智慧型安全電子付款系統之研究
Designing Intelligent Secure Electronic Payment Systems

曹偉駿* 廖仁億 劉彥含
大葉大學資訊管理研究所
彰化縣 51505 大村鄉山腳路 112 號
*E-mail: wjtsaur@mail.dyu.edu.tw

摘要

目前電子付款系統在實務的設計上，多採數位憑證為基礎的方式來處理相關的安全付款事宜，但是此作法有一個很重要的先決條件，那就是系統認證中心須是誠實的且必須保護金鑰目錄，另外還需額外耗費驗證系統憑證中心之簽章的時間。本論文所探討的是智慧型安全電子付款技術，故除了安全層級的顧慮外，還必須兼顧安全機制運算上的便捷與效率。因此，本論文發展出一套以橢圓曲線密碼系統為基礎的具自我認證公開金鑰密碼系統，並以此自我認證公開金鑰密碼系統發展出交談金鑰、數位簽章及盲簽章等安全機制，且將這些技術實際應用在較具傳統方式的電子現金型付款系統，藉以提升這類付款機制的安全與效率，使即時性的安全電子付款成為可行的方案。總之，本論文目的在於發展出有效率之自我認證為基礎的安全機制，藉此可使電子商務交易之安全付款機制更臻完備且更切實際，以提昇使用者對使用電子商務付款服務的信心。

關鍵字： 電子付款系統、橢圓曲線密碼系統、自我認證公開金鑰密碼系統、盲簽章

一、前言

隨著電子商務的快速發展，電子付款系統的需求迫切性與日俱增。電子商務所提供的服務種類繁多，而每一種類型的電子商務服務其所使用的電子付款機制也不盡相同。在 E-cash [24]、Payme [18]、NetCash [14]、SET [13]、iKP [2]、NetBill [6]、NetCheque [17]及 CyberCash [7]等電子付款系統安全機制中，密碼機制的主要運算方式都是大指數運算，如 RSA [21]、ElGamal [8]與 Schnorr [22]。毋庸置疑的，這些密碼機制提供了電子付款系統所需的安全性，但相對的，在計算效率上則須花費較多的儲存空間。由 Victor Miller [15]與 Nei Koblitz [12]兩位學者所提出來的橢圓曲線密碼系統(Elliptic Curve Cryptosystem；ECC)，提供較小金鑰達到與上述密碼機制相同的安全等級，例如 ECC 只需要 160 位元就可以達到與 RSA 1024 位元相同的安全等級。ECC 因所需金鑰較小，故可節省儲存空間，其安全性主要是建立在解橢圓曲線離散對數的問題上。

目前在實務的設計上，多採行電子憑證為基礎(Certificate-based)的方式來處理相關的安全付款事宜(例如 SET 安全機制)，但是此作法有一個很重要的先決條件，那就是系統認證中心須是誠實而且必須保護金鑰目錄，另外在使用時還需額外增加驗證系統認證中心簽章的步驟。在現實的環境中，其實我們並不能絕對認定系統憑證中心一定是誠實的，或者其實系統憑證中心也是有機會被駭客入侵的，故發展自我認證(Self-certified) [9]的機制的確有其必要性。所謂的自我認證是指交談的雙方僅需要靠雙方傳送一些公開的資訊，即可達成雙方身分的確認，而不需透過第三者來做保證或協調。本論文所欲設計的智慧型安全電子付款系統，故除了安全層級的顧慮外，還必須兼顧安全機制運算上的便捷與效率。因為橢圓曲線公開金鑰密碼系統較現存的其它公開金鑰密碼系統可用較少之位元數達到相同的安全度。因此，本論文欲發展出一套以橢圓曲線密碼系統為基礎的自我認證公開金鑰密碼系統，並以此自我認證公開金鑰密碼系統發展盲簽章安全機制，並將盲簽章技術應用在類似傳統方式的電子現金型付款系統，藉以提升安全付款機制的效率。此自我認證為基礎的安全機制，可使電子商務交易之安全付款機制更臻完備且更切實際，亦即使電子商務服務更具保障，以提昇使用者對電子商務服務上安全的信心。

二、電子付款系統之相關密碼技術

目前，針對身分認證方式的不同，存在三種不同的公開金鑰密碼系統，分別是身份為基礎的公開金鑰密碼系統(Identity-based public key cryptosystem)、憑證為基礎的公開金鑰密碼系統(Certificate-based public key cryptosystem)以及自我認證的公開金鑰密碼系統(Self-certified public key cryptosystem)。現階段的電子商務應用上，多是以憑證基礎的方式來作身分確認。以下簡單地說明這三種公開金

鑰密碼系統與橢圓曲線密碼系統的特性。

[身分為基礎的公開金鑰密碼系統]

身分為基礎的公開金鑰密碼系統由 Shamir [23]在 1984 年所提出。系統的特色是系統中心不需要額外儲存使用者的 PK，使用者的 PK 即是本身的身分(ID)，而驗證 PK 的有效性時，以使用者的 SK 來驗證，亦即 PK = ID，SK = G (G表示系統中心的簽署保證)。因此，此種公開金鑰密碼系統所要儲存的參數大為減少。然而，身分為基礎的公開金鑰密碼系統中，存在著兩個很大的問題：

1. 使用者的密鑰和公鑰都是由系統中心計算出來，系統中心知道每一個使用者的密鑰，系統中心可以假扮任何一位使用者而不被發現。因此，當系統中心被入侵之後，每一個使用者的權益將遭受莫大的傷害。
2. 系統中心必須透過一個安全的秘密管道傳送密鑰給使用者，這是一個可行的假設，不過在網際網路上，安全的秘密管道是不容易建立的，因此這個假設較不合理。

[憑證為基礎的公開金鑰密碼系統]

此公開金鑰密碼系統中除了有公鑰 PK 及密鑰 SK 外，為了能夠在網路上確認每個使用者的身分，還需要兩個重要的參數，即個人身分(ID)及系統中心的簽署保證(G)。SK 是由個人保管，而 PK 是完全的公開，很容易被仿冒或篡改。系統中心為了要保證 PK 的正確性，PK、ID 及 G 三個參數必須被儲存在系統中心，其中 G 參數是被用來保證 PK 的正確性[9]。但是，憑證為基礎的公開金鑰密碼系統]還是有兩個缺點：

1. 系統中心自行計算 G 值來保證使用者 PK 的正確性，一旦系統中心不誠實或被入侵成功，可以很容易地偽造一個完全不存在的使用者。
2. 系統中心計算 G 值來保證 PK 的正確性，故要多儲存一個參數增加系統中心的負擔。此外，為驗證 PK 正確與否，必須多一道驗證系統中心之簽署文 G 的步驟。

現今許多的電子商務交易都是使用憑證為基礎的公開金鑰密碼系統，必須由公正第三單位來簽發憑證，除了面臨上述的兩種缺點外，還需要維護金鑰目錄中參數的一致性與增加儲存空間，而這正是目前其金鑰管理所面臨的問題。

[自我認證的公開金鑰密碼系統]

自我認證的公開金鑰密碼系統由 Girault [9]在 1991 年提出，其中 Girault 提出三個安全等級：

Level 1：系統中心知道所有使用者的密鑰，且在任何時候都可以假扮任何一位使用者而不會被發現，身分為基礎的公開金鑰密碼系統明顯地只達到這個等級。

Level 2：系統中心不知道使用者的密鑰，但是可以假造一個不合法的使用者而不會被發現，憑證為基礎的公開金鑰密碼系統就是屬於這一個層次。例如：系統中心因自行假造一個不合法使用者的 G 值，讓此使用者是合法的，而一般的使用者是無法察覺出來的。

Level 3：系統中心任何的欺騙行為都會被察覺出來，而 Girault 所提出的自我認證公開金鑰密碼系統就是屬於此等級。

自我認證公開金鑰密碼系統，其優點在於使用者的 SK 是自行選定，系統中心無法由傳送的資料當中獲得使用者的 SK；而且在計算使用者 PK 時，是由使用者與系統中心一起合作計算出來的，當系統中心把 PK 回傳給使用者時，使用者可以自行驗證 PK 的真偽，故系統中心無法自行為使用者產生 PK 及驗證，假如系統中心有詐騙行為的話，都將會被偵錯出來，因此達到 Level 3 的安全等級。

[橢圓曲線密碼系統]

橢圓曲線密碼系統是由 Koblitz [12]和 Miller [15]兩位學者所提出來的。在有限體 K 之下，給定橢圓曲線 E 上的兩個點 P 及 Q，當點 P 的秩(order)夠大時，要找出一個整數 x，使得 Q = x*P是很困難的，此問題稱為解橢圓曲線離散對數問題(Elliptic curve discrete logarithm problem；ECDLP)。在 RSA 與 ElGamal 系統中需要使用 1024 位元的模數，才能達到足夠的安全等級，而在 ECC 只需要使用為 160 位元的模數即可。

[盲簽章(Blind Signature)]

在數位簽章的方法中，簽章者知道他所要簽署的文件內容。假使我們要求某人簽署一文件，然而他並不知道文件的內容，這稱之為盲簽章。盲簽章的方法最早由 Chaum [4]於 1983 年所提出，主要是應用於電子付款系統中以保護客戶資料的隱密性。盲簽章必須滿足兩種特性：

1. 簽章者不知道所簽署的文件內容。
2. 日後公佈文件及文件簽名，簽章者也無法追蹤文件與盲簽章的關係。

除了 Chaum [4]於 1983 年所提出盲簽章法外，Chaum 等人 [5]也在 1988 年提出植基於 RSA 機制的盲簽章法，之後許多學者陸續提出植基於解離散對數問題(Discrete Logarithm Problem)之困難度的盲簽章法[3]，但 Harn [10]在 1995 年破解了植基於解離

散對數問題困難度之盲簽章法。另外，還有植基於 ElGamal [8]簽章機制的盲簽章法 [16]及植基於 Schnorr [22]簽章機制的盲簽章法 [20]。

因上述所提出的盲簽章法，都要使用到大指數的運算(Modular Exponentiation)，而大指數的運算是非常耗費時間的。因此，本論文所要設計的新盲簽章法是植基於 ECC based self-certified public key cryptosystem，因為 ECC 在金鑰的長度方面比其它方法來的短，但 ECC 卻可以達到與它們相同的安全等級。此外，本論文所提出之新盲簽章法於驗證盲簽章時，亦可同時驗證公開金鑰的有效性。

三、植基於 ECC 的自我認證公開金鑰密碼系統(ECC based self-certified public key cryptosystem)

金鑰分配可分為系統設定、使用者註冊及交談金鑰分配這三個階段。系統設定階段時，系統中心(以下簡稱 SA)建立本身的基本參數，並公佈公開參數出來；使用者註冊階段時，使用者將自己的身份相關資訊傳給 SA 取得公鑰及證明，並得到密鑰；在交談金鑰分配階段時，要互相溝通的使用者間，透過傳遞公開資訊就可以建立一把共享的交談金鑰，使用在傳遞資料時的加/解密。以下就各階段作詳細說明。

3.1 系統設定階段

SA 的初始化，用以建立公開及秘密參數。由 SA 計算以下參數：

1. E：橢圓線方程式 $y^2 \equiv x^3 + ax + b$，其中 $4a^3 + 27b^2 \neq 0$
2. p：大質數(使得點 B 存在)
3. q：$p-1$ 的質因數($q \mid p-1$)
4. B：秩為 q 的點，橢圓曲線上的生成數
5. X_{SA}：SA 的秘密金鑰
6. Y_{SA}：SA 的公開金鑰，亦即其中滿足 $Y_{SA} = X_{SA} B \bmod p$
7. $h()$：單向雜湊函數(One way hash function) 輸出一整數

當 SA 建立這些參數後，將$\{p, q, B, h(), Y_{SA}, E\}$公開，保留 X_{SA} 當成其密鑰。

3.2 使用者註冊階段

使用者U_i執行以下步驟向 SA 註冊以取得合法公鑰及證明，並計算其密鑰。

1. U_i 隨機選取一個數 K_i，並計算 $R_i = K_i B \pmod p$，將自己的身分相關資訊 ID_i 與 R_i 傳給 SA。
2. SA 收到 ID_i, R_i 後，隨機選取一個數 Z_i，計算以下式子：

$Y_i = R_i + Z_i B \pmod p + (Y_{ix}, Y_{iy})$, and

$W_i = X_{SA} h(ID_i, Y_{ix} \| Y_{iy}) + Z_i \pmod q$

並將 $\{Y_i, W_i\}$傳給參與者。

3. 當 U_i 收到 $\{Y_i, W_i\}$後，計算 $X_i = W_i + K_i \pmod q$，並根據下面公式驗證 Y_i 的有效性：

$X_i B = Y_i + Y_{SA} h(ID_i, Y_{ix} \| Y_{iy}) \pmod p$

若等式成立的話，X_i為U_i的秘密金鑰，Y_i為U_i的公開金鑰；否則代表Y_i在傳輸過程中被竄改或替換過。

[定理一] 使用者可自行驗證公鑰的有效性。

證明：

$$\begin{aligned} X_i B &= (W_i + K_i) B \pmod p \\ &= (X_{SA} h(ID_i, Y_{ix} \| Y_{iy}) + Z_i + Ki) B \pmod p \\ &= X_{SA} B h(ID_i, Y_{ix} \| Y_{iy}) + Z_i B + Ki B \pmod p \\ &= Y_{SA} h(ID_i, Y_{ix} \| Y_{iy}) + Z_i B + R_i \pmod p \\ &= Y_i + Y_{SA} h(ID_i, Y_{ix} \| Y_{iy}) \pmod p \end{aligned}$$

由上述證明可知，當使用者U_i收到$\{Y_i, W_i\}$可依據 $X_i B = Y_i + Y_{SA} h(ID_i, Y_{ix} \| Y_{iy}) \pmod p$ 驗證公鑰及證明的有效性。

3.3 交談金鑰分配階段

交談金鑰的分配可分為兩種方式，一是固定式交談金鑰分配，此方法在每次建立交談金鑰時，都會建立相同的交談金鑰；另一種為非固定式交談金鑰分配，此方法在每次建立交談金鑰時，都會建立不同的交談金鑰，以下就分別提出這兩種交談金鑰分配的方法。

當使用者U_i與U_j要共享一把交談金鑰時，可採用的交談金鑰分配方法如下：

[固定式交談金鑰分配]

1. U_i 將 ID_i 與 Y_i 傳給 U_j。

   U_j 將 ID_j 與 Y_j 傳給 U_i。

2. U_i 計算交談金鑰

$SK_{ij} = (Y_j + Y_{SA} h(ID_j, Y_{jx} \| Y_{jy})) X_i \pmod p$

U_j 計算交談金鑰

$SK_{ji} = (Y_i + Y_{SA} h(ID_i, Y_{ix} \| Y_{iy})) X_j \pmod p$

[定理二] 由固定式交談金鑰分配，使用者U_i與U_j可得到相同的交談金鑰。

證明：

$$
\begin{aligned}
SK_{ij} &= (Y_j \cdot Y_{SA}^{\,h(ID_j, Y_{jx} \| Y_{jy})})^{X_i} \pmod p \\
&= (R_j \cdot Z_j^{\,B} \cdot X_{SA}^{\,B h(ID_j, Y_{jx} \| Y_{jy})})^{X_i} \pmod p \\
&= (K_j^{\,B} \cdot Z_j^{\,B} \cdot X_{SA}^{\,B h(ID_j, Y_{jx} \| Y_{jy})})^{X_i} \pmod p \\
&= (K_j \cdot Z_j \cdot X_{SA}^{\,h(ID_j, Y_{jx} \| Y_{jy})})^{B X_i} \pmod p \\
&= (K_j \cdot W_j)^{B X_i} \pmod p \\
&= X_j^{\,X_i B} \pmod p \\
&= SK_{ji} \pmod p
\end{aligned}
$$

由上證明可知，任何人可在得知對方身份相關資訊及公鑰的情況下，正確求出雙方共享的交談金鑰。

[非固定式交談金鑰分配]

1.U_i隨機選取一個數P_i並計算

$T_i = P_i B \pmod p$，將Y_i, ID_i, T_i傳給U_j。

U_j隨機選取一個數P_j並計算

$T_j = P_j B \pmod p$，將Y_j, ID_j, T_j傳給U_i。

2.U_i計算交談金鑰

$$SK_{ij} = (Y_j \cdot Y_{SA}^{\,h(ID_j, Y_{jx} \| Y_{jy})})^{P_i \cdot X_i \cdot T_j} \pmod p$$

U_j計算交談金鑰

$$SK_{ji} = (Y_i \cdot Y_{SA}^{\,h(ID_i, Y_{ix} \| Y_{iy})})^{P_j \cdot X_j \cdot T_i} \pmod p$$

[定理三] 由非固定式交談金鑰分配，使用者U_i與U_j可得到相同的交談金鑰。

證明：

$$
\begin{aligned}
SK_{ij} &= (Y_j \cdot Y_{SA}^{\,h(ID_j, Y_{jx} \| Y_{jy})})^{P_i \cdot X_i \cdot T_j} \pmod p \\
&= (R_j \cdot Z_j^{\,B} \cdot X_{SA}^{\,B h(ID_j, Y_{jx} \| Y_{jy})})^{P_i \cdot X_i \cdot P_j B} \pmod p \\
&= (K_j^{\,B} \cdot Z_j^{\,B} \cdot X_{SA}^{\,B h(ID_j, Y_{jx} \| Y_{jy})})^{P_i \cdot X_i \cdot P_j B} \pmod p \\
&= (K_j \cdot Z_j \cdot X_{SA}^{\,h(ID_j, Y_{jx} \| Y_{jy})})^{B P_i \cdot X_i \cdot P_j B} \pmod p \\
&= (K_j \cdot W_j)^{B P_i \cdot X_i \cdot P_j B} \pmod p \\
&= X_j^{\,B P_i \cdot X_i \cdot P_j B} \pmod p \\
&= (X_j^{\,P_i \cdot X_i \cdot P_j})^{B} \pmod p \\
&= SK_{ji} \pmod p
\end{aligned}
$$

以固定式交談金鑰分配來建立交談金鑰時，只要知道對方的身份相關資訊及公鑰，即可求出共享的交談金鑰。此方式傳輸量較小，而以非固定式交談金鑰分配來建立交談金鑰時，每次都可得到不同的交談金鑰，可防止攻擊者利用取得的舊有交談金鑰來對雙方進行攻擊。

以下將植基於本章所提出的公開金鑰密碼系統，發展出安全且有效率的電子現金型付款系統。

四、智慧型安全電子付款系統

目前電子付款系統的交易，其雙方的身分識別要透過公正的第三者(Certificate Authority；CA)來協助驗證，例如SET機制就是屬於此類的。在本論文所設計的智慧型電子付款系統中，所謂的智慧，就是指在交易過程中，不需所謂的公正第三者來參與身份的識別，只要交易雙方就可以彼此確認身分以及辨識電子貨幣的真偽。第三章所述的自我認證公開金鑰密碼系統正可以被使用來達成這樣的智慧型需求。

根據前述之電子付款系統的安全需求、考慮因素及應具備的功能，我們可描述智慧型電子付款系統整個交易的流程共包含四個主要的階段：初始階段、提款階段、付款階段與清償階段。我們參考 Petersen 和 Poupard [19]所提之機制並採用上述提出的密碼機制來修改，用來達成我們所需的安全需求及效率。

4.1 初始階段

這階段包括使用者要向銀行開設帳戶以及所有參與角色金鑰對的取得。所有參與的角色均採用上一章所提出的植基於 ECC 的自我認證公開金鑰密碼系統向系統中心註冊以取得金鑰對。其中，使用者為達到匿名性，所以使用假名PS_U代替ID_U來跟系統中心溝通，以取得金鑰對。另外，由於匿名性無法提供追蹤性，使得有些非法的交易容易逃過追蹤而不會被發現，所以要加入適當的可追蹤性，來追查非法的交易行為。在這階段中，加入一個受託管理單位，使用者必須向受託管理單位註冊，只有受託管理單位知道使用者真實身份與假名的關連性，以後有問題發生時，就可以請求與受託管理單位合作，追蹤有問題的交易。以下就這階段的各個參與角色的參數作一說明：

<u>銀行(Bank，簡稱 B)</u>

1. ID_B：銀行的身份
2. X_B：銀行的秘密金鑰
3. Y_B：銀行的公開金鑰
4. U-DB：儲存使用者的身分以及銀行帳戶號碼的資料庫
5. W-DB：儲存提款副本(Withdrawal Transcripts)的資料庫
6. D-DB：儲存清償副本(Deposit Transcripts)的資料庫

<u>使用者(User，簡稱 U)：</u>

1. ID_U：使用者的身份
2. PS_U：使用者的假名
3. X_U：使用者的秘密金鑰
4. Y_U：使用者的公開金鑰

5. C-DB：儲存所提領還沒有花費的電子現金的資料庫
6. PS_X-DB：儲存使用者秘密金鑰及秘密資訊的資料庫

商家(Shop，簡稱 S)：

1. ID_S：商家的身份
2. X_S：商家的秘密金鑰
3. Y_S：商家的公開金鑰
4. P-DB：付款資料庫來儲存還沒有清償的電子現金

另外商家還必須要有廢止列表(Revocation Lists)，以便能對金鑰作管理。

5. U-BL：記錄廢止的使用者公開金鑰
6. C-WL：記錄合法的電子現金
7. B-BL：記錄廢止的銀行公開金鑰
8. PS-WL：記錄合法的使用者公開金鑰
9. T-BL：記錄廢止的受託管理單位公開金鑰

受託管理單位(Trustee，簡稱 T)：

1. ID_T：受託管理單位的身份
2. X_T：受託管理單位的秘密金鑰
3. Y_T：受託管理單位的公開金鑰
4. PS_Y-DB：儲存使用者假名與使用者身分之對照關係的資料庫

[開設帳戶]

1. 使用者 U 使用 ID_U 向銀行 B 開設帳戶，得到一組帳戶號碼 acc_U。
2. B 儲存 $?ID_U, acc_U?$ 在 U-DB 資料庫中。

[使用者向受託管理單位註冊]

1. 使用者 U 與受託管理單位 T 共同產生交談金鑰 SK_{UT}，以便在傳輸資料時供加/解密用。
2. U 隨機選取一個數 K_U，並計算以下簽章 R_U 與 S_U：

$$R_U \; ? \; K_U \; ?B \pmod p \; ? \; (R_{UX}, R_{UY}),$$
$$S_U \; ? \; X_U \; ?h(R_{UX} \| R_{UY}, ID_U, Y_{UX} \| Y_{UY})$$
$$? \; K_U \pmod q, \text{ and } ?_U \; ? \; (R_U, S_U).$$

將 $E_{SK_{UT}}(?_U, ID_U, PS_U, Y_U)$ 加密傳給 T。

3. T 收到後驗證以下式子：

$$S_U \; ?B \; ? \; (Y_U \; ? \; Y_{SA} \; ?h(PS_U, Y_{UX} \| Y_{UY})$$
$$?h(R_{UX} \| R_{UY}, ID_U, Y_{UX} \| Y_{UY}) \; ? \; R_U \pmod p$$

若驗證等式成立後，則隨機選取一個數 K_T 並計算

$$R_T \; ? \; K_T \; ?B \pmod p \; ? \; (R_{TX}, R_{TY}),$$
$$S_T \; ? \; X_T \; ?h(R_T, Y_{UX} \| Y_{UY}) \; ? \; K_T \pmod q, \text{and}$$
$$?_T \; ? \; (R_T, S_T).$$

將 $E_{SK_{UT}}(?_T)$ 傳給 U 後，則儲存 $ID_U, PS_U, Y_U, ?_u$ 在 PS_Y-DB 資料庫中。

4. U 收到 $E_{SK_{UT}}(?_T)$ 後驗證以下式子：

$$S_T \; ?B \; ? \; (Y_T \; ? \; Y_{SA} \; ?h(ID_T, Y_{TX} \| Y_{TY})$$
$$?h(R_{TX} \| R_{TY}, Y_{UX} \| Y_{UY}) \; ? \; R_T \pmod p$$

若等式驗證成立後，則儲存 $X_U, Y_U, ?_T$ 在 PS_X-DB 資料庫中。

[定理四] 驗證簽章的同時，亦可驗證對方公開金鑰的有效性。

證明：

驗證簽章如下：

$$S_U \; ?B \; ? \; (X_U \; ?h(R_{UX} \| R_{UY}, ID_U, Y_{UX} \| Y_{UY}) \; ? \; K_U)$$
$$‘ \; B \pmod p$$
$$? \; (X_U \; ?B \; ?h(R_{UX} \| R_{UY}, ID_U, Y_{UX} \| Y_{UY})$$
$$? \; K_U \; ?B \pmod p)$$
$$?(Y_U \; ? \; Y_{SA} \; ?h(PS_U, Y_{UX} \| Y_{UY}) \; ?h(R_{UX} \| R_{UY}, ID_U, Y_{UX} \| Y_{UY})$$
$$? \; R_U \pmod p$$

驗證對方公開金鑰的有效性如下：

根據 $S_U \; ?B \; ? \; (Y_U \; ? \; Y_{SA} \; ?h(PS_U, Y_{UX} \| Y_{UY})$
$?h(R_{UX} \| R_{UY}, ID_U, Y_{UX} \| Y_{UY}) \; ? \; R_U \pmod p$ 式子，我們可以推導出下面式子

$$X_U \; ?B \; ?h(R_{UX} \| R_{UY}, ID_U, Y_{UX} \| Y_{UY})$$
$$? \; (Y_U \; ? \; Y_{SA} \; ?h(PS_U, Y_{UX} \| Y_{UY})$$
$$?h(R_{UX} \| R_{UY}, ID_U, Y_{UX} \| Y_{UY}) \pmod p$$

來驗證對方公開金鑰的有效性。

4.2 提款階段

使用者在銀行開設帳戶後，當使用者要從事電子交易時，可以向銀行提領經過其盲簽章之電子現金，然後銀行會從使用者帳戶中扣除其所提領的款項，使用者提領的電子現金則可以存在智慧卡(Smart Card)中或使用者的電腦。使用者執行以下步驟向銀行提領電子現金：

1. 使用者 U 與銀行 B 共同產生交談金鑰 SK_{UB}，在傳輸資料時加/解密用。
2. U 選擇 $C \; ?_R \; ?0 : 2^{24} ?$ 並計算 $\tilde{C} \; ? \; h(Y_{UX} \| Y_{UY}, C)$ 將 $E_{SK_{UB}}(\tilde{C}, C)$ 傳給 B。
3. B 收到後，隨機選取一個數 K_B 並計算

$\tilde{R}_B$? K_B ?$B(\mathrm{mod}\, p)$將 $\tilde{R}_B$ 傳給 U。

4. U 收到後，隨機選取兩個數 a 與 b 並計算

R_B ? $\tilde{R}_B$?a ? b ?B (mod p) ? (R_{BX}, R_{BY}),

E ? $h(R_{BX} \| R_{BY}, C, Y_{UX} \| Y_{UY})$, and

$\tilde{E}$? E / a (mod q).

將 $\tilde{E}$ 傳給 B。

5. B 收到 $\tilde{E}$ 後，計算$\tilde{S}_B$? X_B ?$\tilde{E}$? K_B (mod q)，

並將 $\tilde{S}_B$ 傳給 U。 然後 B 把 $ID_U, \tilde{C}, X_B$ 儲存在 W-DB 資料庫中，並從使用者帳戶中扣款。

6. U 收到 $\tilde{S}_B$ 後計算 S_B ? $\tilde{S}_B$?a ? $b(\mathrm{mod}\, q)$，? $_B$? (R_B, S_B) 並驗證

S_B ?B ? $(Y_B$? Y_{SA} ?$h(ID_B, Y_{BX} \| Y_{BY})$

?$h(R_{BX} \| R_{BY}, C, Y_{UX} \| Y_{UY})$? $R_B(\mathrm{mod}\, p)$

驗證等式成立後，則儲存電子現金 $(C, ?_B, Y_U)$ 在 C-DB 資料庫中。

4.3 付款階段

在付款階段中，使用者下訂單而經商家確認訂單後，會要求使用者付款，使用者就把從銀行提領的電子現金，傳送給商家作付款的動作。使用者執行以下步驟將電子現金傳送給商家：

1. 商家 S 寄給使用者 U 時間戳記 $mess$。
2. U 收到後，隨機選取一個數 K_C 並計算

R_C ? K_C ?B (mod p) ? (R_{UX}, R_{UY}),

S_C ? X_U ?$h(R_{UX} \| R_{UY}, C, ID_U, mess, Y_{UX} \| Y_{UY})$

? K_C (mod q), and ? $_C$? (R_C, S_C).

將 $C, Y_U, ?_C, ?_B, ?_T$ 傳給商家。

3. 商家收到 $C, Y_U, ?_C, ?_B, ?_T$ 後檢驗：

假如 Y_U ? U-DB，則拒絕電子現金 C；

假如 Y_B ? B-BL 且 $h(Y_{UX} \| Y_{UY}, C)$? C-WL，則拒絕電子現金 C；

假如 Y_T ? T-BL 且 Y_U ? PS-WL，則拒絕電子現金 C。

4. 若上述的三項檢驗都沒問題，則驗證下面三個等式是否成立：

S_T ?B ? $(Y_T$? Y_{SA} ?$h(ID_T, Y_{TX} \| Y_{TY})$

?$h(R_{TX} \| R_{TY}, Y_{UX} \| Y_{UY})$? R_T (mod p),

S_B ?B ? $(Y_B$? Y_{SA} ?$h(ID_B, Y_{BX} \| Y_{BY})$

?$h(R_{BX} \| R_{BY}, C, Y_{UX} \| Y_{UY})$? $R_B(\mathrm{mod}\, p)$, *and*

S_C ?B ? $(Y_U$? Y_{SA} ?$h(PS_U, Y_{UX} \| Y_{UY})$

?$h(R_{UX} \| R_{UY}, C, ID_U, mess, Y_{UX} \| Y_{UY})$? $R_C(\mathrm{mod}\, p)$.

若等式都驗證成立的話，將 $(C, Y_U, mess, ?_T, ?_C, ?_B)$ 儲存在 P-DB 資料庫中，接受使用者的付款。

4.4 清償階段

商家將收到的電子現金傳送給銀行要求清償動作，銀行會檢驗電子現金的正確性，假如一切合法，則將錢存入商家的帳戶中。商家執行以下步驟向銀行要求清償動作：

1. 商家 S 將 $C, Y_U, ?_B, ?_T$ 傳送給銀行 B。
2. B 收到 $C, Y_U, ?_B, ?_T$ 後，驗證以下等式是否成立，並檢驗 C 是不是已經清償過了：

S_T ?B ? $(Y_T$? Y_{SA} ?$h(ID_T, Y_{TX} \| Y_{TY})$

?$h(R_{TX} \| R_{TY}, Y_{UX} \| Y_{UY})$? R_T (mod p),

S_B ?B ? $(Y_B$? Y_{SA} ?$h(ID_B, Y_{BX} \| Y_{BY})$

?$h(R_{BX} \| R_{BY}, C, Y_{UX} \| Y_{UY})$? R_B (mod p).

搜尋 D-DB 資料庫中，假如找到 $(C, ?_C')$，則代表是重複清償，將 $?_C'$ 傳給 S；如沒找到，則 B 傳給 S 一個有關電子現金 C 是合法的回應。

3. S 再將 $ID_S, accs, mess, ?_C$ 傳給 B。
4. B 再驗證以下等式是否成立，並檢驗是否有超支：

S_C ?B ? $(Y_U$? Y_{SA} ?$h(PS_U, Y_{UX} \| Y_{UY})$

?$h(R_{UX} \| R_{UY}, C, ID_U, mess, Y_{UX} \| Y_{UY})$? R_C (mod p)

假如超支的話，則執行以下使用者追蹤協定：

(1) 銀行寄使用者先前的付款副本給受託管理單位：

$(C, Y_U, mess_1, ?_B, ?_T, ?_{C,1})$,.....

$, (C, Y_U, mess_k, ?_B, ?_T, ?_{C,k})$

(2) 受託管理單位驗證所有的付款副本：

Y_U ? U-DB

Y_B ? B-BL 且 $h(Y_{UX} \| Y_{UY}, C)$? C-WL

Y_T ? T-BL 且 Y_U ? PS-WL

S_T ?B ? $(Y_T$? Y_{SA} ?$h(ID_T, Y_{TX} \| Y_{TY})$

?$h(R_{TX} \| R_{TY}, Y_{UX} \| Y_{UY})$? R_T (mod p)

S_B ?B ? $(Y_B$? Y_{SA} ?$h(ID_B, Y_{BX} \| Y_{BY})$

?$h(R_{BX} \| R_{BY}, C, Y_{UX} \| Y_{UY})$? R_B (mod p)

S_C ?B ? $(Y_U$? Y_{SA} ?$h(PS_U, Y_{UX} \| Y_{UY})$

?$h(R_{UX} \| R_{UY}, C, ID_U, mess, Y_{UX} \| Y_{UY})$? R_C (mod p)

(3) 假如都正確，則從 PS_Y-DB 資料庫中

找尋 $(ID_U, Y_U, ?_U)$，將 $(ID_U, ?_U)$ 傳給銀行，以便找出是誰超支花費。假如沒有超支的話，則將錢存入商家的帳戶中，銀行儲存 $(C, Y_U, ID_S, ?_T, ?_C)$ 在 D-DB 資料庫中。

五、 安全性分析

本論文所提出之方法的安全性主要是基於解橢圓曲線離散對數問題之困難度與雜湊函數之安全性。

1. 系統中心無法取得參與者的秘密金鑰 X_i

   系統中心由 W_i, R_i, Y_i, I_i 無法得知參與者的秘密金鑰 $X_i ? W_i ? K_i \pmod q$。因為系統中心無法知道 K_i 而只能由 $R_i ? K_i ? B \pmod p$ 中去求得 K_i，但是這面臨到解橢圓曲線離散對數問題的困難度，所以系統中心根本無法得知參與者的秘密金鑰。

2. 攻擊者無法取得參與者秘密金鑰 X_i 及系統中心秘密金鑰 X_{SA}

   攻擊者可得到的資料為 $W_i, R_i, Y_i, I_i, Y_{SA}$，但由 R_i 不能獲取 Ki 的安全度是基於解橢圓曲線離散對數問題的困難度。$Y_i ? R_i ? Z_i ? B \pmod p$ 則是由使用者與系統中心所共同產生，攻擊者無法得知 Z_i，這也是基於解橢圓曲線離散對數問題的困難度。而由 W_i, Y_i, Y_{SA} 並無法得到 Z_i 與 X_{SA}，這同樣也是面臨到解橢圓曲線離散對數問題的困難度。因此，攻擊者無法經由偽造的 W_i, K_i 以獲得參與者的秘密金鑰 X_i 與系統中心的秘密金鑰 X_{SA}。

3. 參與者無法自行計算有效的公鑰及證明

   由於參與者的公鑰是與系統中心共同產生的，所以使用者無法自行產生公鑰。而在 SA 計算 $W_i ? X_{SA} ? h(ID_i, Y_{ix} \| Y_{iy}) ? Z_i \pmod q$ 以證明公鑰的有效性的這部分，使用者亦無法自行計算，因為不知道系統中心的密鑰 X_{SA} 及 Z_i，因此無法自行計算公鑰的證明。

4. 使用非固定式交談金鑰分配，第三者無法解出交談金鑰

   基於解橢圓曲線離散對數問題的困難度，第三者無法得知下式中的 P_i, X_i，因此第三者無法計算出交談金鑰 SK_{ij}。另外，由於每次要交談時所產生的交談金鑰都不一樣，因此也大大地增加了安全性。

   $$SK_{ij} ? (Y_j ? Y_{SA} ? h(ID_j, Y_{jx} \| Y_{jy})) ? P ? X_i ? T_j \pmod p$$

5. 使用者無法共謀求出系統中心的密鑰 X_{SA}

   若有兩個使用者要共謀求出系統中心的密鑰 X_{SA} 時，須解出以下聯立方程式：

   $$\begin{cases} W_i ? X_{SA} ? h(ID_i, Y_{ix} \| Y_{iy}) ? Z_i \pmod q \\ W_j ? X_{SA} ? h(ID_j, Y_{jx} \| Y_{jy}) ? Z_j \pmod q \end{cases}$$

   其中 Z_i, Z_j 為未知，因兩個方程式有三個未知數，故其 X_{SA} 無法從聯立方程式來解出。

6. 達到 Level 3 的安全等級

   系統中心可偽造出另一組的 X_i', Y_i' 來假扮使用者 U_i，但是這樣發生了一個使用者同時有兩個公開金鑰 Y_i, Y_i'，因此，系統中心的詐騙是會被發現的，因為金鑰的產生是採用 3.2 節所述的自我認證的方式，驗證式 $X_i ? B ? Y_i ? Y_{SA} ? h(ID_i, Y_{ix} \| Y_{iy}) \pmod p$ 並不能成功通過。

7. 達到 Level 4 的安全等級

   使用者使用假名 PS_U 來代替使用者的身分 ID_U 與系統中心溝通，系統中心並不知道使用者的真實身份，因此達到了匿名性，亦即 level 4 的安全等級。

8. 非法交易的可追蹤性

   由於匿名性無法提供追蹤性，使得有些非法的交易容易逃過追蹤而不會被發現，所以要加入適當的可追蹤性，來追查非法的交易行為。在系統中加入一個受託管理單位，使用者必須向受託管理單位註冊，只有受託管理單位知道使用者真實身份與假名的關連性，以後有問題發生時，可以與受託管理單位合作，追蹤有問題的交易。

9. 電子現金之無法偽造性

   在提款階段時，使用者所提領的電子現金是經由銀行所簽署過的，所以沒有人可以偽造銀行來發行電子現金。說明如下：

   $\tilde{S}_B ? X_B ? \tilde{E} ? K_B \pmod q$ 是由銀行所計算的，使用者想要自行發行電子現金，必須得到銀行的密鑰 X_B 以及 K_B，使用者無法由 Y_B 及 $\tilde{R}_B$ 中得到密鑰 X_B 及 K_B，因為會面臨到解橢圓曲線離散對數問題的困難度。因此，電子現金可達到無法偽造性。

10. 超支追蹤(Overspent-tracing)及其使用者之

追蹤(User-tracing)

(1) 為防護電子現金 C 被超支花費，所以銀行寄使用者先前的付款副本給受託管理單位：

$(C, Y_U, mess_1, ?_B, ?_T, ?_{C,1}), \ldots\ldots$

$, (C, Y_U, mess_k, ?_B, ?_T, ?_{C,k})$

(2) 受託管理單位驗證所有的付款副本：

Y_U ? U-DB

Y_B ? B-BL 且 $h(Y_{UX} \| Y_{UY}, C)$? C-WL

Y_T ? T-BL 且 Y_U ? PS-WL

$S_T ?B ? (Y_T ? Y_{SA} ?h(ID_T, Y_{TX} \| Y_{TY})$

$?h(R_{TX} \| R_{TY}, Y_{UX} \| Y_{UY}) ? R_T \pmod p$

$S_B ?B ? (Y_B ? Y_{SA} ?h(ID_B, Y_{BX} \| Y_{BY})$

$?h(R_{BX} \| R_{BY}, C, Y_{UX} \| Y_{UY}) ? R_B \pmod p$

$S_C ?B ? (Y_U ? Y_{SA} ?h(PS_U, Y_{UX} \| Y_{UY})$

$?h(R_{UX} \| R_{UY}, C, ID_U, mess, Y_{UX} \| Y_{UY}) ? R_C \pmod p$

(3) 假如都正確，則從 PS_Y-DB 資料庫中找尋 $(ID_U, Y_U, ?_U)$，將 $(ID_U, ?_U)$ 傳給銀行，以便找出是誰超支花費。

11. 可自行驗證電子現金的真偽

商家在收到使用者花費的電子現金後，可自行驗證電子現金的真偽，而不用請發行電子現金的銀行來幫忙驗證電子現金的真偽，此乃由於採用自我認證的方式。此外，在驗證電子現金真偽（銀行簽章的驗證）的同時，也可以驗證對方的身分及其公開金鑰的有效性，由下面驗證式即可以看出：

$?_B ? (R_B, S_B),$

$S_B ?B ? (Y_B ? Y_{SA} ?h(ID_B, Y_{BX} \| Y_{BY})$

$?h(R_{BX} \| R_{BY}, C, Y_{UX} \| Y_{UY}) ? R_B \pmod p.$

12. 安全的盲簽章

採用盲簽章的技術，銀行便無法去追蹤經過盲簽章的電子現金。銀行就算看到使用者所付款的電子現金，也無法對應到之前經盲簽章所簽署的電子現金，所以銀行無法去追蹤電子現金與其經盲簽章所簽署的電子現金之關係。

六、 複雜度分析

複雜度分析包含計算時間複雜度及資料傳輸量二方面；計算複雜度為各階段所花費的時間，而資料傳輸量為各階段所需的資料通訊傳輸量。

6.1 計算複雜度

智慧型電子付款系統可分為：初始階段、提款階段、交易階段與清償階段。以下分別說明時間複雜度參數之定義以及其各階段之計算複雜度：

分析時間複雜度所用到的參數之定義如下：

T_h：表示執行一次單向雜湊函數運算所需花費的時間

T_{div}：表示執行一次除法模數運算所需花費的時間

T_{mul}：表示執行一次乘法模數運算所需花費的時間

$T_{E?add}$：表示執行一次橢圓曲線加法運算所需花費的時間

$T_{E?mul}$：表示執行一次橢圓曲線乘法運算所需花費的時間

T_{SYM}：表示對稱式密碼系統加/解密所需花費的時間

T_{inv}：表示執行一次乘法反元素運算 $(aa^{?1} ? 1 \bmod p)$所需花費的時間

$T_{\exp}$：表示執行一次指數運算所需花費的時間

在付款系統各階段中，各個參與角色所花費的計算時間複雜度如表 6.1 所示。根據文獻 [1, 11]所作的假設得知，在橢圓曲線中，計算 kP 是針對所有的點 $P ? E(Z_p)$ 且 $p ? 2^{160}$，其中 k 是一個 160 位元的整數。此外，對於 $g^x \bmod p$ 的計算,其中 p 是一個 1024 位元的大質數，x 是一個 160 位元的隨機整數。於是可以推得出 $T_{E?mul} ? 29T_{mul}$，$T_{div} ? T_{mul}$，$T_{E?add} ?$ (5/41) $T_{mul} ?$ 0.12 T_{mul} 與 $T_{\exp} ? 240T_{mul}$ 並得到如表 6.2 所示的計算時間複雜度的概略估計值。此外，表 6.3 列出本論文所設計的植基於 ECC based self-certified public key cryptosystem 之新盲簽章法、植基於 RSA 機制的盲簽章法 [5]、植基於 ElGamal 簽章機制的盲簽章法 [16]及植基於 Schnorr 簽章機制的盲簽章法 [20]之比較。表 6.4 則列出本論文所提出之智慧型安全電子付款系統(以下簡稱方法一)與兩位學者 Petersen 和 Poupard [19](以下簡稱方法二)所提出之電子付款系統作計算時間複雜度比較。從表 6.3 及表 6.4 可得知，本論文所提出之盲簽章法比其它盲簽章法更能使電子付款系統具有高效率，且提出的智慧型安全電子付款系統也比 Petersen 和 Poupard 提出的電子付款系統更有效率。

表 6.1 付款系統中各階段的計算時間複雜度

	初始階段		提款階段		付款階段	清償階段
系統中心	$5T_{E?mul}+4T_{E?add}+4T_{mul}+4T_h$					
使用者	固定式	$9T_{E?mul}+4T_{E?add}+2T_{mul}+5T_h+2T_{SYM}$	固定式	$6T_{E?mul}+4T_{E?add}+T_{mul}+5T_h+T_{SYM}+T_{div}$	$T_{E?mul}+T_{mul}$	
	非固定式	$11T_{E?mul}+5T_{E?add}+2T_{mul}+5T_h+2T_{SYM}$	非固定式	$8T_{E?mul}+5T_{E?add}+T_{mul}+5T_h+T_{SYM}+T_{div}$		
受託管理單位	固定式	$9T_{E?mul}+4T_{E?add}+2T_{mul}+5T_h+2T_{SYM}$				
	非固定式	$11T_{E?mul}+5T_{E?add}+2T_{mul}+5T_h+2T_{SYM}$				
銀行	$2T_{E?mul}+T_{E?add}+T_h$		固定式	$3T_{E?mul}+T_{E?add}+T_{mul}+T_{SYM}$		$6T_{E?mul}+6T_{E?add}+6T_h$
			非固定式	$5T_{E?mul}+2T_{E?add}+T_{mul}+T_{SYM}$		
商家	$2T_{E?mul}+T_{E?add}+T_h$				$6T_{E?mul}+6T_{E?add}+6T_h$	

表 6.2 付款系統中各階段的計算時間複雜度之概略估計

	初始階段		提款階段		付款階段	清償階段
系統中心	$149.48T_{mul}+4T_h$					
使用者	固定式	$263.48T_{mul}+5T_h+2T_{SYM}$	固定式	$176.48T_{mul}+5T_h+T_{SYM}$	$30T_{mul}$	
	非固定式	$321.6T_{mul}+5T_h+2T_{SYM}$	非固定式	$234.6T_{mul}+5T_h+T_{SYM}$		
受託管理單位	固定式	$263.48T_{mul}+5T_h+2T_{SYM}$				
	非固定式	$321.6T_{mul}+5T_h+2T_{SYM}$				

銀行	$58.12T_{mul}+T_h$	固定式	$88.12T_{mul}+T_{SYM}$		$174.72T_{mul}+6T_h$
		非固定式	$146.24T_{mul}+T_{SYM}$		
商家	$58.12T_{mul}+T_h$			$174.72T_{mul}+6T_h$	

表 6.3 各盲簽章法的計算時間複雜度之比較

	盲簽章產生		簽章驗證
本論文提出之盲簽章法	時間複雜度	$T_h+T_{E?add}+3T_{E?mul}+2T_{mul}+T_{div}+2T_{SYM}$	$2T_h+2T_{E?add}+2T_{E?mul}$
	時間複雜度概略估計	$90T_{mul}+T_h+2T_{SYM}$	$58.24T_{mul}+2T_h$
植基於 RSA 之盲簽章法 [5]	時間複雜度	$2T_{\exp}+T_{mul}+T_{div}$	$T_{\exp}$
	時間複雜度概略估計	$482T_{mul}$	$240T_{mul}$
植基於 ElGamal之盲簽章法 [16]	時間複雜度	$T_{\exp}+7T_{mul}+4T_{inv}$	$3T_{\exp}+T_{mul}$
	時間複雜度概略估計	$247T_{mul}+4T_{inv}$	$721T_{mul}$
植基於 Schnorr 之盲簽章法 [20]	時間複雜度	$5T_{\exp}+6T_{mul}+T_h$	$3T_{\exp}+2T_{mul}$
	時間複雜度概略估計	$1206T_{mul}+T_h$	$722T_{mul}$

表 6.4 電子付款系統的計算時間複雜度之比較

		初始階段		提款階段		付款階段	清償階段
方法一	系統中心	$149.48T_{mul}+4T_h$					
方法二		$240T_{mul}$					
方法一	使用者	非固定式	$321.6T_{mul}+5T_h+2T_{SYM}$	非固定式	$234.6T_{mul}+5T_h+T_{SYM}$	$30T_{mul}$	
方法二			$1683T_{mul}+3T_h+2T_{SYM}$		$1683T_{mul}+3T_h+4T_{SYM}$	$241T_{mul}+1T_h$	
方法一	受託管理單位	非固定式	$321.6T_{mul}+5T_h+2T_{SYM}$				
方法二			$1443T_{mul}+3T_h+2T_{SYM}$				

方法一	銀行	$58.12T_{mul}+T_h$	非固定式	$146.24T_{mul}+T_{SYM}$		$174.72T_{mul}+6T_h$
方法二		$240T_{mul}$		$962T_{mul}+T_h+4T_{SYM}$		$723T_{mul}+3T_h$
方法一	商家	$58.12T_{mul}+T_h$			$174.72T_{mul}+6T_h$	
方法二		$240T_{mul}$			$723T_{mul}+3T_h$	

6.2 **資料傳輸量**

資料傳輸量的分析，以資料傳輸的總量來計算，以下分別說明資料傳輸量參數之定義以及其各階段之資料傳輸總量：

分析資料傳輸量所用到的參數之定義如下：

$|p|, |q|$**：經過乘法模數運算後的大小**

$|I|$**：使用者身份** I **或是假名的長度**

$|E|$**：使用交談金鑰加密過的長度**

$|m|$**：時間戳記的長度**

$|X|$**：使用者隨機選取的參數長度。例如：使用者所選取的** C

$|accs|$**：銀行帳號長度**

在付款系統各階段中，各階段的資料傳輸總量如表 6.5 **所示。**

表 6.5 **付款系統中各階段資料傳輸總量**

初始階段		提款階段		付款階段	清償階段
固定式	$6\|I\|+10\|p\|+2\|E\|+4\|q\|$	固定式	$2\|I\|+3\|p\|+2\|q\|+\|E\|$	$4\|p\|+3\|q\|+\|X\|+\|m\|$	$4\|p\|+3\|q\|+\|X\|+\|m\|+\|I\|+\|accs\|$
非固定式	$6\|I\|+12\|p\|+2\|E\|+4\|q\|$	非固定式	$2\|I\|+5\|p\|+2\|q\|+\|E\|$		

七、結論與建議

本論文發展出一套以橢圓曲線密碼系統為基礎的具自我認證公開金鑰密碼系統，且以此自我認證公開金鑰密碼系統建構出交談金鑰、數位簽章及盲簽章等安全機制，並將這些技術實際應用在較具傳統方式的電子現金型付款系統，藉以提升安全電子付款機制的效率，使即時性的安全電子付款成為可行的方案。

橢圓曲線密碼系統的運算較現存的其它公開金鑰密碼系統更快速，且以較少之位元數達到相同的安全度，相對地金鑰的儲存空間也減少。自我認證公開金鑰密碼系統，除了可以防止系統中心偽造或假扮使用者外，據其所發展出來的安全機制，可以不用系統中心的幫忙驗證，就可以自行驗證交易雙方的身分與公鑰的有效性。在本論文中，驗證電子錢的真偽，不用透過銀行來協助，交易雙方就可以自行驗證，且同時也可驗證銀行的公開金鑰的正確性，減少了額外花費的驗證時間。另外，由於系統提供使用者匿名性，為了防止非法交易能逃過追蹤，在系統中加入了受託管理單位，來幫助對有問題的非法交易作適當的追蹤，以達到一個公平的智慧型電子付款機制。

電子現金是傳統紙張貨幣數位化的理想，它須擁有傳統紙張貨幣所應具有之特性，而本論文中所設計的電子現金，還未具有現金

可分割性之特性，因此未來的研究方向將使其達到此目標，使得智慧型安全電子付款機制可以更臻完備。

參考文獻

[1] **林祝興、李正隆，** "Elliptic-curve undeniable signature schemes," **十一屆全國資訊安全會議，**2001**，**pp. 331-338。

[2] Bellart, M., Garay, J.A., Hauser, R., Herzberg, A., Krawczyk, H., Steiner, M., Tsudik, G., and Waidner, M., "iKP -- a family of secure electronic payment protocols," *Proceedings of the First USENIX Workshop on Electronic Commerce*, New York, July 1995.

[3] Carmenisch, J.L., Piveteau, J.M., and Sradler, M.A., "Blind signatures based on the discrete logarithm problem," *Rump Session of Eurocrypt'94*, Pergia, Italy, 1994.

[4] Chaum, D., "Blind signature for untraceable payments," *Advances in Cryptology:Crypto'82*, 1983, pp.199-203.

[5] Chaum, D., Fiat, A., and Naor, M., "Untraceable electronic cash," *Advances in Cryptology: Crypto'88*, 1988, pp. 319-327.

[6] Cox, B., Tygar, J.D., and Sirbu, M., "NetBill security and transaction protocol," *Proceedings of the First USENIX Workshop on Electronic Commerce*, New York, July 1995.

[7] Cybercash web site, URL: http://www.cybercash.com.

[8] ElGamal, T., "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, Vol. IT-31, No. 4, 1985, pp. 469-472.

[9] Girault, M., "Self-certified public keys," *LNCS 547, Advances in Cryptology: Proc Eurocrypt'91*, Springer, 1992, pp. 490-497.

[10] Harn, L., "Cryptanalysis of the blind signatures based on the discrete logarithm problem," *Electronics Letters*, Vol. 31, No.14, 1995.

[11] Jurisic, A., and Menezes, A.J., "Elliptic curves and cryptography," *Dr. Dobb's Journal*, 1997, pp. 26-35.

[12] Koblitz, N., "Elliptic curve cryptosystems," *Mathematics of Computation*, Vol. 48, No. 17, 1987, pp. 203-209.

[13] MasterCard and VISA, Secure Electronic Transaction (SET) Specification, 1996.

[14] Medvinsky, G., and Neuman, B.C., "NetCash: a design for practical electronic currency on the Internet," *Proceedings of 1st the ACM Conference on Computer and Communication Security*, Nov. 1993.

[15] Miller, V.S., "Use of elliptic curves in cryptography," *Advances in Cryptology: Crypto'85*, 1985, pp. 417-426.

[16] Mohammed, E., Emarah, A.E., and El-shennawy, K.H., "A blind signature scheme based on ElGamal signature," *Seventeenth national radio science conference*, Feb. 2000, pp. 22-24.

[17] Neuman, C., and Medvinsky, G., "Requirements for network payment: the NetCheque perspective," *Proceedings of IEEE COMPCON'95*, March 1995.

[18] Peirce, M., and O'Mahony, D., "Scaleable, secure cash payment for www resources with the payme protocol set," *4th International World Wide Web Conference*, Dec. 11-14, 1995.

[19] Petersen, H., and Poupard, G, "Efficient scalable fair cash with off-line extortion prevention," *Technical Report LIENS –97-7*, Ecole Normale Superieure,May 1997.

[20] Pointcheval, D., and Stern, J., "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, Vol. 13, 2000, pp. 361-396.

[21] Rivest, R., Shamir, A., and Adleman, L., "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, Vol. 21, No. 2, Feb. 1978, pp. 120-126.

[22] Schnorr, C.P., "Efficient identification and signatures for smart cards," *Advances in Cryptology: Crypto'89*, 1990, pp. 339-351.

[23] Shamir, A., "Identity-based cryptosystems and signature schemes," *Advances in Cryptology: Crypto'84*, 1985, pp. 47-53.

[24] Schoenmakers, B., "Basic security of the ecash™ payment system," *State of the Art in Applied Cryptography, Course on Computer Security and Industrial Cryptography*, Leuven, Belgium, June 3-6,1997, Vol. 1528 of Lecture Notes in Computer Science, pp. 338-352.