慧卡的無爭議簽章方法
RSA-Based Undeniable Signature Using Smart Cards

Cheng-Lung Lee(李正隆) and Chu-Hsing Lin (林祝興)

Department of Computer Science and Information Engineering,
Tunghai University
181 Taichung-kang Rd., Sec. 3, Taichung, 407 Taiwan, R.O.C.
E-mail: chlin@mail.thu.edu.tw

摘要

在本論文中，我們將智慧卡導入 Gennaro、Krawczyk 與 Rabin 於 1997 年所提出的「植基於 RSA 之無爭議簽章 (RSA-Based Undeniable Signature)」[9]；由於智慧卡的運算能力較差，因此我們利用了 C.H. Lin 與 C.C. Chang 於 1994 年提出的適於 RSA 加密法的伺服輔助計算協定的方法[1]，讓智慧卡將計算量大的運算委託給計算能力較強的終端機，以縮短計算時間，並且不洩漏該計算之隱密資料。

關鍵字：無爭議簽章，伺服輔助計算，植基於 RSA 之無爭議簽章。

Abstract

In this paper, we apply smart cards in the "RSA-Based Undeniable Signature" which was proposed by Gennaro and Krawczyk in 1997. Because of the weakness of the computing power of smart cards, we utilize the "Server-Aided Computation Protocol for RSA Enciphering Algorithm" proposed by Lin and Chang in 1994, and design a new protocol. In the proposed protocol the terminal will help smart card to compute the necessary operation, and there is not any secret data leaked out.

Keyword: Undeniable Signature, Server-Aided Computation, RSA-Based Undeniable Signature.

一、簡介

由於網路的盛行與電子商務的蓬勃發展，人們對於網路上傳遞資訊的安全性越來越重視，而資訊的安全性一般來說不外乎就是機密性(Confidentiality)、完整性(Integrity)、識別(Identification)、鑑別(Authentication)與不可否認性(Non-repudiation)，以目前被廣泛應用的數位簽章(Digital Signature)技術來說，它可以達到資料鑑別(Data Authentication)與不可否認性；雖然數位簽章相當的方便，但卻不適合於某些特定的應用，舉例來說，軟體公司為了避免該公司的軟體被植入病毒或者特洛伊木馬之類的程式，他們會在程式中嵌入一個數位簽章，然而他們希望只有購買該軟體的合法使用者能夠驗證簽章，但一般的數位簽章的特性是所有拿到簽章者公開金鑰的使用者皆能自己對該簽章做驗證，因此並不適合於這類的應用，而 D. Chaum 與 H. van Antwerpen 在 1989 年所提出的無爭議簽章 (Undeniable Signature) [2]正好符合了這類應用的需求，因為無爭議簽章的驗證必須透過簽章者的協助。

本文將智慧卡導入 Gennaro、Krawczyk 與 Rabin 於 1997 年所提出的「植基於 RSA 之無爭議簽章」[9]；由於智慧卡的運算能力

較弱，本文提出一個協定，運用伺服器輔助運算的技巧，讓智慧卡將運算量較大的資料經過某些特定函式的轉換後傳送到計算能力較強的機器計算出智慧卡所需要的結果；此外，外部使用者將無法從在該協定過程中取得任何隱密資料，再配合智慧卡硬體的安全特性，使用者將無法輕易地複製智慧卡，進而可以避免 Yvo Desmedt 與 Moti Yung[13]以及 Jakobsson[7]所提出的攻擊。

本文的架構如下，第二節將簡介有關無爭議簽章的相關研究，第三節介紹伺服器輔助運算的相關理論，第四節描述系統架構，第五節為安全性分析，第六節作一結論。

二、相關研究

(一) Chaum 的無爭議簽章

Chaum 於 1989 年首次提出無爭議簽章的概念，在 1990 年時另外提出了一個「零知識無爭議簽章(Zero-Knowledge Undeniable Signature)」[3]，在 Chaum 所提出的零知識無爭議簽章中有兩個協定，一個是確認協定(Confirmation Protocol)另一個則是否認協定(Deniable Protocol)，確認協定的用途是讓簽章者證明該簽章的正確性，反之，否認協定可以讓簽章者證明驗證者所送過來的簽章與使用者所持有的文件並不相符。在此我們簡單的介紹確認協定的過程。

假設 p 為一大質數，g 是 GF(p)中的一個原根，簽章者 Alice 選擇一亂數 x 作為私密金鑰(Private Key)，其中 $x\in[1,p-1]$且 $\gcd(x, p-1) = 1$，而公開金鑰為 $y=g^x \bmod p$，當 Alice 想要簽署一份文件 m 時，計算出 $z = m^x \bmod p$，Z 即為該文件之簽章；當使用者 Bob 想要驗證此簽章時會進行下列步驟：

1. Bob 隨機選擇兩個亂數 $a, b\in[1,p-1]$，之後計算出 $r=m^a g^b \bmod p$，並將結果 r 傳送給 Alice。
2. Alice 收到 r 之後先產生一亂數 $q\in[1,p-1]$，並計算出 $s_1=r\cdot g^q \bmod p$, $s_2=g^{qx}\cdot r^x \bmod p$，將 s_1 與 s_2 傳回給 Bob。
3. Bob 收到 s_1 與 s_2 後將 a,b 傳給 Alice。
4. Alice 檢查 $r \stackrel{?}{=} m^a g^b \bmod p$，若等式成立則將 q 傳給 Bob，若不成立則中斷協定。
5. Bob 收到 q 後即可驗證 $s_1 \stackrel{?}{=} r\cdot g^q \bmod p$, $s_2 \stackrel{?}{=} (g^x)^{b+q}\cdot z^a \bmod p$，若等式成立則表示該簽章正確無誤。

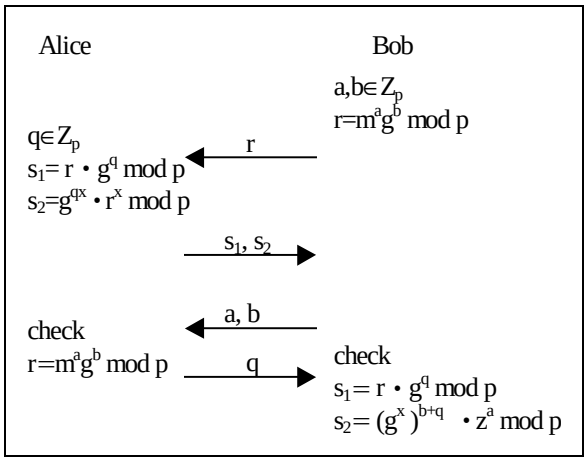

圖 1、零知識無爭議簽章(Zero-Knowledge Undeniable Signature) — 確認協定(Confirmation Protocol)

(二) 植基於 RSA 之無爭議簽章(RSA-Based Undeniable Signature)

Gennaro、Krawczyk 與 Rabin 於 Crypto'97 提出一個植基於 RSA 之無爭議簽章法，該簽章法除了與 Chaum 的零知識無爭議簽章一樣具有確認協定與否認協定，而且還具有可轉換(convertible)成一般簽章的特性，也就是說簽章者若公開其部分私密金鑰，使用者便可以利用公開的私密金鑰將原本的無爭議簽章轉換成一般的簽章[5][6]，往後若要對簽章進

行驗證時便不再需要透過確認協定與簽章者溝通，以下我們假設簽章者為 Alice 而驗證者為 Bob，並針對確認協定、否認協定與可轉換性作一介紹。

1 金鑰與簽章產生階段(Key and Signature Generation Phase)

(1) 產生兩個長度至少 512 位元的大質數 p、q，使得 n=pq。

(2) 選擇出一個質數 e 與一整數 d，使其滿足 $ed \equiv 1 \pmod{(p-1)(q-1)}$。

(3) 選擇一對參數(w, S_w)，符合下列條件：$w \in Z_n^*$, $w \neq 1$, $S_w = w^d \bmod n$。

(4) 令(n, w, S_w)為公開金鑰，(e, d)為私密金鑰。

(5) 假設 Alice 所要簽章的的文件為 m，則先讓 m 經過雜湊函數的運算產生 H(m)，並且令 $\hat{m}$ =H(m)，而簽章 $S_m = \hat{m}^d \bmod n$。

(6) Alice 將公開金鑰與簽章(n, w, S_w, m, S_m)傳給使用者 Bob。

2 簽章確認協定(Signature Confirmation Protocol)

(1) 使用者 Bob 選擇兩個亂數 $i, j \in \{1,\ldots,n\}$，之後計算出 $Q = S_m^i S_w^j \bmod n$，並將 Q 傳給 Alice。

(2) Alice 計算出 $A = Q^e \bmod n$，並將 A 傳回給 Bob。

(3) Bob 驗證 $A \stackrel{?}{=} \hat{m}^i w^j \bmod n$，若等式成立則 Bob 可以確定簽章的正確性，反之則 Bob 無法確定簽章之正確性。

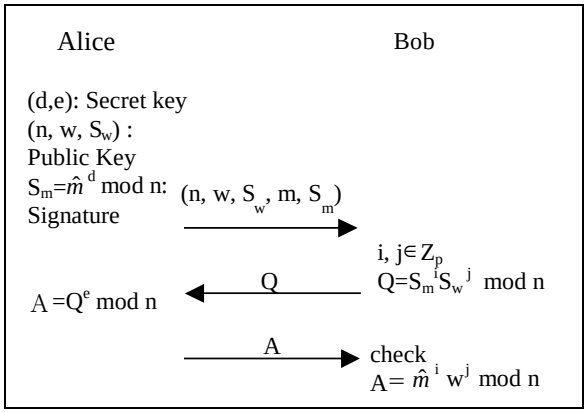

圖 2、植基於 RSA 之無爭議簽章(RSA-Based Undeniable Signature) — 確認協定 (Confirmation Protocol)

3 簽章否認協定(Signature Deniable Protocol)

(1) Bob 選擇一亂數 i=4b，$b \in Z_k$，與 $j \in \{1,\ldots,n\}$，其中 k 為系統選定的一個整數，計算出 $Q_1 = \hat{m}^i w^j \bmod n$，$Q_2 = S_m^i S_w^j \bmod n$，並將 Q_1 與 Q_2 傳給 Alice。

(2) Alice 計算出 Q_1/Q_2^e，並且從{1,…,k}之中一個一個測試也就是錯誤嘗試法以找出 i 的值，並令 A = i,，假如找不出 i 的值則表示該簽章並非假簽章，因此可中斷該協定。

(3) Bob 若收到 A，則檢查 $A \stackrel{?}{=} i$，如果等式成立表示 Bob 可以相信該簽章非該文件之簽章，反之則表示 Bob 無法確定簽章之正確性。

在此協定中 Bob 會選擇一個數字 k，而從該協定來看，若 Alice 要欺騙 Bob 的成功機率即為 1/k，如果 Bob 選擇 k=1024，並執行否認協定 10 次時，則 Alice 欺騙使用者的成功機率為 $1/2^{100}$。

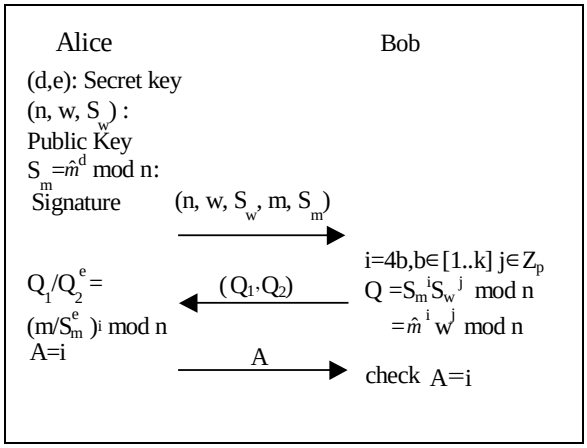

圖 3、植基於 RSA 之無爭議簽章(RSA-Based Undeniable Signature) — 否認協定(Deniable Protocol)

4 可轉換性(Convertible)

在這一個無爭議簽章的方法中，簽章者只要公開 e，公開金鑰就變成(e, n)，私密金鑰為 d，而原本的無爭議簽章就成為一般的 RSA 簽章。

三、伺服器輔助計算

Matsumoto 於 Crypto '88 提出加速秘密計算於不安全裝置的方法[11]，而該方法的效率(Performance)在[10]中被加以討論，除此之外，C.H. Lin 與 C.C. Chang 於 1994 年提出一個適於 RSA 加密法的伺服輔助計算協定，在本文中我們對 C.H. Lin 與 C.C. Chang 協定加以修改以適用於無爭議簽章的需求，以下我們針對適於RSA加密法的伺服輔助計算協定做一簡要描述。

(一) RSA 加密法的伺服輔助計算協定

該協定主要目的就是利用伺服器來協助計算 $C=M^e \bmod n$，並且不對伺服器洩漏 C 與 M 的資訊，其中(e, n)為公開金鑰，且 n=pq，p 與 q 為大質數，而 M 為明文，C 則為密文，圖一為該協定的過程。

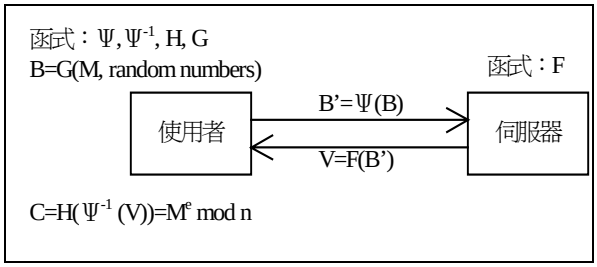

圖 4、RSA 加密法的伺服輔助計算協定

在圖一中Ψ, Ψ^{-1}, H, G 是屬於使用者端的函式，而 F 則為伺服器端的函式，以下是整個協定的過程。

(1) 使用者產生 t 個亂數($a_1, a_2,\ldots,a_t$)，其中 $p \nmid a_i$、$q \nmid a_i$，也就是 p 與 q 不是 a_i 的因數，並計算出 $a_0 = (M \prod_{i=1}^{r} a_i) \bmod n$，其中 r<t。

(2) 使用者計算出 $b_i=a_i^2 \bmod n$，其中 i=0,1,2,…,t，而所有計算出來的 b_i 值組成一向量 $B=(b_0,b_1,\ldots,b_t)$。

(3) 利用一隨機排列函式Ψ將向量 B 重組：$B'=\Psi(B)= (b_0',b_1',\ldots,b_t')$，並將 B'傳給伺服器。

(4) 伺服器收到 B'後計算出 $V=F(B')=(v_0,v_1,\ldots,v_t)$，並將 V 傳回給使用者，其中 $v_i=(b_i')^{(e-1)/2} \bmod n$。

(5) 使用者收到 V 後計算出 $U=\Psi^{-1}(V)=(u_0,u_1,\ldots,u_t)$，其中 Ψ^{-1},為Ψ逆排列。

(6) 使用者計算出

$$C = H(U) = (u_0 a_0)((\prod_{i=1}^{r} u_i)(\prod_{i=1}^{r} a_i))^{-1} \bmod n$$

，而 $C=M^e \bmod n$。

四、系統架構

(一) 系統概觀

在本文所提出的架構中，我們設定了以下的劇情：簽章者為一公司，該公司發行簽章的目的是為了讓員工以及顧客檢驗該公司所發出之文件或軟體的完整性與正確性，或者讓顧客檢驗該公司所發行的軟體沒有被放置惡意的程式(如病毒或特洛伊木馬)，因此使用者在取得軟體或文件後必須向該公司要求取得簽章。

首先使用者必須先向簽章者申請智慧卡與簽章，簽章者收到使用者的申請之後產生包含公開金鑰與簽章的智慧卡，當使用者要驗證簽章時會將智慧卡插入某台終端機所附的讀卡機中，此終端機可以是個人電腦、個人數位助理(PDA)、掌上型電腦(Pocket PC)、行動電話或嵌入式系統(Embedded System)，當智慧卡要進行某些複雜的運算時，會將相關數據送給該終端機作計算，之後該終端機再將結果傳回智慧卡，智慧卡便可執行無爭議簽章的協定。

在 Lin 與 Chang 所提出的 RSA 加密法伺服輔助計算協定協定中，伺服器必須知道使用者的公開金鑰(e, n)以便計算出 $C= M^e \bmod n$，但是在本文中是讓智慧卡運算植基於 RSA 之無爭議簽章，所以智慧卡必須先產生兩個亂數 i 與 j，然後計算出 $S_m^i S_w^j \bmod n$ 以及 $\hat{m}^i w^j \bmod n$，而本文所提出的方法則是要讓終端機協助智慧卡對該算式做運算。

(二) 協定流程

本文所提出的系統架構僅針對簽章的確認與否認協定，並無考慮無爭議簽章的可轉換性。以下為系統流程，在本系統中共有五個階段：金鑰產生階段、簽章與智慧卡產生階段、簽章確認協定、簽章否認協定與簽章更新協定。

1. 金鑰產生階段(Key Generation Phase)

在此階段中所有的動作由簽章者所完成，過程如下。

(1) 產生兩個長度至少 512 位元的大質數 p、q，使得 n=pq。

(2) 選擇出一個質數 e 與一整數 d，使其滿足 $ed \equiv 1 \bmod ((p-1)(q-1))$。

(3) 選擇一對參數(w, S_w)，符合下列條件：$w \in Z_n^*$, $w \neq 1$, $S_w = w^d \bmod n$。

(4) 令(n, w, S_w)為公開金鑰，(e, d)為私密金鑰。

2 簽章與智慧卡產生階段(Signature and Smart Card Generation Phase)

與上一個階段相同，此階段所有動作也是由簽章者所完成。

(1) 假設要簽章的文件為 m，讓 m 經過一單向雜湊函數 H(x)的運算，得到 m'= H(m)，並產生一隨機序號 Rand，再將 m'與 SN 以及要產生的智慧卡卡號(CID)再次經過單向雜湊函數 H(x, y, z)的運算得到 $\hat{m}$ =H(m'|| CID|| Rand)，而簽章 $S_m=(\hat{m})^d \bmod n$。

(2) 將$(n, w, S_w, S_m, Rand)$寫入智慧卡的記憶體中，並將智慧卡與文件 m 發送給使用者，該智慧卡之卡號 CID 與 Rand 則存入簽章者的資料庫中。

3 簽章確認協定(Signature Confirmation Protocol)

步驟一：使用者將智慧卡插入終端機後，輸入個人識別碼(PIN)，智慧卡確認識別碼正確

無誤後，將卡號 CID 傳給終端機，終端機再將 CID 傳送給驗證伺服器要求進行簽章確認協定，驗證伺服器收到要求後便進行以下步驟。

(1.1) 確認 CID 正確無誤後隨機產生一亂數 s 並計計算出 $\hat{s}$ = H(s ‖ Rand ‖ CID) ，$S_t = \hat{s}^d \cdot \hat{s}$ 。

(1.2) 將(S_t,s)傳給終端機。

(1.3) 終端機收到 S_t 後，將文件 m 經過雜湊函數運算出 m’並將(m',s,S_t)傳給智慧卡。

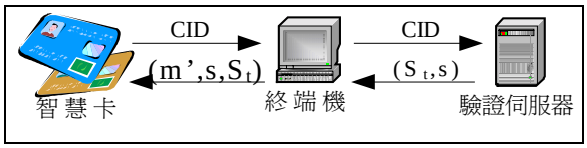

圖 5、簽章確認協定---步驟一

步驟二：智慧卡收到(m' ,S_t)後進行以下步驟。

(2.1) 智慧卡計算出 $\hat{m}$ =H(m’ ‖ CID ‖ Rand)，$\hat{s}$ = H(s ‖ Rand ‖ CID)，$V_s=(S_m \cdot S_t/\hat{s})^2 \bmod n = (S_m \cdot \hat{s}^d)^2 \bmod n$。

(2.2) 智慧卡產生 t 個亂數$(a_1, a_2,\cdots,a_t)$，與一亂數 r 且 1<r<t，其中 p∤a_i、q∤a_i，並計算出 $a_0 = (\hat{m} \cdot \hat{s})(\prod_{i=1}^{r} a_i) \bmod n$ 。

(2.3) 計算出一向量 B={$b_i \in Z_n$ |$b_i = a_i^2 \bmod n$,0<i<t }。

(2.4) 利用一隨機排列函式Ψ將向量 B 重組：B’=Ψ(B)= $(b_0',b_1',\ldots,b_t')$。

(2.5) 智慧卡隨機產生 I,J∈Z_n^*且 I∈Z_o，之後將(I, J, V_s’, B’, n)傳給終端機。

(2.6) 終端機在收到(I, J, V_s’, B’, n)後，計算

$$\alpha = (V_s)^{(I-1)/2} \cdot S_w^{\,J} \bmod n$$

$$\beta = w^J \bmod n$$

$$V = F(B) = \{v_i \in Z_n \mid v_i = (b_i')^{(I-1)/2} \bmod n, 0 \le i \le t\}$$

(2.7) 終端機將 α 傳給驗證伺服器。

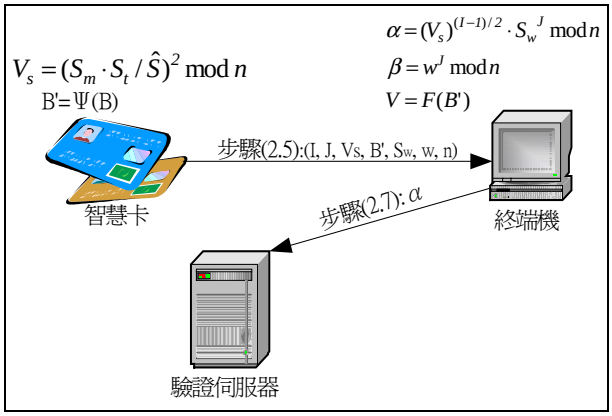

圖 6、簽章確認協定---步驟二

步驟三：驗證伺服器收到 α 後會進行以下步驟。

(3.1) 驗證伺服器計算出

$$A \equiv \alpha^e \equiv (\hat{s} \cdot \hat{m})^{I-1} w^J \pmod{n}$$

(3.2) 將 A 傳回終端機。

(3.3) 終端機收到 A 後將(A, β ,V)傳給智慧卡。

步驟四：智慧卡收到(A, β ,V)後進行以下步驟。

(4.1) 計算出

U=$\Psi^{-1}(v_1,v_2,\ldots,v_t,v_{t+2}) = (u_0,u_1,\ldots u_t)$。

(4.2) 智慧卡計算出

$$\gamma \equiv H(U) \equiv (u_0)(\prod_{i=1}^{r} u_i)^{-1} \equiv (\hat{m} \cdot \hat{s})^{I-1} \pmod{n}$$

(4.3) 終端機收到 A 之後檢查 $\beta \cdot \gamma \stackrel{?}{=} A$，若等式成立表示此次簽章確認協定成功，且該文件與簽章相符，反之則不能確定文件的正確性。

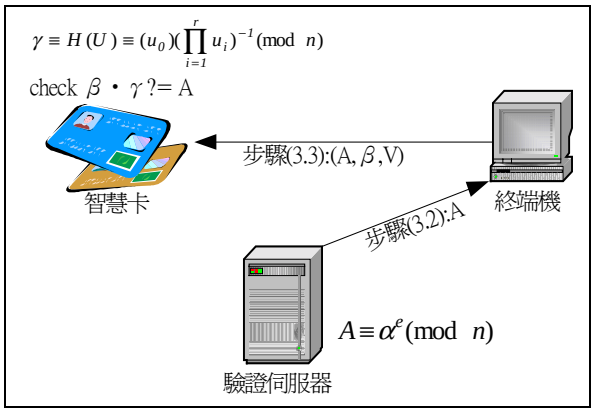

圖 7、簽章確認協定---步驟三、四

4 簽章否認協定(Signature Deniable Protocol)

此協定的步驟與確認協定相似，亦是由使用者將 m 運算出 m'後傳給智慧卡，之後進行下列步驟。

步驟一：與簽章確認協定之步驟一相同。

步驟二：

(2.1) 終端機將假文件 m_{fake} 經過雜湊函數運算出 m_{fake}'後將(m', S_t)傳給智慧卡。

(2.2) 智慧卡計算出 $\hat{m}_{fake}$ =H(m_{fake}' || CID || Rand)， $\hat{s}$ = H(s || Rand || CID)，V_s=(S_m • S_t/ $\hat{s}$) mod n =(S_m • $\hat{s}^d$) mod n。

(2.3) 智慧卡產生 t 個亂數(a_1, a_2,…,a_t)，與一亂數 r 且 1<r<t，其中 p$\nmid a_i$、q$\nmid a_i$，並計算出 $a_0 = (\hat{m} \cdot \hat{s})(\prod_{i=2}^{r} a_i) \bmod n$ 。

(2.4) 計算出一向量 B={$b_i \in Z_n$ |b_i= a_i^2 mod n,0<i<t }。

(2.5) 利用一隨機排列函式Ψ將向量 B 重組：B'=Ψ(B)= (b_0',b_1',…,b_t')。

(2.6) 計算出 V_s'=V_s^2 mod n，並且隨機產生 I,J$\in Z_n^*$且 I$\in Z_o$，之後將(I, J, V_s', B', n)傳給終端機。

(2.7) 終端機在收到(I, J, V_s', B', n)後，計算

$$\alpha = (V_s')^{(I-1)/2} \cdot S_w^{\ J} \bmod n$$
$$\beta = w^J \bmod n$$
$$V = F(B') = \{v_i \in Z_n \mid v_i = (b_i')^{(I-1)/2} \bmod n, 0 \le i \le t\}$$

並將(β,V)傳給智慧卡

(2.8) 智慧卡計算出

$$\gamma \equiv (u_0)(\prod_{i=1}^{r} u_i)^{-1} \equiv (\hat{m} \cdot \hat{s})^{I-1} (\text{mod } n)$$

$$Q_1 \equiv \gamma \cdot \beta \equiv (\hat{m}_{fake} \cdot \hat{s})^{I-1} \cdot w^J (\text{mod } n)$$

並將 Q_1 傳回終端機。

(2.9) 終端機將(α,Q_1)傳給驗證伺服器。

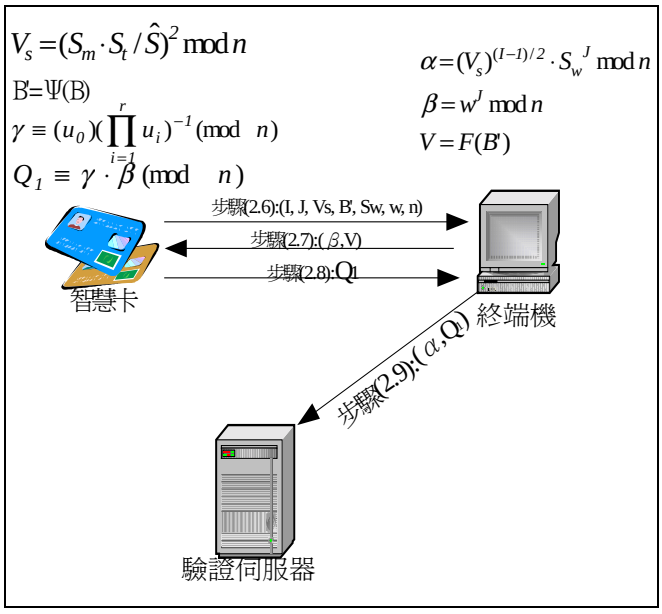

圖 8、簽章確否認協定---步驟二

步驟三：伺服器收到(α,Q_1)後進行下列步驟。

(3.1) 驗證伺服器計算出

$$Q_2 \equiv \alpha^e \equiv (\hat{s} \cdot \hat{m})^{I-1} w^J (\text{mod } n)$$
$$\frac{Q_1}{Q_2} \equiv \frac{(\hat{m}_{fake} \cdot \hat{s})^{I-1} \cdot w^J}{(\hat{s} \cdot \hat{m})^{I-1} w^J} \equiv (\frac{\hat{m}_{fake}}{\hat{m}})^{(I-1)} (\text{mod } n)$$

(3.2) 伺服器利用錯誤嘗試法計算出(I-1)的

值，假如找出該值則令 A=(I−1)，並將 A 傳回終端機，若找不出該值則中斷此次協定。

步驟四：終端機收到 A 之後檢查 $A \stackrel{?}{=} (I-1)$，若等式成立表示該文件與簽章不符，反之則無法確定文件之正確性。

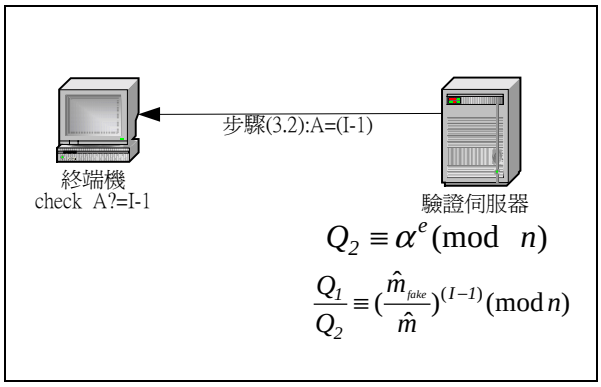

圖 9、簽章否認協定---步驟三、四

5 更新簽章 (Signature Update Protocol)

由於智慧卡具有可重複讀寫的功能，簽章者可透過安全的方式讓智慧卡中的簽章得以更新，而在本文所提出的協定中會將一隨機序號放入智慧卡中，並將該序號儲存於簽章者的資料庫中，因此智慧卡若要於網路上安全的更新簽章，可由簽章者直接利用該序號作為對稱式加密法之秘密金鑰，將新的簽章加密後傳送到智慧卡中，但本文提出另一個方法，只需要作一次乘法與除法以及一次的單向雜湊函式的運算，便可將簽章安全的傳送至智慧卡端，使得更新簽章的過程中，外部的使用者無法取得更改過的簽章。以下為此協定的步驟。

步驟一：智慧卡將 CID 傳送給終端機，終端機再將 CID 傳送給驗證伺服器。

步驟二：伺服器確認 CID 的正確性後，首先產生一亂數 s，再將要更新的簽章所屬的文件 m_2 經過經過單向雜湊函式產生 $m_2'=H(m_2)$，讓 m_2'、智慧卡卡號與該智慧卡中的亂數值再次經過單向雜湊函式的運算產生 $\hat{m}_2 = H(m_2' \| CID \| Rand)$，$\hat{s} = H(s \| CID \| Rand)$。

步驟三：伺服器計算：$S_t = \hat{s} \cdot (\hat{m}_2)^d \bmod n$，並將$(s,S_t)$傳給終端機。

步驟四：終端機收到(s,S_t)後將該資料傳給智慧卡，智慧卡便可計算出：

$\hat{s} = H(s \| CID \| Rand)$

$S_{m_2} = (S_t / \hat{s}) \bmod n = (\hat{m}_2)^d \bmod n$

步驟五：智慧卡進行簽章確認協定與否認協定以確認 S_{m_2} 的正確性。如果簽章正確無誤則智慧卡便可用 S_{m_2} 取代 S_m。

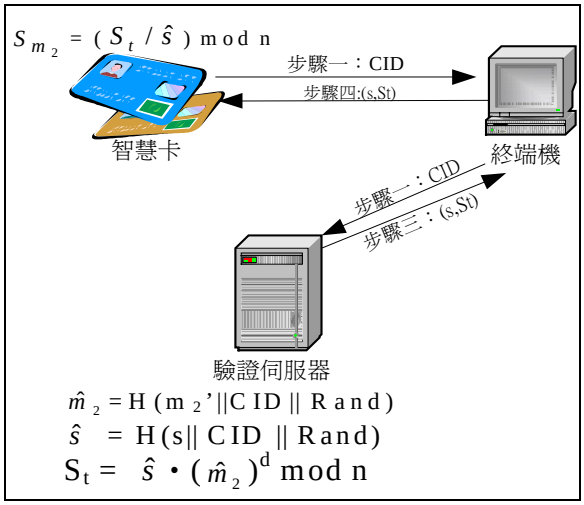

圖 10、簽章更新協定

五、複雜度與安全性分析

由於本文所提出的簽章確認與否認協定中，智慧卡所作的計算是相當類似的，因此在本節所作的分析都將只針對確認協定。

(一) 複雜度分析

在複雜度方面，本文所提出的架構中，簽章確認與否認協定中智慧卡的計算量是相同的，所以在此只針對確認協定的複雜度作

一討論；在確認協定中，智慧卡需要執行兩次的單向雜湊函數的運算，以及(t +2r+2)次的模乘，一次的除法與一次的逆運算(inverse operation)，根據 Euclid's extended algorithm，逆運算需要大約(0.843ln(n)+1.47)次的除法，而步驟(2.4)與(4.1)中的隨機排列，由於在實作時只需要去記憶每個向量值的指標位址，所以這兩個步驟的計算成本可以被忽略。在通信成本方面，也就是在步驟(2.5)與(3.3)中，智慧卡共需傳送 2t+8 個長度至多為 ln(n)的整數。而在簽章更新協定中智慧卡只需要做一次的乘法與一次的單向雜湊函數，由以上簡單的分析可得知本文所提出的方法中智慧卡並不需要作大量的運算，可節省相當多的時間。

(二) 安全性分析

在簽章確認協定中，攻擊者所能得到且有幫助的資料為步驟一的(CID, s)與步驟二的(Vs, B')，假如攻擊者想透過這些資料偽造簽章，必須先解出步驟(2.1)中的 $V_s=(S_m \cdot S_t/\hat{s})^2 \bmod n=(S_m \cdot \hat{s}^d)^2 \bmod n$，破解者若要透過 B'找出 $\hat{m}$ 的值，也就是必須破解者要解出 $b_i=a_i^2 \bmod n$，而在這兩個式子中 n 為兩個大質數的乘積，因此破解者必須解出這個二次同餘的式子，但是根據 Rabin's Factorization Theorem [4]，解出這一個二次同餘式子的困難度等同於對 n 做因數分解，但到目前為止大數分解的問題仍是一個相當困難的問題，由此可知攻擊者若要解出 S_m 與 $\hat{m}$ 將會相當的困難；此外，破解者也無法從 Vs 與 B'偽造出新的簽章，因為每次協定進行時伺服器都會產生亂數 s，而每次的 V_s 與 B'都將隨著 s 的不同而改變，因此攻擊者也無法從中偽造出假的簽章。

Yvo Desmedt 與 Moti Yung 針對 Chaum 的無爭議簽章提出了幾種攻擊，在他們的攻擊法中，攻擊者必須要先拿到簽章者的簽章，然而這類的攻擊法在我們所提出的系統中並不適用，因為在我們的系統中所有的簽章都是被保護於智慧卡中，而智慧卡都具有偵測破壞電路(Tamper Detection and Zeroization)，當智慧卡受到破壞時會在極的短時間內消除晶片中所有的機密資料。

六、結論與討論

在本篇論文中，我們針對特定的環境將智慧卡導入無爭議簽章，並設計了一個系統，然而這個系統仍然有許多值得加強的地方，例如在伺服器輔助運算的部分，我們並沒有加入資料的驗證，因此智慧卡無法檢驗伺服器所協助計算的結果是否有遭受竄改，而在無爭議簽章的協定部分，其實還可以加入使用者驗證的功能以強化系統的安全性，例如[12]的方法就可以直接套用到我們的系統之中。

七、參考文獻

[1] C.H. Lin and C.C. Chang: "Server-Aided Computation Protocol for RSA Enciphering Algorithm," International Journal of Computer Mathematics Vol. 53, 1994, pp. 149-155.

[2] D. Chaum and H. Van Antwerpen: "Undeniable Signatures," Advances in Cryptology- Crypt'89, August 22-24, 1989, pp.212-216.

[3] D. Chaum: "Zero-Knowledge Undeniable Signature," Advances in Cryptology-Eurocrypt'90, Springer Verlag, 1990, pp. 458-464.

[4] E. Kranakis, Primality and Cryptography, Wiley-Teubner series in computer science, John Wiley & Sons Ltd, New York, 1986, pp.149-152.

[5] Ivan Damgård, Torben P. Pedersen: "New

Convertible Undeniable Signature Schemes," Advances Cryptology: Workshop on the Theory and Application of Cryptographic Techniques, EUROCRYPT '96 (Zaragoza, Spain, May 12-16, 1996). LNCS; Springer-Verlag, (1070): 372-386, 1996.

[6] J. Boyar, D. Chaum, I. Damgard, T. Pedersen, "Convertible Undeniable Signature," Lecture Notes in Computer Science 537, Advances in Cryptology: Proc. Crypto ’90, Springer Verlag, 1991, pp.189-205.

[7] M. Jakobsson: "Blackmailing Using Undeniable Signatures," Lecture Notes in Computer Science 950, Advances in Cryptology-Eurocrypt’94, Springer Verlag, 1995, pp. 425-427

[8] M. Jakobsson: "Designated Verifier Proofs or Making Proofs of Knowledge Non-transferable," manuscript, 1995, available at:

http://www-cse.ucsd.edu/users/markus

[9] R.Gennaro, H.Krawczyk and T.Rabin: "RSA-Based Undeniable Signatures," Preliminary version in proceedings of CRYPTO'97, Springer-Verlag, LNCS 1294, pp.132-149.

[10] S. Kawamura and A. Shimbo: "Performance Analysis of Server-Aided Secret Computation Protocols for the RSA Cryptosystem," The Transactions of The Institute of Electronics, Information and Communication Engineers IEICE, Vol. E73, NO. 7, 1990, pp.1073-1080.

[11] T. Matsumoto, K. Kato and H. Imai: "Speeding up Secret Computations with Insecure Auxiliary Devices," Crypto ’88, Lecture Note on Computer Sciences 403, Springer-Verlag, Berlin 1990, pp.497-506.

[12] Wen-Her Yang and Shiuh-Pyng Shieh: "Password Authentication Schemes with Smart Card," Computers & Security, Vol. 18, No. 8, pp.727-733, 1999

[13] Y. Desmedt, M. Yung: "Weaknesses of Undeniable Signature Schemes," Lecture Notes in Computer Science547, Advances in Cryptology-Eurocrypt’91, Springer Verlag, 1992, pp.205-220.