植基於橢圓曲線之隱藏內容並存簽章方法設計
Design Encrypted Message of Concurrent Signature Method Based on ECC Technology

蘇品長
國防大學資訊管理學系
Pin-Chang Su
Department of Information Management, NDU
Email: spc.cg@msa.hinet.net

張鈞富
國防大學資訊管理學系
Chun-Fu Chang
Department of Information Management, NDU
Email: kh891425@hotmail.com

摘　要

鑑於通訊網路及電腦計算技術進步，電子商務應用日益蓬勃發展，電子商務系統的安全性及交易之公平性成為研究的關鍵問題。公平交換數位簽章架構在電子商務應用上是一個廣泛研究的議題，尤其是由 Asokan 等學者在 1998 年首先提出的樂觀公平簽章(Fair Signature)後，許多研究提出實現離線之可信任第三方的公平交換架構。Chen 等學者巧妙地觀察出在許多應用中公平交換簽章架構是非必要的，因為他們發現有一種機制可以更自然的解決爭端而不需要可信任的第三方(Trusted Third Party, TTP)的參與，就是並存簽章(Concurrent Signature)的概念。並存簽章是公平交換架構的新想法，不需要 TTP 的參與，且交易雙方不需要多次交互溝通，所以與傳統的公平交換架構相較之下，並存簽章更有效率。但是，有文獻指出先前所提出的並存簽章概念並不完全安全，會遭受到訊息取代攻擊。另外，並存簽章對於擁有 keystone 的一方有些許的優勢，因此有學者提出有責任的(Accountability)並存簽章架構，使任意第三者可利用演算法驗證簽章真偽。若要破解並存簽章，破密者將遭遇解離散對數(Discrete Logarithm Problem, DLP)難題。蒐整相關文獻記載並存簽章的方法，訊息的傳遞多以明文的形式傳送，且通訊前雙方已知對方之簽章訊息，此類協定容易導致訊息內容被惡意的第三者知悉，且與電子商務的應用流程不相符合。本研究提出植基於橢圓曲線密碼系統(Elliptic Curve Cryptosystem, ECC)的並存簽密演算法，達成隱藏內容後，先交換訊息再執行簽章步驟；本研究的特色為導入有責任的並存簽章架構，提出更安全且適用於電子商務應用的演算法。

關鍵詞：公平簽章、橢圓曲線密碼系統、並存簽章。

一、緒論

公平交換數位簽章此議題，是密碼學中的基本問題，在電子商務應用上是一個廣泛研究的議題。在[2]首先提出的樂觀(optimistic)公平簽章，其主要概念為，交換的雙方中，其中一方不能讓另一方無限期的等待接收資料訊息，必須在一定的時間限制內進行公平交換，此交換的傳輸方式為非同步的。當交易的雙方中，有某一方為惡意者，則可能造成誠實的一方無限期的等待資料訊息的接收，或是沒有收到所需要的資料訊息，亦或收到錯誤的資料訊息。此時誠實的一方會透過可信任第三方(Trusted Third Party, TTP)來執行兩個子程序來解決雙方所產生的紛爭，使得雙方皆獲得所需的資料訊息，或是雙方皆無法獲得達成公平交換的目的。樂觀公平交換採用離線型可信任第三方(off-line TTP)，若在交易過程中有爭端產生，此時才需要TTP來協助處理紛爭，許多研究提出實現離線之可信任第三方的公平交換架構[3]。Chen等學者巧妙地觀察出在許多應用中公平交換簽章架構是非必要的，因為他們發現有一種機制可以更自然的解決爭端而不需要TTP的參與，就是並存簽章(Concurrent Signature)的概念[6]。並存簽章允許簽章的雙方產生並交換模糊簽章，任意第三者無法得知模糊簽章的原始簽章者，直到額外的資料片段(keystone)被其中一方公佈後，任意第三方即可利用此資訊驗證模糊簽章是何者所簽署的。

為了強化Chen等學者提出的並存簽章架構的匿名性，Susilo等學者提出了完美並存簽章協定(Perfect Concurrent Signature, PCS)[11]。不幸的，Lin等觀察出，在PCS協定中，初始簽章者獨立產生兩個keystone，這樣會使初始簽章者可以自由綁定不同的模糊簽章，而不是綁定原本要傳送給匹配簽章者的正確簽章，此舉將造成對匹配簽章者的不公平性，所以提出了改進的完美並存簽章(Improved Perfect Concurrent Signature, iPCS)[9]。PCS和iPCS的不同在於iPCS協定中之初始簽章者及匹配簽章者可分別產生一個keystone，但PCS則只有初始簽章者參與keystone的產生。

之後，Chow and Susilo參考iPCS之架構而發展出基於身分認證之完美並存簽章[7]。另外，像是非對稱並存簽章[10]、三方並存簽章[12]及多方並存簽章[13]也隨後被相繼提出。

然而，在先前的文獻[4、5、6、8、9、10、11、12]中被觀察到有弱點易遭受到以下之攻擊：在簽章協定中，任意一方皆可產生很多模糊簽章，每個模糊簽章中所包含的訊息皆不同，這些訊息可同時被相同的keystone綁定，但這些假冒的訊息卻不是當初所應傳送之簽章訊息，此稱為訊息取代攻擊

(Message Substitute Attack)。

基於這些觀察，一個安全上的特性在文獻中提出有責任(Accountability)的特性[14]。有責任的特性是指任意第三方在keystone被公佈後，透過並存簽章協定中之VERIFY演算法驗證無誤後，即可確定此模糊簽章是唯一的。但是，因為[14]中所提出的改進並未達到此特性，因此[1]提出將Alice與Bob所要交換之訊息皆加入綁定訊息(keystone fix)中，進而達到完全有責任的特性，但是發現先前提出之方法在訊息的傳送方式是以明文的形式傳送，且通訊前雙方已知對方之簽章訊息，將導致訊息內容可被惡意的第三者竊取且與實際電子商務流程不符。因此，本篇論文將明文訊息以橢圓曲線密碼系統(Elliptic Curve Cryptosystem，ECC)加密，達成隱藏內容後，先交換訊息再執行簽章步驟；另將橢圓曲線數位簽章演算法(Elliptic Curve Digital Signature Algorithm，ECDSA)導入完全有責任的並存簽章架構，改進後達到先前文獻主張之安全性且相同安全程度下金鑰長度較短，破密者會遭受橢圓曲線離散對數問題之安全性。

在第二章中我們回顧橢圓曲線簽章演算法及有責任性的並存簽章架構。之後，我們在第三章中提出改進之協定並在第四章提出安全性及效益分析。最後，第五章是本論文之結論。

二、文獻探討

本章首先介紹橢圓曲線簽章演算法的概念，接著回顧有責任的完美並存簽章改進後之演算法。

(一)橢圓曲線簽章演算法

橢圓曲線數位簽章演算法(Elliptic Curve Digital Signature Algorithm，ECDSA)，是一種標準的橢圓曲線數位簽章演算法，方法總共分為三部分，系統初始階段、簽章階段及驗證簽章階段，各階段分述如下：

1.系統初始階段

在有限域F_p上選取一條安全的橢圓曲線$E(F_p)$，(p 為一個 160bit 以上之大質數)並在$E(F_p)$上選取一階數(order)為 n 的基點 G，使得 nG=O，其中 O 為此橢圓曲線之無窮遠點。

簽章者隨機選擇一整數n_A當成私鑰，其中n_A介於[1,n-1]計算簽章者公鑰$P_A = n_A G$，將(E, G, P_A)公開。

2.簽章階段

假設欲簽章之訊息為 m，隨機選擇一整數 r 介於[1,n-1]，計算$R = rG = (x, y)$、計算$s = r^{-1}(h(m) - n_A x) \bmod n$，將訊息 m 的簽章(m,s,R)傳給收訊方。

3.驗證簽章階段

取得簽章者之公鑰及系統公開訊息(E, G, P_A)，檢驗 r 及 s 是否介於[1,n-1]，若不在範圍內則否定其簽章，計算$V_1 = x \cdot P_A + s \cdot R$及$V_2 = h(m) \cdot G$，若$V_1 = V_2$則驗收，否則拒絕。

(二)有責任的並存簽章架構

1. 並存簽章架構由 SETUP、ASIGN、AVERIFY、VERIFY 四種演算法所組成，描述如後：

(1)SETUP：輸入一個安全參數 l，首先隨機產生兩個大質數 p、q，其中$q \mid (p-1)$，及階數(order)為 q 的生成子(generator) g，其中$g \in Z_p$，接著選擇一個雜湊函數$h: (0,1)^* \to Z_q$。再來，設置訊息空間 M、keystone 空間 K 以及 keystone fix 空間 F。其中$M = K = \{0,1\}^*$、$F = Z_q^*$。此外假設$(x_A, y_A = g^{x_A} \bmod p)$與$(x_B, y_B = g^{x_B} \bmod p)$分別為 Alice 和 Bob 的私鑰和公鑰。

(2)ASIGN：藉由輸入(y_A, y_B, x_A, s, m)等資訊輸出模糊簽章$\sigma = (c, s', s)$，其中y_A、y_B是 Alice 和 Bob 的公鑰$y_A \neq y_B$、x_A是 Alice 的私鑰、$s = h(k)$，$s \in F$，keystone $k \in Z_q$，$m \in M$是被簽署的訊息。此演算法將做下列參數之運算：

- 選擇一個隨機亂數$x \in Z_q$。
- 計算$c = h(m, g^x y_B^s \bmod p)$。
- 計算$s' = (x - c) \cdot x_A^{-1} \bmod q$。
- 輸出匿名的模糊簽章$\sigma = (c, s', s)$。

(3)AVERIFY：輸入(σ, y_A, y_B, m)，驗證$c = h(m, g^{s'} y_A^{s'} y_B^s \bmod p)$如果成立輸出接受(accept)，否則輸出拒絕(reject)。

(4)VERIFY：此演算法輸入(k, S)，$k \in K$，$S = (\sigma, y_A, y_B, m)$，透過驗證 AVERIFY(S)如果輸出為 accept，且 keystone k 透過驗證驗算法為正確的，則 VERIFY 亦輸出 accept，否則輸出 reject。

2. 有責任的並存簽章協定之改進

假設已執行過 SETUP 演算法，且初始簽章者 Alice 與匹配簽章者 Bob 想要交換他們對訊息m_A與m_B所做的簽章。

(1)Alice 執行如下：

- 隨機選擇一個 keystone $k \in K$，且設置$s_k = h(k, m_A, m_B)$。
- 執行$\sigma_A \leftarrow ASIGN(y_A, y_B, x_A, s_k, m_A)$，其$\sigma_A = (c, s_1, s_2)$中。
- 送出模糊簽章(σ_A, m_A)給 Bob。

(2)收到(σ_A, m_A)，Bob 確認 AVERIFY$(\sigma_A, y_A, y_B, m_A)$=accept，如果輸出為 reject，則 Bob 中止動作，否則，Bob 執行以下動作：

- 隨機選擇一個$t \in Z_q$，計算

$\hat{t} = y_B{}^{r} \bmod p$。

- 計算$r = y_A{}^{x_B r}$，且計算 $k' = h(r, m_A, m_B) \bmod q$
- 設置$s_1' = s_1 + h(k') \bmod q$。
- 執行 $\sigma_B \leftarrow \mathrm{ASIGN}(y_A, y_B, x_B, s_1{}^{t}, m_B)$，其中$\sigma_B = (c', s_1{}^{t}, s_2{}^{t})$
- 將簽章$(\sigma_B = (c', s_1', s_2'), m_B, \hat{t})$傳送給 Alice。

(3)Alice 接收到$(\sigma_B = (c', s_1', s_2'), m_B, \hat{t})$執行如下：

- 計算$r = \hat{t}^{x_A} \bmod p$，且計算 $k' = h(r, m_A, m_B) \bmod q$。
- 驗證$s_1' = s_1 + h(k') \bmod q$是否成立，如果不成立則中止動作。
- 確認 AVERIFY$(\sigma_B, y_A, y_B, m_B) \stackrel{?}{=}$ accept，如果沒通過驗證，則 Alice 中止動作。
- 如果通過 AVERIFY 驗證，則 Alice 公佈 keystone(k,k')，使得兩個模糊簽章同時綁定且生效。

(4)VERIFY 演算法：一旦 keystone(k,k')公佈後，任意驗證者皆可藉由驗證下列式子驗證簽章是分別由 Alice 與 Bob 所簽署：

- $s_2 = h(k, m_A, m_B)$ 且 $s_1' = s_1 + h(k') \bmod q$
- AVERIFY$(\sigma_A, y_A, y_B, m_A)$ = accept
- AVERIFY$(\sigma_B, y_A, y_B, m_B)$ = accept

三、演算法改進

第二章所提到改進 i2PCS1 的概念可成功克服訊息取代攻擊的弱點，本研究提出基於 ECDSA 之並存簽章改進方式，使簽章之安全性及速度皆較原本的簽章架構更安全且快速。

本研究方法整體運作循序圖如圖 3-1 所示：

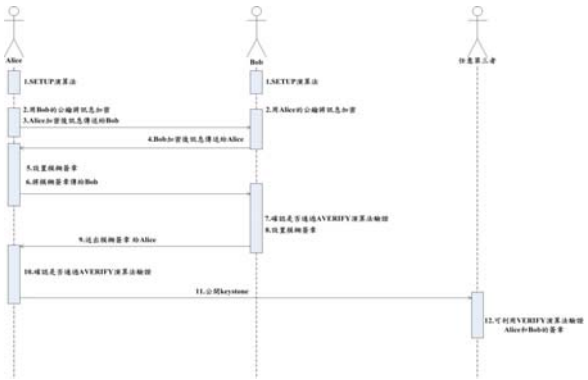

圖 3-1 整體運作循序圖

假定：

Alice 與 Bob 要進行電子交易並交換簽章，簽章之訊息為雙方交易的內容，其簽章的訊息為m_A、m_B。例如：Alice 在網路上跟 Bob 買了一件商品，則 Alice 利用訊息m_A產生模糊簽章，並將此簽章傳送給 Bob，當 Bob 收到簽章後進行驗證，確認無誤後即利用m_B產生一個模糊簽章傳送給 Alice。待 Alice 公佈 keystone 之後，使得兩個簽章同時生效，使交易雙方得到正確的訊息。

改進後之步驟如下：

(一)執行 SETUP 演算法

1.從$y^2 = x^3 + ax + b \pmod p$，$4a^3 + 27b^2 \neq 0$建立$E_p(a, b)$，選一個階數(order)極大的 n 基點$G(x,y)$在$E_p(a, b)$上$n \times G = O$。

2.選擇一個赫式函數(hash function)$h = \{0,1\}^* \to Z_n$。

3.Alice 選擇$n_A \in Z_n$當私鑰，並產生公鑰$P_A = n_A \times G$；Bob 選擇$n_B \in Z_n$當作私鑰，而公鑰是 $P_B = n_B \times G$。

4.設置訊息空間 M、keystone 空間 K、簽章空間 S 與 keystone fix 空間 F。其中$M = K = \{0,1\}^*$與$F = Z_n{}^*$。

(二)Alice 和 Bob 交換雙方訊息：

1.Alice 傳送密文訊息給 Bob：

(1)Alice 利用$H_{m2p}()$函數將欲傳送之訊息編碼成橢圓曲線之點，即 $J_A = H_{m2p}(m_A) = (x_J, y_J)$

(2)任選一個整數$f \in Z_n{}^*$，然後計算密文(L_1, L_2)，其中$L_1 = f \cdot G$，$Y = (x_Q, y_Q) = f \cdot P_B$，

$L_2 = (x_J, y_J) = (x_Q \cdot x_J \bmod n,\ y_Q \cdot y_J \bmod n)$

(3)Alice 將密文(L_1, L_2)傳送給 Bob

(4)Bob 解密程序如下：

- 計算$Z = (x_Z, y_Z) = n_B \cdot L_1$
- 計算明文 $J_A = (x_1 \cdot x_2{}^{-1} \bmod n,\ y_1 \cdot y_2{}^{-1} \bmod n)$

2.Bob 傳送密文訊息給 Alice 計算步驟同上。

(三)初始簽署者 Alice 執行如下：

1.隨機選擇一個keystone $k \in K$，且設置 $s_2 = h(k, m_A, m_B)$

2.執行 $\sigma_A \leftarrow ASIGN(P_A, P_B, n_A, s_2, m_A)$

(1)選擇一個隨機亂數 $\alpha \in Z_n$

(2)計算 $T_1(x_1, y_1) = \alpha G + s_2 P_B$

(3)計算 $c = h(m_A, x_1)$

(4)計算 $s_1 = (\alpha - c) n_A^{-1}$

(5)輸出 $\sigma_A = (c, s_1, s_2)$

3.送出模糊簽章給匹配簽章者 Bob

(四)Bob 收到模糊簽章後執行如下：

1.計算 $T_2(x_2, y_2) = cG + s_1 P_A + s_2 P_B$

2.確認是否通過 AVERIFY 演算法：
$c = h(m_A, x_2)$

如果驗證失敗，則 Bob 終止協議，否則，執行如下：

(1)隨機選擇一個 $t \in Z_n$，計算 $T_3(x_3, y_3) = tP_B$，且 $t' = x_3$

(2)計算 $T_4(x_4, y_4) = (n_B t) P_A$，$r = x_4$

(3)計算 $k' = h(r, m_A, m_B)$

(4)設置 $s_1' = s_1 + h(k') \bmod n$

(5)執行 $\sigma_B \leftarrow ASIGN(P_B, P_A, n_B, s_1', m_B)$

- 選擇一個隨機亂數 $\beta \in Z_n$
- 算 $T_5(x_5, y_5) = \beta G + s_1' P_A$，$c' = h(m_B, x_5)$
- 算 $s_2' = (\beta - c') n_B^{-1}$
- 出 $\sigma_B = (c', s_1', s_2')$

(6)送出簽章
$(\sigma_B = (c', s_1', s_2'), P_A, P_B, t')$ 給 Alice

(五)Alice 收到後，執行如下：

1.計算 $r = n_1 t'$，且計算 $k' = h(r, m_A, m_B)$

2.測試 $s_2' = s_2 + h(k') \bmod n$ 是否成立，如果不成立，Alice 終止協定

3.計算 $T_6(x_6, y_6) = c'G + s_1' P_A + s_2' P_B$

4.確認是否通過 AVERIFY 演算法：
$c' = h(m_B, x_6)$

5.如果通過 AVERIFY 驗證，則 Alice 公佈 $keystone(k, k')$，使得兩個簽章
$(\sigma_A = (c, s_1, s_2), P_A, P_B, m_A)$ 與
$(\sigma_B = (c', s_1', s_2'), P_A, P_B, m_B)$ 同時綁定且生效。

(六)VERIFY 演算法：$keystone(k, k')$ 公佈後，若 $s_2 = h(k, m_A, m_B)$ 且 $s_1' = s_1 + h(k') \bmod n$ 成立，則任意的驗證者則可以確認模糊簽章 $(\sigma_A = (c, s_1, s_2))$ 與 $(\sigma_B = (c', s_1', s_2'))$ 是分別由 Alice 或是 Bob 所簽署的簽章。

四、安全性與效益分析

(一)安全性分析

本研究所提之加密機制，其安全性主要植基於橢圓曲線離散對數難題(ECDLP)、非對稱加密方式、並存簽章架構、單向雜湊函數，可達到 ISO 組織所提之資訊安全管理需求 ISO27001:2005，其中包含[6]所提出並存簽章必備之正確性、模糊性、不可偽造性、公平性、機密性、鑑別性與不可否認性等安全需求。以下我們針對安全性分析來進行探討：

1.正確性(Correctness)

根據[6]架構若經過 ASIGN 演算法，且 $AVERIFY(\sigma_A, y_A, y_B, m_A) = accept$，$AVERIFY(\sigma_B, y_A, y_B, m_B) = accept$ 皆成立，表示訊息及模糊簽章皆正確。

本研究除加入訊息交換階段，在模糊簽章及驗證之步驟皆與之前文獻相同，惟改為橢圓曲線之架構，故符合正確性。

2.模糊性(Ambiguity)

並存簽章架構下的模糊簽章，任意第三者無法得知模糊簽章的原始簽章者，直到 keystone 被其中一方公佈後，任意第三方即可利用此資訊驗證模糊簽章是何者所簽署的。

本研究(三)之模糊簽章，因交易雙方之訊息即 m_A, m_B 是經過單向無碰撞雜湊函數計算，可使第三者無法推測出簽章是由哪一方所簽署，故達到簽章的模糊性。

3.不可偽造性(Unforgeability)

不可偽造性指的是若攻擊者試圖偽造文件或簽章，任何人能夠經由參數驗證得知文件或簽章是否偽造。

本研究中，假若 Bob 想對其他的訊息 $\bar{m}_B$ 產生另一個模糊簽章，並且在 keystone 公佈後此假冒的簽章 $\bar{\sigma}_B$ 亦會與 σ_A 同時綁定與生效，Bob 將會失敗。因為，Alice 與 Bob 欲交換的訊息 m_A、m_B，已經被綁進 keystone fix 中，所以 Bob 所簽署出假的簽章 $(\bar{\sigma}_B, \bar{m}_B)$，在 keystone($k$, k')公佈之後，任何人皆可去執行 VERIFY 演算法來驗證簽章的正確性。

4.公平性(Fairness)

並存簽章之公平性指的是當協定完成後，可以讓任意驗證者知道該簽章是否為 Alice 及 Bob 所簽署。

本研究中假設 Bob 為欺騙的一方，因雙方訊息皆已綁定在 keystone fix 中，所以 Bob 所簽署的假簽章，在 keystone(k, k')公佈後，驗證者可執行 VERIFY 演算法來驗證簽章的正確性。

5.機密性(Confidentiality)

機密性指的是資料不得被未經授權之個人、實體或程序所取得或揭露的特性。

本研究方法中，傳輸訊息皆使用橢圓曲線加密法加密，如被惡意第三者竊取了密文(L_1, L_2)，但在破解$L_1 = j \cdot G$，因需從$j \cdot G$之中推算出j的數值，破密者將面臨橢圓曲線離散對數的難題。

6.不可否認性(Non-repudiation)

不可否認性指的是對一已發生之行動或事件之證明，使該行動或事件往後不能被否認的能力。不論是傳送方或接收方皆不能否認訊息曾被傳送的事實，保證任一個網路節點不能否認其所發送出去的訊息及不能否認它以前傳送訊息的行為。

本研究當並存簽章協定完成且公佈 keystone(k, k')後，任意第三者皆可藉由執行 VERIFY 演算法來驗證簽章是由哪一方所簽署，故可達成不可否認之特性。

(二)效益分析

依本研究目前的研究結果所使用的演算法，與王智弘等學者[1]之有責任的並存簽章演算法比較分析如表 4-1，比較表可看出本研究方法優於其他系統。

表 4-1 本研究與有責任的並存簽章演算法比較表

比較項目 / 演算法	金鑰長度	安全性	責任性	隱藏內容
有責任的並存簽章演算法	較長	較低	有	無
本研究方法	較短	較高	有	有

五、結論

由於之前文獻所提出的並存簽章架構都存在訊息取代攻擊的風險，在此篇論文中，我們發現有責任的並存簽章架構為了改進此弱點，而提出了一個新的架構來預防此攻擊。然而，我們發現有責任的並存簽章修改後的方式流程與實際電子商務流程不符合，且訊息以明文的方式傳送，存在遭竊取的風險。因此，我們提出了一個隱藏內容的改進方法來解決此問題。我們將明文訊息以接收者的公鑰加密後傳送給接收者，接收者再以私鑰解密，雙方訊息交換完成後再進行後續步驟，有效解決訊息安全性的問題。另外，我們將演算法以橢圓曲線的架構加以改進，使相同安全程度下金鑰長度較短，破密者會遭受橢圓曲線離散對數問題。此架構在 keystone 公佈之前，任意的第三方無法得知簽章是由哪一方所簽署，達到模糊的效果，另外，也達到了有責任的特性，將雙方的訊息皆予以綁定，使得交換的雙方都能達到公平性。

綜整本研究達成之貢獻如下：

一、利用橢圓曲線金鑰長度短、處理速度快且安全性高的特性，使系統達到更快且更安全的運作。

二、將明文訊息加密達成隱藏內容後，先交換訊息再執行簽章步驟，可改善之前文獻與電子商務交易流程不相符合之處。

三、以並存簽章方法進行簽章交換，不需要 TTP 的參與，且交易雙方不需要多次交互溝通，所以與傳統的公平交換架構相較之下，並存簽章更有效率。

參考文獻

[1]王智弘、陳昭權、吳嘉峻，2010，並存簽章在公平交換的使用與改進。*資訊安全通訊*，第16卷・第三期：60~71頁。

[2]N. Asokan, V. Shoup, and M. Waidner, “Optimistic fair exchange of signatures,” *In:EUROCRYPT’98*, LNCS (1403), pp. 591-606, 1998.

[3]N. Asokan, V. Shoup, and M. Waidner, “Optimistic fair exchange of digital signatures,” *IEEE Journal on SelectedAreas in Communication* (18:4), pp. 593-610, 2000.

[4]F. Bao, R. H. Deng, and W. Mao, “Efficient and practical fair exchange protocols with off-line TTP,” *IEEE Symposium on Security and Privacy*, pp. 77-85, 1998.

[5]D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and variably encrypted signatures from bilinear maps,” *In: EUROCRYPT’03*, LNCS (2656), pp. 416-432, 2003.

[6]L. Chen, C. Kudla, and K. G. Paterson, “Concurrent signatures,” *In: Eurocrypt ’04*, LNCS (3027), pp.287-305, 2004.

[7]S. Chow and W. Susilo, “Generic construction of (identity-based) perfect concurrent signatures,” *7th International Conference on Information and Communications Security*, LNCS (3783) , pp. 194-206, 2005.

[8]J. Garay, M. Jakobsson, and P. MacKenzie, “Abuse-free optimistic contract signing,” *In: CRYPTO’99*, LNCS (1666), pp. 449-466, 1999.

[9]W. G. Lin, B. Feng, and Z. J. Ying, “The fairness

of perfect concurrent signatures," *In: ICICS'06*, LNCS (4307), pp. 435-451, 2006.

[10]k. Nguyen, "Asymmetric Concurrent Signatures," *ICICS2005*, LNCS (3783), pp. 181-193, 2005.

[11]W. Susilo, Y. Mu, and F. Zhang, "Perfect concurrent signature schemes," *In: ICICS '04*, LNCS (3269), pp. 14-26, 2004.

[12]W. Susilo, and Y. Mu, "Tripartite concurrent signatures," *IFIP/SEC'05*, pp. 425–441, 2005.

[13]D. Tonien, W. Susilo, and R. S. Naini, "Multi-party Concurrent Signatures," *ISC06*, LNCS (4176), pp. 131-145, 2006.

[14]L. Yunteng, H. Dake, and L. Xianhui, *Accountability of Perfect Concurrent Signature*, International Conference on Computer and Electrical Engineering, 2008.

[15]Y. Zhang and X. Wang, *Message Substitute Attack on Concurrent Signatures Protocol and its Improvement*, International Symposium on Electronic Commerce and Security, 2008.