離線且可計息之電子現金

陳紀盈
世新大學
chi1119@so-net.net.tw

莊文勝
世新大學
wsjuang@cc.shu.edu.tw

摘要

根據多項調查數據顯示，網際網路的使用率正逐年上升中，隨著網際網路的普及也帶動了電子商務的發展。然而電子商務是否能成功的重要關鍵之一就是該市場是否有公平且安全的電子現金系統。若是顧客對網路交易的信心不足，就不會從事網路交易，也就阻礙了電子商務的發展，因此一個好的電子現金系統可以協助電子商務成功的推動。

什麼樣的電子現金系統能讓使用者覺得安全、便利且願意使用呢？在本論文中，我們先對過去學者所提出來的電子現金方案加以研究。我們發現之前多位學者所發表的電子現金相關論文已對加強安全性的部分提出不少可行的方案，然而電子現金的使用率卻依然不盡理想。是什麼原因造成使用上的不普及呢？我們歸納出二個主要原因，第一是當電子現金被提出後未使用之前，電子現金在帳務上還是屬於銀行，但是顧客卻無法取得該段期間的利息而蒙受損失，這成為顧客不願使用電子現金的原因之一。第二是商家與銀行所要付出的連線交易成本，在之前學者所提的論文中，大多屬於連線交易，商家與銀行為了保持交易的順暢及不中斷，必須使用相當的頻寬，購買足夠頻寬所需付出的成本也許是商家與銀行所不願意負擔的，這也成為阻礙電子現金發展的原因之一。

於是我們針對這二點提出了一個離線且可計息的電子現金方案，希望能透過利息的給予來吸引顧客及商家使用電子現金；並透過離線交易來降低商家及銀行加入本機制的成本以及因斷線而無法交易所造成的損失，藉以提升電子現金在電子商務上的應用。

關鍵詞：電子現金、盲簽章、可計息電子現金、數位簽章、離線交易。

1.前言

因為網際網路環境的不安全再加上作業系統設計上的漏洞，讓有心人士能利用這些弱點來從事不法行為，造成偽造、詐騙、盜用等等事件層出不窮，電子商務的糾紛也因而產生。為防止這些不法行為的發生，多位學者所發表的電子現金相關論文[6][7][4][11] 已對此部份做了很好的研究，然而電子現金的使用率卻依然不理想。相信除了安全問題的顧慮之外，還有其他的問題存在。

以實務方面來看，電子現金大多屬於預付，這點可能是電子現金至今未能普及的原因之一。因為在顧客將電子現金提出之後到消費該電子現金的期間，雖然錢在帳務上還是屬於銀行，但顧客卻無法取得該期間的利息所得。為了吸引及鼓勵顧客使用電子現金，應該讓顧客在該段期間仍享有應得的利息。因此我們希望設計出一個可計息的電子現金方案，提供顧客將電子現金從銀行提領後，尚未至商家消費前仍可享有計息的服務，用以改善一般電子現金預付的缺點，來增加一般大眾對預付機制的接受度。

此外，由於連線交易的安全性較高，大多數的電子現金都採用連線交易。在一般正常情況之下，連線交易當然是最佳的選擇，但是有可能遇到網路擁塞，網路斷線，伺服器當機的情形，若是因此而無法交易所蒙受的損失將無法估計，再加上連線交易需要固定的頻寬，在硬體設備的建立上也要花費不少成本。所以為了減少無法連線交易所造成的損失，以及降低小本經營的商家加入本研究的門檻，我們的機制採用離線交易。如何使用離線交易減少

交易時的連線成本又能達到防止重複消費及其他安全性問題，這些也是我們所要考慮並探討的。

我們將會先對之前學者所提出的電子現金方案加以蒐集及探討，來找出他們的優點及不足的地方，再來設計出我們的電子現金方案。我們的目標是設計出一個安全、便利、又能吸引人使用的電子現金系統，進而提高電子現金的應用，以促進電子商務的發展。

本論文有以下二點假設：

本論文所使用的 Tamper Proof Device [1][2][3] 為一個可以信賴、防破壞及防資料被竄改的硬體裝置。它必須有能力防止未授權的存取，而由他產生的資料是可被信賴的。再設計出我們的電子現金協定之前，我們必須先假設這樣的硬體裝置是存在的。然而如何設計出 Tamper Proof Device 以及 Tamper Proof Device 交易時的詳細運作過程，並不在本研究的範圍之內。

利率的高低影響著利息的多寡，利息的多寡也影響本研究計算利息的意義。若是銀行提供的利率過低、單筆消費金額太少、提款日期到消費日期間的期間太短以至於計算出之利息小於最小面額或是不足以吸引顧客使用，這些並不在本研究的討論範圍，本研究的重點在於提供了計息這項服務。

2.相關文獻的探討

Chaum 學者在 1982 年提出了第一個盲簽章 [6]，讓顧客在消費時能保有其匿名性而不會暴露個人資訊。Fan 等學者在 2000 年提出可附加日期的電子現金[7]，在電子現金上附加消費日期，可用來做利息的計算。 Chang 等學者於 2003 年發表可彈性附加日期的電子現金 [4]，其所附加的日期可隨不同曆法變化而彈性做調整。Juang 學者 2007 年發表了可彈性附加日期的預付式電子現金 [11]，他的方案在提款時就由銀行將提款日期內嵌在電子現金之中，而由顧客在消費時附加消費日期，銀行就可以支付提款日期到回存日期之間的利息。以上四位學者的方案，我們將在本章做詳細的介紹。

2.1 Chaum 學者之方案

Chaum 學者在 1982 年所提出的不可追蹤的盲簽章 [6]，目的是確保顧客在交易時的匿名性。在 Chaum 學者的方案中，包含了三個主要參與角色：銀行、顧客以及商店。在真正消費前，顧客會先至銀行提領電子現金，當要消費時顧客將電子現金支付給商家進行交易，待商家連線至銀行確認沒有重複消費及驗證此筆電子現金的合法性之後，商家才會接受並完成該筆交易。此方案分為四個階段：

（一）初始化階段 (Initializing)

銀行先隨機選擇兩個不同的大質數 p 和 q，然後計算兩個數 $n = pq$，$\phi = (p-1)(q-1)$，銀行再選擇一個大整數 e $(1 < e < \phi)$ 且 $GCD(e,\varphi) = 1$，並計算整數 d $(1 < d < \phi(n))$，d 必須滿足 $ed \equiv 1 \bmod \phi$。最後銀行會公佈公鑰 (e,n) 及保留私鑰 (d,p,q)。

（二）提款階段 (Withdrawing)

假設銀行所發行的電子現金面額都是 w 元。當顧客要提出 w 元時，先隨機選取一個亂數 $r \in z_n^*$，而 m 是要被簽署的訊息，計算 $\alpha = r^e H(m) \bmod n$，將 α 傳送給銀行做簽章。當銀行收到 α 後，銀行使用私鑰 d 對 α 做簽章，其方法為計算 $t = \alpha^d \bmod n$，並將 t 回傳給顧客。最後銀行從顧客的帳戶中扣除 w 元的提款金額。

（三）消除盲因子 (Unblinding)

當顧客收到 t 後，顧客藉由計算 $s = r^{-1}t \bmod n$ 來消除盲因子，s 就是 m 經由銀行簽署的簽章，(m,s) 表示為 w 元的電子現金。

（四）存款階段 (Depositing)

顧客使用電子現金 (m,s) 向商家消費，當商家收到電子現金之後，透過計算 $s^e \equiv H(m) \bmod n$ 來驗證電子現金的正確性。商家驗證成功之後，會連線至銀行要求銀行檢查該筆電子現金是否有重複消費。若銀行驗證該電子現金無誤且無重複消費情形後，則將 w 元的

金額存入商家的帳戶之中，並將該筆電子現金記錄於銀行的資料庫中，用來防止未來電子現金被重複使用的情形發生,最後商家會完成該筆交易。

在提款階段，由於顧客先隨機選取亂數 r 再與 m 計算之後，才傳送 α 給銀行做簽章，所以銀行不知道自己簽署訊息的內容。而銀行簽署好的簽章為 t ， t 消除盲因子之後為 s ，顧客在消費時使用的是電子現金 (m,s)，銀行並不能透過 t 來追蹤到 s，銀行也就無法得知是誰使用了這筆電子現金，基於上述二點顧客在交易時可以達到匿名性。

2.2 Fan 等學者之方案

Fan 等學者在 2000 年提出可附加日期的電子現金 [7]，他讓顧客在消費時在電子現金上附加消費日期，可用來做計息之用，而附加消費日期並不會影響電子現金的不可連結性。此方案是以 Chaum 學者的方案為基礎。首先，以 1+ 目前西元年最後二個有意義的位元來表示年;假設今年為西元 2007 年，則年以 08 來表示。月份則是分別以 1 到 12 來表示。日則可能為 1 到 28、29、30、31，視當月天數而定。除此之外，令 H 為單向雜湊函數，並定義 $H^0(m)=m$ ，$H^i(m)=H(H^{i-1}(m))$，$i\geq 1$。這個方案包含四個階段：

（一）初始化階段 (Initializing)

銀行隨機選擇兩個不同的大質數 p 和 q，計算二個數 $n=pq$，$\phi=(p-1)(q-1)$。然後銀行隨機選擇一個大整數 e 其中 $1<e<\varphi$ 且 $GCD(e,\varphi)=1$，並計算整數 d $(1<d<\phi)$ 且 $ed\equiv 1 \bmod \varphi$，最後銀行會公佈公鑰 (e,n) 及保留私鑰 (d,p,q)，假設每個由銀行發行的電子現金面額為 w 元。

（二）提款階段 (Withdrawing)

顧客隨機選取一個整數 $r\in z_n^*$ 為盲因子及六個亂數 $(x_1,x_2,x_3,x_4,x_5,x_6)$，盲因子及六個亂數為祕密訊息並不公開。之後顧客計算並傳送 $\beta=r^eH(m) \bmod n$ 給銀行，此處的 $m=H^{100}(x_1)\|H^{100}(x_2)\|H^{12}(x_3)\|H^{12}(x_4)\|H^{31}(x_5)\|H^{31}(x_6)$（符號 ‖ 表示連結符號)。當銀行收到 β 值後，銀行計算 $t=\beta^d \bmod n$ 並將簽章結果 t 值回傳給顧客,最後銀行會從顧客的帳戶中扣除 w 元的提款金額。

（三）消除盲因子 (Unblinding)

當顧客收到 t 值後，顧客進行消除盲因子的動作，計算 $s=r^{-1}t \bmod n$， s 就是銀行對 m 做簽章的結果。(m,s) 則為面值 w 元的電子現金。

（四）存款階段 (Depositing)

當顧客要消費時,需將消費當天的日期附加於電子現金上。顧客藉由計算 $a_1=H^a(x_1)$ ，$a_2=H^{100-a}(x_2)$ ， $a_3=H^b(x_3)$ ， $a_4=H^{12-b}(x_4)$ ，$a_5=H^c(x_5)$ ，$a_6=H^{31-c}(x_6)$ 將電子現金附加上日期。（符號 a 代表年；a = 1 + 目前年份最後二個有意義的位元；b 代表月；c 代表日）。之後顧客將附加消費日期的電子現金 $(a,b,c,s,a_1,a_2,a_3,a_4,a_5,a_6)$ 傳送給商家，商家在收到電子現金後藉由計算 $s^e\equiv H(H^{100-a}(a_1))\|H^a(a_2)\|H^{12-b}(a_3)\|H^b(a_4)\|H^{31-c}(a_5)\|H^c(a_6)) \bmod n$ 來驗證電子現金的正確性。驗證無誤之後，商家再將此筆電子現金傳送給銀行，銀行先檢查資料庫確認顧客是否有重複消費此筆電子現金的情形，之後同樣藉由計算 $s^e\equiv H(H^{100-a}(a_1))\|H^a(a_2)\|H^{12-b}(a_3)\|H^b(a_4)\|H^{31-c}(a_5)\|H^c(a_6)) \bmod n$ 驗證電子現金的正確性。最後，銀行將附加消費日期的電子現金儲存至資料庫中，並將面值 w 元的金額存入至商家的帳戶中。

Fan 等學者的方案比 Chaum 學者 [6] 的電子現金增加了附加消費日期的功能,他附加消費日期的方式是使用了大量的雜湊函數計算。舉例來說，當符號 a、 b、 c，分別代表年、月、日，則表示要做 4(100 + 12 + 31) = 572 次雜湊函數的計算才能取得並驗證一個電子現金。其長度為 6λ，λ 為單向雜湊函數的計算的結果。若是將日期簡化為 (a,u)，a = 1 + 目前西元年最後二個有意義的位元，u 為當年的第幾個日子，$1\leq u\leq 366$。在此假設下取得並驗證一個電子現金

必須做 4(100 + 366) = 1864 次雜湊函數的計算，此時的長度為 4λ。較少次數的雜湊函數計算所產生的電子現金會較大；較多次數的雜湊函數計算所產生的電子現金會較小，要如何決定就取決於使用者是要節省計算時間或是儲存空間。

使用雜湊函數運算雖然在效能上比簽章好，但雜湊函數目前已相繼被破解，安全上恐有疑慮。另外，他所附加的消費日期格式是固定的且只能使用在西曆上，其他曆法無法使用。而且消費日期由顧客產生，商家可以與顧客密謀將消費日期提前來向銀行詐騙更多的利息。

2.3 Chang 等學者之方案

Chang 等學者於 2003 年發表之可彈性附加日期的電子現金 [4]，他的方案也是以 Chaum [6] 的方案為基礎。他附加消費日期的方式是首先在提款階段，顧客除了取得一個是電子現金之外，還要取得另一個日期憑據 (Date slip)，日期憑據是用來在付款階段請求銀行附加消費日期的一個憑據。之後在付款階段，顧客傳送電子現金及日期憑據給銀行，銀行驗證日期憑據的合法性之後將電子現金附加上消費日期。這個方案總共有五個階段：

（一）初始階段 (Initializing)

銀行產生二對 RSA 公鑰，公鑰 (e,n) 和 (e',n') 及私鑰 (d,p,q) 和 (d',p',q') 並且公開二個單向雜湊函數 G 及 H。

（二）提款階段 (Withdrawing)

顧客隨機選擇一個盲因子 $r_1 \in Z_n^*$，並計算 $a = r_1^e H(m) \bmod n$，將 a 傳給銀行，請求銀行對 m 做簽章。銀行接到 a 後以其私鑰進行簽章，計算 $t_1 = a^d \bmod n$ 後傳送 t_1 給顧客並且從顧客帳戶扣除 w 元的金額。

（三）除盲因子階段 (Unblinding)

顧客接收到 $t_1 = a^d \bmod n$ 後，進行除盲因子的動作，計算 $s = r_1^{-1} t_1 \bmod n$， s 是銀行對 m 所作的簽章，(m,s) 表示為面值 w 元的電子現金。顧客再隨機選出另一個盲因子 $r_2 \in Z_{n^*}^*$，計算 $\beta = r_2^{e^*} G(s) \bmod n^*$ 後將 β 傳送給銀行請求銀行簽章做為日期憑據。銀行以他的私鑰做簽章，計算 $t_2 = \beta^{d^*} \bmod n^*$ 並傳回 t_2 給顧客，顧客做消除盲因子計算 $\delta = r_2^{-1} t_2 \bmod n^*$，$\delta$ 就是日期憑據。

（四）附加日期階段 (Date-attaching)

當顧客決定消費時，傳送 δ, s 和消費日期 (a,b,c) 給銀行，銀行藉由計算 $\delta^{e^*} = G(s) \bmod n^*$ 來做驗證，若驗證成功，銀行以私鑰 d^* 對 $G(s \| a \| b \| c)$ 作簽章，計算 $s' = G^{d^*}(s \| a \| b \| c) \bmod n^*$ 後傳送 s' 給顧客。在這階段顧客並沒有透露他的身份，銀行是透過對 δ 的信任來對 $G(s \| a \| b \| c)$ 作簽章，而且是使用匿名通道來傳送。

（五）存款階段 (Depositing)

顧客傳送 $(s', m, s, (a,b,c))$ 給商家消費時，商家藉由計算 $s^e = H(m) \bmod n$ 及 $s'^{e^*} = G(s \| a \| b \| c) \bmod n^*$ 來驗證，若驗證成功則此電子現金為合法的。之後商家請求銀行確認是否有重複消費，若沒有，商家就接受該筆交易。在傳送該電子現金給銀行時，銀行並不會立即將電子現金存入商家的帳戶中，而是等到所附加的消費日期到了之後，銀行才將此筆電子現金存入商家的帳戶，但是銀行會以交易日期在資料庫中記錄該筆電子現金已經使用過。

在 Chang 等學者的方案中附加消費日期的方式，相較於 Fan 等學者 [7] 的方式更具有彈性，他所附加的消費日期格式可隨著不同的曆法而做調整，並提供顧客合法變更電子現金上所附加之消費日期的功能。由於消費日期經過銀行的簽章，無法將消費日期提前，所以可防止顧客與商家密謀詐騙利息。但也由於使用了二個簽章，在運算及通訊上增加了不少成本。

2.4 Juang 學者之方案

Juang 學者 2007 年發表了可彈性附加日期的預付式電子現金 [11]，他的方案在提款時就由銀行將提款日期內嵌在電子現金之中，而由顧客在

消費時附加消費日期，若要更改消費日期只要重簽電子現金即可。Juang 學者的方案包含了四個階段，（一）初始化階段、（二）提款階段、（三）附加日期階段及（四）存款階段。在初始化階段銀行產生他的私鑰並公佈相對應的公鑰。在提款階段，顧客以部分盲簽章的方式向銀行請求一個附加提款日期電子現金。這個過程中顧客還隨機產了數位假名金鑰對，其中數位假名私鑰可讓顧客在附加日期階段對消費日期做簽章，而且消費日期必須大於提款日期。在消費時，顧客就可以將這個附加了消費日期的電子現金傳送給商家，商家在驗證成功之後，可以請求銀行檢查是否有重複消費，銀行回應驗證成功之後，商家付提款日期到消費日期之間的利息給顧客。在存款階段，商家可以將電子現金存入他在銀行的帳戶之中，銀行可付提款日期到存款日期之間所產生的利息給商家。其詳細的作法如下：

假設 m 是要被簽章的訊息，h_1 和 h_2 為二個單向雜湊函數，p 和 q 為二個大質數，而 $(p-1)$ 被 q 整除，ρ 是 Z_p^* 的一個生成值。令 $g \equiv_p \rho^{(p-1)/q}$，$\varpi(x,y,c) \equiv_q h_1(c)x + h_2(c)y$ 為公開多項式，c_1 為提款日期，c_2 為消費日期。

（一）初始階段 (The initializing phase)

銀行隨機產生金鑰對，公鑰為 $y_1 \equiv_p g^{z_1}, y_2 \equiv_p g^{z_2}$，私鑰為 $z_1, z_2 \in Z_q$。

（二）提款階段 (The withdrawing phase)

假設顧客的提款日期為 c_1，在顧客提款時銀行產生相對應的金鑰，其公鑰為 $y_{c_1} \equiv_p g^{\varpi(z_1,z_2,c_1)} \equiv_p g^{h_1(c_1)z_1 + h_2(c_1)z_2} \equiv_p y_1^{(h_1(c_1))} y_2^{(h_2(c_1))}$，此公鑰包含了提款日期 c_1，私鑰為 $\varpi(z_1,z_2,c_1) \equiv_q h_1(c_1)z_1 + h_2(c_1)z_2$。

1. 銀行隨機選擇 $k \in Z_q$，計算 $\hat{r} \equiv_p g^k$，將 $\hat{r}$ 傳送給顧客。
2. 顧客在接收到 $\hat{r}$ 之後作下列步驟：
   A. 隨機產生數位假名私鑰 $z_3 \in Z_q$，其相對應的公鑰為 $y_3 \equiv_p g^{z_3}$，並計算 $M = (m \| y_3)$，m 是包含事先約定訊息的盲訊息。
   B. 隨機產生 $\alpha \in Z_q$ 及 $\beta \in Z_q^*$，並計算 $r \equiv_p Mg^{\alpha}\hat{r}^{\beta}$ 及 $\hat{m} \equiv_q \beta^{-1} r$。
   C. 若 $\hat{m} \neq 0$ 則將 $\hat{m}$ 傳送給銀行，若不是則重回步驟 A。
3. 銀行在接收到 $\hat{m}$ 之後，計算 $\hat{s} \equiv_q \hat{m}\varpi(z_1,z_2,c_1) + k$ 後將 $\hat{s}$ 傳送給顧客。並從顧客帳戶中扣除 w 元。
4. 顧客計算 $S \equiv_q \hat{s}\beta + \alpha$，$(r,s,c_1)$ 就是 m 的部分盲簽章，表示面額 w 元之電子現金。

（三）附加日期階段 (The date-attaching phase)

當顧客要使用該電子現金時，先隨機產生 $k' \in Z_q$，再計算 $r' = (g^{k'} \bmod p) \bmod q$，$s' \equiv_q k'^{-1}(h_1(r \| s \| c_1 \| c_2)) + z_3 r'$，在電子現金 (r,s,c_1) 上簽署消費日期 c_2，然後顧客傳送 (r,s,r',s',c_1,c_2) 給商家。商家計算 $M \equiv_p g^{-s} y_{c1}^{r} r = (m \| y_3)$ 來驗證這個電子現金，驗證結果若 m 包含事先約定的檢查訊息，則計算 $u_1 = h_1(r \| s \| c_1 \| c_2)s'^{-1} \bmod q$ 及 $u_2 = r's'^{-1} \bmod q$，$(g^{u1} y_3^{u2} \bmod p) \bmod q = r'$ 來確認顧客簽章的正確性，且消費日期必須大於提款日期 $c_2 \geq c_1$，若此電子現金是合法的，商家可請求銀行檢查是否有重複消費的情形。若是沒有重複消費情形，商家就會接受該筆交易，並支付提款日期到消費日期之間的利息給顧客。

（四）存款階段 (The depositing phase)

在消費日期之後，商家可以將此電子現金存入銀行帳戶中。銀行會加入 w 元至商家的帳戶，並付提款日期至存款日期之間的利息給商家，且記錄此筆交易於資料庫中。

相較於 Fan [7] 以及 Chang [4] 學者的方案只能給予消費日期到回存日期之間的利息，在 Juang 學者的方案中，由於在提款階段附加了提款日期，他的方案可以提供提款日期到回存日期之間的利息。這讓顧客可以享有提款日期到消費日期之間的利息，而商家則可享有消費日期到回存日期之間的利息，而且也不會有向銀行詐騙利息的問題發生。由於 Juang 學者的方案的計息方式較為完整，提款日期及消費日期的格式也具有彈性，而附

加消費日期只使用到一個簽章，因此我們將以 Juang 學者方案為基礎加以改良，來設計出我們的電子現金方案。

3.本研究的機制

我們的機制為離線且可計息之電子現金，是以 Juang 學者的方案 [11] 為基礎，再加上參考 Juang 學者的匿名且離線的多方授權付款方案 [10] 來提出本研究的協定。為了減少因斷線而造成無法交易的損失，我們的機制採用離線交易。並透過利用 Tamper-Proof Device 的安全性以及 Payword 的特性來防止重複消費的發生，達到與連線交易相同的安全性。在我們電子現金機制中，顧客在實際發生消費行為前，必須先至銀行提領電子現金，在提領的同時，銀行產生內嵌提款日期的金鑰來對電子現金做簽章，並從顧客的銀行帳戶中扣除此筆金額。而顧客在消費時使用 T 的私鑰在電子現金上附加消費日期，由商家給予顧客提款日期至消費日期之間的利息。最後商家將該筆電子現金存入銀行，並向銀行領取提款日期至存款日期之間的利息。我們的方案可分為（一）準備階段、（二）初始化階段、（三）提款階段、（四）付款階段及（五）存款階段共五個階段，還有三個主要的角色 Tamper-Proof Device、銀行及商家，圖 1 為本機制的流程。

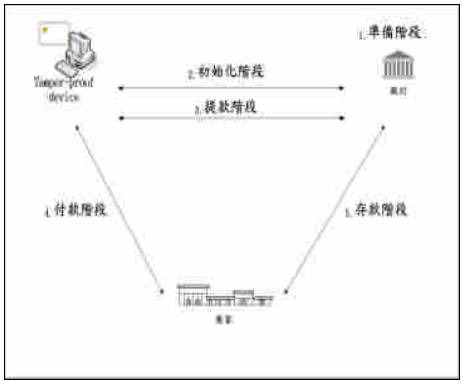

圖 1　本機制流程圖

3.1 符號說明

表 1 所列是本機制所會使用到的參數及符號以及他們的說明。

表 1 本研究參數及符號說明

符號	說明
T	Tamper-Proof Device
$Card_{no}$	T 的出廠序號
RD	事先所約定好的訊息
d_T	初始階段完成前存放於 T 的私鑰
e_T	初始階段完成前存放在銀行與 T 的私鑰相對應的公鑰
d_b	銀行用來對 T 數位假名公鑰簽章的金鑰
$Cert_{d_b}(H(e_z))$	T 被銀行用 d_b 簽章後的公鑰
ID_c	顧客 ID
ID_s	商家 ID
$P_{ID} = \xi(ID_C \| \lambda)$	Unique Header，ζ 是一個重排函數(Permutation Function)；λ 是一個亂數，用來避免暴力攻擊。
C_1	提款日期
C_2	本次消費日期
C_3	前次消費日期
p,q	二個大質數，p 整除於 $q-1$
ρ	$\rho \in Z_p^*$
H, h_1, h_2	單向雜湊函數
$g \equiv_p \rho^{(p-1)/q}$	公開參數
$\varpi(x,y,c) \equiv {}_q h_1(c)x + h_2(c)y$	公開多項式
PK_B	銀行與商家之間傳遞訊息之公鑰
SK_B	銀行與商家之間傳遞訊息之私鑰

3.2.1 準備階段

在這個階段銀行會公佈一些我們機制中會使用的參數，首先令 H，h_1 和 h_2 為單向雜湊函

數，p 和 q 為二個大質數，而 $(p-1)$ 被 q 整除，ρ 是 Z_p^* 的一個生成值。令 $g \equiv_p \rho^{(p-1)/q}$ 為公開參數，$\varpi(x,y,c) \equiv_q h_1(c)x + h_2(c)y$ 為公開多項式。最後銀行隨機產生金鑰對，公鑰為 $y_1 \equiv_p g^{z_1}, y_2 \equiv_p g^{z_2}$，私鑰為 $z_1, z_2 \in Z_q$。

3.2.2 初始階段

在這階段 T 必須產生一對數位假名金鑰，其中公鑰必須經由銀行簽章後，才能在未來的交易過程中使用。在這個過程我們使用 Nyberg-Rueppl 的盲簽章系統 [15] 來完成。之後經由銀行簽章的公鑰與其相對應的私鑰都會存放在 T 中，且私鑰只有 T 知道。詳細的步驟如下：

T 在發行時已經將 d_T 及 $Card_{no}$ 存放在該裝置上，其中 d_T 為該裝置的私鑰，相對應的公鑰 e_T 存放在銀行的資料庫中，$Card_{no}$ 為 T 的出廠序號。T 將請求對數位假名公鑰簽章的訊息傳送給銀行，在該訊息中包含了 $Card_{no}$ 及 $Cert_{d_T}(H(Card_{no}))$，銀行使用 $Card_{no}$ 搜尋資料庫得到與 d_T 相對應的公鑰 e_T 來對 $Cert_{d_T}(H(RD))$ 做驗證，若驗證成功則接受 T 的請求，反之則結束交易。

驗證成功之後銀行隨機選擇 $k_1 \in Z_q$，計算 $\hat{r}_1 \equiv_p g^{k_1}$，使用 e_T 對 $\hat{r}_1$ 加密後傳送 $Cert_{e_T}(H(\hat{r}_1))$ 及 $\hat{r}_1$ 給 T。T 在接收到 $Cert_{e_T}(H(\hat{r}_1))$ 後先使用 d_T 驗證 $\hat{r}_1$ 的正確性。驗證成功後 T 再隨機產生一對數位假名金鑰，公鑰為 e_z，私鑰為 d_z，並隨機產生 $\alpha_1 \in z_q$、$\beta_1 \in z_q^*$，計算 $r_1 \equiv_p H(e_z) g^{\alpha_1} \hat{r}_1^{\beta_1}$ 及 $\hat{m}_1 \equiv_q \beta_1^{-1} r_1$，若 $\hat{m}_1 \neq 0$ 則將 $\hat{m}_1$ 傳送給銀行。銀行在接收到 $\hat{m}_1$ 之後，以銀行的私鑰 d_b 對 $\hat{m}_1$ 做簽章，計算 $\hat{s}_1 \equiv_q \hat{m}_1 d_b + k_1$ 後將 $\hat{s}_1$ 傳送給 T，T 做除盲因子的計算 $s_1 \equiv_q \hat{s}_1 \beta_1 + \alpha_1$ 取得簽章，並計算 $H(e_z) \equiv_p g^{-s_1} y^{r_1} r_1$ 來驗證簽章 s_1 的正確性。若驗證成功 (r_1, s_1) 就是被銀行簽章後的公鑰，我們以 $Cert_{d_b}(H(e_z))$ 來表示，$Cert_{d_b}(H(e_z))$ 與 d_z 都會存放於 T 中供未來交易使用。在完成初始階段後，T 會將 $Cert_{d_T}(H(Card_{no}))$ 及 d_T 從資料庫中刪除，而銀行所存放與其對應公鑰 e_T 也會一併刪除。

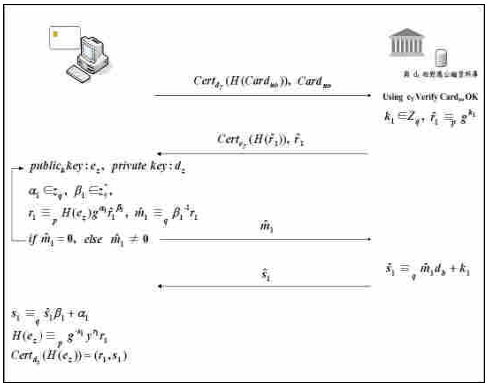

圖 2 初始階段

3.2.3 提款階段

在這個階段中，顧客會透過 T 運用部分盲簽章的方式向銀行提出一個面額為 x 電子現金。在這個過程我們使用 Juang 等學者基於解離散對數問題之部分盲簽章系統 [9] 來完成。

假設顧客的提款日期為 c_1，銀行在顧客提款時產生一對金鑰。其中公鑰為 $y_{c_1} \equiv_p g^{\varpi(z_1,z_2,c_1)} \equiv_p g^{h_1(c_1)z_1 + h_2(c_1)z_2} \equiv_p y_1^{(h_1(c_1))} y_2^{(h_2(c_1))}$，其對應的私鑰為 $\varpi(z_1,z_2,c_1) \equiv_q h_1(c_1)z_1 + h_2(c_1)z_2$，公鑰及私鑰都內嵌了提款日期 c_1。銀行再隨機選擇一個亂數 $k_2 \in Z_q$，計算 $\hat{r}_2 \equiv_p g^{k_2}$，並將 $\hat{r}_2$ 透過安全的通道傳送給 T。T 在接收到 $\hat{r}_2$ 之後做下列步驟：

A. 計算 $H_x(\sigma)$，其中 x 為提款金額，σ 為一隨機亂數，且 $H_0(\sigma) = \sigma$，$H_i(\sigma) = H(H_{i-1}(\sigma))$，$1 \le i \le x$。

B. 隨機產生數位假名私鑰 $z_3 \in Z_q$，以及與其相對應的公鑰 $y_3 \equiv_p g^{z_3}$，並計

算 $m=(RD\|H_x(\sigma)\|P_{ID}\|y_3)$。其中 RD 是包含事先約定訊息的盲訊息，而 P_{ID} 因為經過重排函數 [16] 的運算，其輸入及輸出值會是一對一的結果，不會發生碰撞，這樣可以使得 m 是唯一的，且所產生的盲簽章也會是唯一的。

C. 隨機產生二個亂數 $\alpha_2 \in z_q, \beta_2 \in z_q^*$，並計算 $r_2 \equiv_p mg^{\alpha_2}\hat{r}_2^{\beta_2}$ 及 $\hat{m}_2 \equiv_q \beta_2^{-1} r_2$。

D. 若 $\hat{m}_2 \neq 0$ 則將 $\hat{m}_2$ 傳送給銀行，若不是則重回步驟 B。

銀行在接收到 $\hat{m}_2$ 之後，先檢查該顧客的帳戶是否有足夠的餘額。若帳戶餘額 $<x$ 銀行就拒絕該筆交易。反之若帳戶餘額 $\geq x$ 則使用銀行私鑰對 $\hat{m}_2$ 做簽章，計算 $\hat{s}_2 \equiv_q \hat{m}_2 (z_1, z_2, c_1) + k_2$ 後將 $\hat{s}_2$ 傳送給 T，並從顧客帳戶中扣除 x 元。T 計算 $s_2 \equiv_q \hat{s}_2\beta_2 + \alpha_2$，$(r_2, s_2)$ 就是 m 的部分盲簽章。T 再計算 ${}_p g^{-s_2} y^{r_2}_{c_1} r_2 \equiv m \equiv (\mathrm{RD}\|P_{ID}\|H_x(\sigma)\|y_3)$，來驗證 RD 是否為事先約定的訊息及 $H_x(\sigma)$ 是否為 T 所產生，如果驗證成功則將 $(r_2, s_2, m, c_1, 0, H_x(\sigma), \sigma)$，也就是表示面額為 x 的電子現金存在 T 的資料庫中。

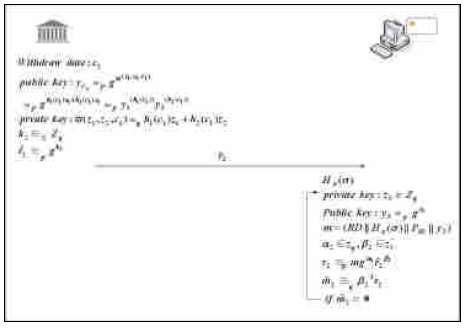

圖 3 提款階段（一）

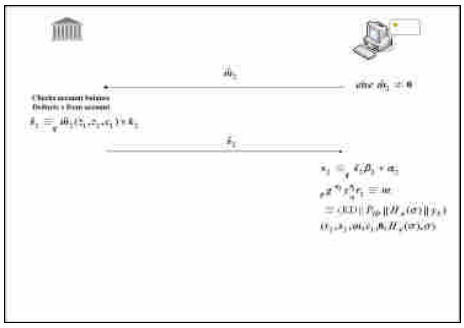

圖 4 提款階段（二）

3.2.4 付款階段

假設本次消費 ε 元，之前已消費 τ 元，且之前已消費金額小於提款金額 $\tau \leq x$。T 首先檢查 $(r_2, s_2, m, c_1, c_3, \tau, h_{x-\tau}(\sigma))$ 是否存在資料庫中且 $\tau+\varepsilon \leq x$，若存在則 T 使用私鑰 d_z 對消費金額 ε、Payword 值 $H_{x-\tau-\varepsilon}(\sigma)$ 及商店代號 ID_S 做簽章，取得 $Cert_{d_z}(H(\varepsilon, H_{x-\tau-\varepsilon}(\sigma), ID_S))$。T 再使用數位假名私鑰 z_3 在電子現金上做消費日期 c_2 的簽章，首先 T 先隨機產生 $k' \in Z_q$，並計算 $r' = (g^{k'} \bmod p) \bmod q$，$s' \equiv_q k'^{-1}(h_1(r_2\|s_2\|c_1\|c_2\|H_{x-\tau-\varepsilon}(\sigma)) + z_3 r')$，最後 T 會儲存這筆電子現金消費記錄 $(r_2, s_2, m, c_1, c_2, \tau+\varepsilon, H_{x-\tau-\varepsilon}(\sigma), \sigma)$ 於資料庫中並傳送 $Cert_{d_b}(H(e_z))$、e_z、$Cert_{d_z}(H(\varepsilon, H_{x-\tau-\varepsilon}(\sigma), ID_S))$、$(r_2, s_2, r', s', m, c_1, c_2, \tau, \varepsilon, H_{x-\tau-\varepsilon}(\sigma))$ 到商家表示要消費 ε 元。

商家首先檢查資料庫是否有該筆資料來防止重複消費，如果沒有，則計算 $m \equiv_p g^{s_2} y^{r_2}_{c_1} r_2 = (RD\|H_X(\sigma)\|P_{ID}\|y_3)$，驗證 m 中的 RD 是否為事先約定的訊息。若 RD 為事先約定的訊息，則計算 $u_1 = h_1(r_2\|s_2\|c_1\|c_2\|H_{x-\tau-\varepsilon}(\sigma))s'^{-1}$，$u_2 = r's'^{-1} \bmod q$，再以驗證公式來驗證正確性，計算 $(g^{u_1} y_3^{u_2} \bmod p) \bmod q = r'$ 是否成立，驗證成功後以銀行的公鑰解 $Cert_{d_b}(H(e_z))$ 驗證公鑰 e_z 是否合法。若驗證成功再以 e_z 解密 $Cert_{d_z}(H(\varepsilon, H_{x-\tau-\varepsilon}(\sigma), ID_S))$，分別驗證消費金額 ε、$H_{x-\tau-\varepsilon}(\sigma)$ 以及商店代號 ID_S 的正確性，計算 $H_x(\sigma) = H_{(\tau+\varepsilon)}(H_{(x-(\tau+\varepsilon))}(\sigma))$ 驗證 Payword 值是否正確及檢查消費日期是否大於提

款日期（$c_2 \geq c_1$），已消費金額加本次消費金額是否小於提款金額（$\tau + \varepsilon \leq x$）。如果都正確，商家記錄該筆電子現金於資料庫中。並計算提款日期到消費日期之間的利息給顧客。待一段時間之後，商家再將所收到的電子現金傳給銀行存入自己的帳戶之中。

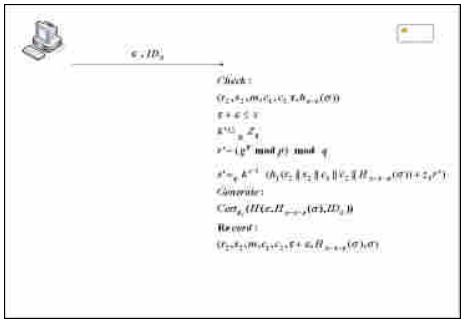

圖 5　付款階段（一）

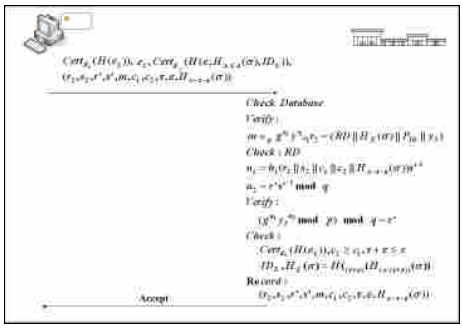

圖 6　付款階段（二）

4.2.5 存款階段

在一段時間之後，商家將電子現金 $(r, s, r', s', m, c_1, c_2, \tau, \varepsilon, H_{x-\tau-\varepsilon}(\sigma))$，$Cert_{d_b}(H(e_z))$，$e_z$，$Cert_{d_z}(H(\varepsilon, H_{x-\tau-\varepsilon}(\sigma), ID_S))$ 以銀行的公鑰 PK_B 加密之後傳送給銀行，銀行以他的私鑰 SK_B 將其解密後，銀行先檢查資料庫是否已記錄兌換過該電子現金。若沒有兌換過，則與執行與商家在 3.2.4 節付款階段相同的驗證過程。如果都正確，銀行將 ε 元再加上 ε 元從提款日期 c_1 到存款當日之間的利息存入商家 ID_S 的帳號之中，並將該電子現金存入銀行資料庫中。

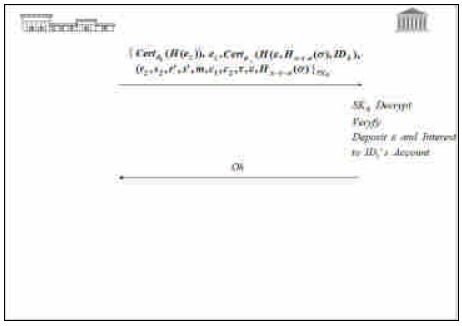

圖 7　存款階段

4 討論

4.1 安全性分析

正確性：

在我們的機制中，透過基於解離散對數問題之部分盲簽章系統 [9] 向銀行提出一個內嵌提款日期的電子現金。之後 T 使用數位假名私鑰 z_3 匿名地將電子現金簽上消費日期 c_2。商家在付款階段時可以使用 $(g^{u_1} y_3^{u_2} \bmod p) \bmod q = r'$ 以及 $m \equiv_p g^{s_2} y^{r_{c_1}} r_2 = (RD \| H_X(\sigma) \| P_{ID} \| y_3)$ 二個驗證公式來做驗證。其驗證公式的正確性可由下列計算式來做證明：

$$
\begin{aligned}
& g^{-s_2} y^{r_{c_1}} r_2 \\
\equiv_p\ & g^{-(\hat{s}_2\beta_2+\alpha_2)} (g^{\varpi(z_1,z_2,c_1)})^{r_2} m g^{\alpha_2+k_2\beta_2} \\
\equiv_p\ & g^{-((\hat{m}_2(\varpi(z_1,z_2,c_1))+k_2)\beta_2+\alpha_2)} g^{\omega(z_1,z_2,c_1)r_2} m g^{\alpha_2+k_2\beta_2} \\
\equiv_p\ & g^{-\omega(z_1,z_2,c_1)r_2-k_2\beta_2-\alpha_2)} g^{\omega(z_1,z_2,c_1)r_2} m g^{\alpha_2+k_2\beta_2} \\
\equiv_p\ & m = (RD \| y_3 \| P_{ID} \| H_x(\sigma))
\end{aligned}
$$

且

$$
\begin{aligned}
& (g^{u_1} y_3^{u_2} \bmod p) \bmod q \\
\equiv\ & (g^{h_1(r_2\|s_2\|c_1\|c_2\|H_{x-\tau-\varepsilon}(\sigma))s^{r-1}} y_3^{r's^{r-1}} \bmod p) \bmod q \\
\equiv\ & (g^{h_1(r_2\|s_2\|c_1\|c_2\|H_{x-\tau-\varepsilon}(\sigma))s^{r-1}} g^{z_3 r's^{r-1}} \bmod p) \bmod q \\
\equiv\ & (g^{h_1(r_2\|s_2\|c_1\|c_2\|H_{x-\tau-\varepsilon}(\sigma))s^{r-1}+z_3 r's^{r-1}} \bmod p) \bmod q \\
\equiv\ & (g^{h_1(r_2\|s_2\|c_1\|c_2\|H_{x-\tau-\varepsilon}(\sigma))+z_3 r')s^{r-1}} \bmod p) \bmod q \\
\equiv\ & (g^{k'} \bmod p) \bmod q \\
\equiv\ & r'
\end{aligned}
$$

不可偽造：

基於解離散對數問題困難度的假設，部分盲簽

章 [9] 是安全的。提款日期 c_1 及未附加消費日期的電子現金只能由銀行產生。顧客若在提款階段偽造了銀行的簽章，那他就可以得到更多的電子現金，但是這違反了公鑰簽章安全存在的假設 [18]。

而在付款階段，消費日期 c_2 只能由 T 亂數產生的數位假名私鑰 z_3 所簽章 $s' \equiv q^{k^{r-1}}(h_1(r_2 \| s_2 \| c_1 \| c_2 \| H_{x-\tau-\varepsilon}(\sigma)) + z_3 r')$，該數位假名私鑰只有 T 知道，所以其他人不可能偽造簽章，若是能偽造則違反了公鑰簽章安全存在的假設 [18]。

如同 3.2.3 節，T 傳送了 $Cert_{d_b}(H(e_z))$、e_z、$Cert_{d_z}(H(\varepsilon, H_{x-\tau-\varepsilon}(\sigma), ID_S)$、$(r_2, s_2, r', s', m, c_1, c_2, \tau, \varepsilon, H_{x-\tau-\varepsilon}(\sigma))$ 到商家表示要消費 ε 元，$\tau + \varepsilon \le x$。若是惡意的顧客可以將電子現金修改為 $Cert_{d_b}(H(e_z))$、e_z、$Cert_{d_z}(H(\omega, H_{x-\tau-\omega}(\sigma), ID_S)$、$(r_2, s_2, r', s', m, c_1, c_2, \tau, \omega, H_{x-\tau-\omega}(\sigma))$ 到商家表示要消費 ω 元，其中 $\varepsilon < \omega$，那麼顧客可以消費超額的電子現金。但這違反了單向雜湊函數的定義 [14][17]、公鑰簽章安全存在的假設 [18] 以及安全的 Tamper-Prove Device [1][2][3] 的假設。

不可否認性：

電子現金在提款階段時，經由銀行私鑰 $\varpi(z_1, z_2, c_1)$ 簽章過之後，才將 $\hat{s}_2 \equiv_q \hat{m}_2 \varpi(z_1, z_2, c_1) + k_2$ 交予 T，T 可以使用銀行的公鑰 y_{c_1} 做驗證。銀行無法否認曾發行該電子現金。而在付款階段，商家同樣地可以使用銀行的公鑰 y_{c_1} 做驗證，銀行也無法否認曾發行該電子現金。再加上 T 使用了數位假名私鑰 z_3 對電子現金做了簽章 $s' \equiv q^{k^{r-1}}(h_1(r_2 \| s_2 \| c_1 \| c_2 \| H_{x-\tau-\varepsilon}(\sigma)) + z_3 r')$，商家可以使用 T 的公鑰 y_3 做驗證，T 無法否認消費該電子現金。

匿名性：

任何人都不知道電子現金是誰使用的，由於電子現金是由銀行以部分盲簽章技術所簽章，當 T 傳送 $\hat{m}_2 \equiv_q \beta_2^{-1} r_2$ 給銀行做簽章，因為 $\hat{m}_2 \equiv_q \beta_2^{-1} r_2$ 含有盲因子，銀行簽署時不能知道所簽署訊息的全部內容，之後也無法得知電子現金與 T 之間的關係，因此可以達到匿名性。

在付款階段，電子現金由 T 隨機產生並經由銀行簽章的數位假名私鑰 z_3 做了簽章。數位假名 [5] 是一個假的身份認證，他結合了銀行的簽章與隨機選擇的公鑰。使用相對應的數位假名私鑰，簽章者可以合法的簽署訊息且不洩露他的身份，因此除了 T 之外沒有任何人知道是誰對該電子現金做簽章，所以可以達到匿名性。

在初始階段 T 是使用 Nyberg-Rueppl 的盲簽章系統 [15] 來取得經銀行簽章的數位假名公鑰 $Cert_{d_b}(H(e_z))$，因此沒有任何人可透過 e_z 來得知是誰做了簽章，因此我們的方案達到了匿名性。

若任何人想從 $P_{ID} = \xi(ID_C \| \lambda)$ 中取得 ID_c 也是不可行的，因為 ζ 是一個重排函數，重排函數是不可逆的，若可以則違反了 [16] 的假設，。

不可重複消費：

我們的方案可達成不可重複消費，因為 T 是一個可信賴、防破壞及防竄改資料的硬體裝置，所以由其產生的資料是可被信任的，再加上經由下列三點的檢查之後商家才會接受交易。

(1) 每次交易時，商家在收到 T 傳來的電子現金後，首先會查詢自己的資料庫中是否已存在該筆電子現金。

(2) 以 $H_{x-\tau}(\sigma)$ 驗證 $H_{x-\tau-\varepsilon}(\sigma)$，計算 $H_x(\sigma) = H_{((\tau+\varepsilon))}(H_{(x-(\tau+\varepsilon))}(\sigma))$（其中提款金額為 x，本次消費 ε 元，之前已消費 τ 元，$\tau + \varepsilon \le x$），也可防止重複消費。

(3) 若是顧客想將電子現金支付給不同商家，雖然 PayWord 的機制無法防止這方面的重複消費，但是在我們的機制中商家可以透過驗證 $Cert_{d_z}(H(\varepsilon, H_{x-\tau-\varepsilon}(\sigma), ID_S)$ 中的 ID_s，確認該筆電子現金的支付對象是否為該商家，基於公鑰簽章安全存在的假設 [18] 及安全的 Tamper-prove device [1][2][3] 的假設，ID_s 是無法被修改的。

無法詐騙利息：

因為電子現金在提款階段時已由銀行將提款日期 c_1 內嵌在電子現金 $(r_2, s_2, m, c_1, 0, H_x(\sigma), \sigma)$

之中，而且消費者在附加消費日期時消費日期 c_2 必須晚於提款日期 c_1。如此，銀行所支付的利息期間就是在提款日期 c_1 到回存日期之間，商家就無法與顧客密謀向銀行詐騙利息。

4.2 比較分析

在基本安全的考量下，令 n 長度為 1024 位元 [12]、p 長度為 1024 位元 [12][13] 以及 q 長度為 160 位元 [13]，在這樣的假設下可讓企圖解離散對數問題及分解因數問題是不可行的。單向雜湊函數 SHA-1 的結果為 160 位元 [14]，盲訊息 m 假設為 1024 位元。經由計算 Chaum [6]、Fan [7]、Chang [4]、Juang [11] 及我們的電子現金大小如下計算。（其中日期的部分不列入計算，括號部分不包含 m 之大小。）

Chaum 學者 [6] 的方案中之電子現金：(m,s)

1024 + 1024 = 2048(1024)

Fan 等學者 [7] 的方案中之電子現金：$(m,s,a_1,a_2,a_3,a_4,a_5,a_6)$

1024 + 1024 + 160 * 6 = 3008(1984)

Chang 等學者 [4] 的方案中之電子現金：(m,δ,s)

1024 + 1024 + 1024 = 3072(2048)

Juang 學者 [11] 的方案中之電子現金：(M,r,s,r',s')

2048 + 1024 + 160 + 160 + 160 = 3552(1504)

其中 $M=(m\|y_3)=2048$

本研究中之電子現金：$(m,r_2,s_2,r',s',\tau,\varepsilon,H_{x-\tau-\varepsilon}(\sigma))$

2240 + 1024 + 160 + 160 + 160＋14 + 14 + 160 = 3932(1692)

其中 $m=(RD\|H_x(\sigma)\|P_{ID}\|y_3)$ 為 2240 位元（RD 為 32 位元、$H_x(\sigma)$ 為 160 位元、P_{ID} 為 1024 位元、y_3 為 1024 位元），在金額方面我們設為 14 位元，最大金額為 16383。因此我們的電子現金大小為 3932(1692) 位元，相較於各位學者的電子現金，我們的電子現金並沒有大很多，並且還多了可分割、可離線交易及可完整計息的功能。

在傳送給商家的訊息方面，其訊息大小的計算如下：假設 c_1 及 c_2 的格式為 2006311，表示 2006 年第 311 天，它的大小為 21 位元。在消費時我們傳送 $(r,s,r',s',m,c_1,c_2,\tau,\varepsilon,H_{x-\tau-\varepsilon}(\sigma)$，$Cert_{d_b}(H(e_z))$，$e_z$，$Cert_{d_z}(H(\varepsilon,H_{x-\tau-\varepsilon}(\sigma),ID_S)))$ 給商家，他的大小為 3932 + 21 + 21 + 1024 + 1024 + 1024 = 7046 (約 881 bytes)。

儲存在 T 中的電子現金是 $(r_2,s_2,m,c_1,c_2,\tau+\varepsilon,H_{x-\tau-\varepsilon}(\sigma),\sigma)$，他的大小為 1024 + 160 + 2240 + 21 + 21 + 14 + 160 + 64 = 3690(約 462 bytes)。

在附加日期方面，Fan 等學者 [7] 所附加的提款日期格式是固定的（$a_1=H^a(x_1)$，$a_2=H^{100-a}(x_2)$，$a_3=H^b(x_3)$，$a_4=H^{12-b}(x_4)$，$a_5=H^c(x_5)$，$a_6=H^{31-c}(x_6)$），而 Chang 等學者 [4]、Junag 學者 [11] 及我們的方案附加的提款日期格式是彈性的。在 Fan 等學者 [7] 的方案中，消費日期是由顧客來附加的，所以商家可以與顧客密謀將消費日期提前來向銀行詐騙更多的利息，而且當顧客附加消費日期後就無法更改消費日期。

Chang 等學者的方案 [4] 為了附加消費日期，顧客在除盲因子階段需向銀行請求另一個簽章 $\delta=r_2^{-1}t_2 \bmod n^*$，且銀行需在附加日期階段透過不可追蹤電子郵件 [5] 傳送 $s'=G^{d^*}(s\|a\|b\|c) \bmod n^*$ 給顧客。由於消費日期是由銀行來做附加，所以商家無法與顧客密謀向銀行詐騙更多利息。也由於消費日期是由銀行來做附加，所當顧客想要更換消費日期時，只要請求銀行重新做簽章即可。

Junag 學者 [11] 與我們的方案因為電子現金在提款階段時已由銀行將提款日期內嵌在電子現金 $(r_2,s_2,m,c_1,0,H_x(\sigma),\sigma)$ 之中，而且消費者在附加消費日期時，消費日期 c_2 必須晚於提款日期 c_1。因此，商家就無法與顧客密謀向銀行騙取更多的利息。而消費日期是由顧客來做附加，所以當顧客想要更換消費日期時，只要顧客重新做簽章即可。

我們的方案使用了 Tamper-Proof Device 及 Payword 的機制，交易不需即時將電子現金傳送至銀行做驗證，因此可以採用離線交易，有別於其他學者的方案大多是連線交易。除此之外，因為使用了 Payword 的機制，讓顧客可以消費任意金額，因此我們的電子現金還達到了可分割性，且顧客還可以跟任意特約商店進行消費，

在重複消費的問題方面，我們的方案中在 3.2.4 節提款階段已經由商家來檢查，因此銀行不需再做重複消費的檢查。但是為了防止惡意的商家，重複傳送電子現金存入，所以銀行仍需將電子現金存入資料庫，用來檢查商家是否重複傳送電子現金來存入。但為了避免銀行的資料庫過於龐大，可規定商家必須在消費日期 c_2 之後一定天數之內向銀行兌換，這樣銀行就可以將超過期限的電子現金從資料庫中刪除，以節省儲存空間。

表三　本研究與相關學者方案之比較表

	Chaum [6]	Fan [7]	Chang [4]	Juang [11]	本研究
C1	X	X	X	O	O
C2	X	O	O	O	O
C3	X	Fix	Free	Free	Free
C4	X	X	O	O	O
C5	X	X	O	O	O
C6	X	572次 Hash 註1	簽章 δ, s'	簽章 s	簽章 s
C7	X	X	X	O	O
C8	O	O	X	O	O
C9	m, s	m, s ,註2	m, s, δ	m, s, r, s', r'	註3
C10	2048	3008	3072	3552	3932
C11	1024	1984	2048	1504	1692
C12	X	△	△	O	O
C13	O	O	O	O	O
C14	O	O	O	O	O
C15	O	O	O	O	O
C16	O	O	O	O	O
C17	X	X	X	X	O
C18	X	X	X	X	O
C19	X	X	X	X	O

符號說明： O: 表示有達到　X: 表示沒有有達到
△: 表示部分達成

C1:提款日期
C2:消費日期
C3:消費日格式
C4:合法修改消費日期
C5:無法詐騙利息
C6:附加日期的方式
C7:隨機簽名
C8:不需使用不可追蹤的電子郵件
C9:電子現金不包含日期
C10:電子現金不包含日期的大小
C11:電子現金不包含 m 及日期的大小
C12:可計息
C13:匿名性
C14:不可重複消費
C15:不可否認性
C16:不可偽造
C17:可離線消費

C18:可分割

C19:銀行不需要做重複費的檢查

註1： 可能為 572 或 1864 次。

註2： $a_1 = H^a(x_1)$，$a_2 = H^{100-a}(x_2)$，$a_3 = H^b(x_3)$，$a_4 = H^{12-b}(x_4)$，$a_5 = H^c(x_5)$，$a_6 = H^{31-c}(x_6)$)。

註3： $m, r_2, s_2, r', s', \tau, \varepsilon, H_{x-\tau-\varepsilon}(\sigma)$。

4.3 討論

在我們的方案中，在初始階段，銀行使用的金鑰 d_b，原本想以在準備階段中產生的私鑰 $z_1, z_2 \in Z_q$ 來取代，以減少金鑰的使用，方便管理。但因考慮到每把金鑰都應有其專屬的功能，例如 d_b 專屬於對 T 的公鑰做簽章，而 $z_1, z_2 \in Z_q$ 專屬於對電子現金做簽章，所以最後決定還是分為二對不同的金鑰。

關於商家如何將利息交給顧客的部分，商家可以在付款階段顧客支付電子現金後先計算出利息金額，直接給予顧客等值的電子現金；或者是在顧客支付電子現金前先計算出利息金額，直接從顧客支付的金額中扣除。

在 3.2 節中我們討論到我們的方案達到了匿名性 (Anonymity)，也就是沒有任何人可以知道是誰使用了該電子現金來消費。另外有個比較強的匿名，稱之為不可連結性 (Unlinkability)，不可連結性是指任何人無法從二筆交易來得知二筆交易之間的關係。在我們的方案中，由於使用了 PayWord 機制來分割電子現金，商家或銀行可透過消費金額及 PayWord 來得知二筆交易是可能是有關聯的。如同我們在付款階段中所提到的前一次消費的電子現金 $(r_2, s_2, m, c_1, c_3, \tau, h_{x-\tau}(\sigma))$ 及本次所消費的電子現金 $(r_2, s_2, m, c_1, c_2, \tau+\varepsilon, H_{x-\tau-\varepsilon}(\sigma), \sigma)$，商家或銀行可以從這二筆電子現金看出這二筆電子現金可能是同一個顧客所使用的，但商家或銀行仍然無法得知是那個顧客使用了這二筆電子現金，所以我們的方案達到了匿名性而沒有達到不可連結性。

在 3.2.4 節存款階段中，T 直接將電子現金 $(r, s, r', s', m, c_1, c_2, \tau, \varepsilon, H_{x-\tau-\varepsilon}(\sigma))$ 傳送給商家。其中消費金額的部分為明文，這可能會讓有心人士可從中得知商家的營業額。若是商家很重視營業額的機密，可再多增一對金鑰，用來保護 T 與商家之間訊息的內容。T 可在傳送電子現金前先將電子現金以商家公鑰加密後再傳送，以防止營業額的外洩。

5.結論與未來研究方向

本論文提出了一個離線交易且可計息的電子現金，讓顧客享有提出電子現金到消費電子現金期間的利息，而商家則享有顧客消費電子現金到商家回存電子現金至銀行期間的利息。為了降低商家與銀行連線驗證電子現金的成本，我們的機制採取離線交易，商家可以等到離峰的時間再與銀行以批次的方式來存入電子現金。這樣不但可以節省商家及銀行對於頻寬的需求，也可降低硬體設備建立的成本，商家與銀行也就比較容易且願意加入本研究的機制。越來越多商家與銀行的參與，在網路上付款也就更便利，也就能幫助電子商務發展地更加順利。更重要的是可以防止因為網路過於擁塞或網路斷線所造成無法交易的情形發生而造成損失，而這方面的損失通常也是無法估計的。根據第五章的分析，我們的方案也不因採用離線交易而未達到一個電子現金應該有的安全性，如匿名性，不可否認性、不可重複消費、不可偽造等等。

大部分的電子現金是不可分割使用的，也就是一個面額 100 元的電子現金，只能一次消費完，不能只消費 50 元而找回剩下 50 元，這與日常生使用現金的習慣不相同，也非常的不方便。而我們的方案可使電子現金分割使用，也就是能消費任意想花費的金額，這樣的做法與現實生活的現金使用方式較為相似，也比較方便。因此，我們的電子現金達到了我們的研究目的：一個安全、便利、又能吸引人使用的電子現金系統。

由於本方案使用的是基於解離散對數問題的部分盲簽章及 DSA ，可將這二個部分都改用橢圓曲線系統。橢圓曲線系統在相同的安全程度之下，

相較於同樣是離散對數的系統會比較有效率，金鑰長度也會比較短，所以可以減少本機制運算及通訊上的成本，而 ECDSA 也較適合使用在智慧卡上。另外也可參考 ISO 7816 [8] 的標準來設計出適合本方案使用的 Tamper-Proof Device，這些都是未來可以繼續研究的方向。

參考文獻

[1] Milton. M. Anderson (1998), "The Electronic Check Architecture", Technical Report Version 1.0.2, Financial Services Technology Consortium, September.

[2] J. Boly, A. Bosselaers, R. Cramer, R. Michelsen, S. Mjolsnes, F. Muller, T. Pedersen, B. Pfitzmann, P. de Rooij, B. Schoenmakers, M. Schunter, L. Vallee and M. Waidner (1994), "The ESPRIT Project CAFE-High Security Digital Payment Systems", *Proceedings of ESORICS 94*, Springer, Berlin, Vol. 857, pp. 217-230.

[3]S. Brands (1994), "Untraceable Off-line Cash In Wallets With Observers", *Advances in Cryptology-Crypto'93*, Springer, Berlin, Vol. 773, pp. 302-318.

[4] C. C. Chang and Y. P. Lai (2003), "A flexible Date-attachment Scheme on E-cash", *Computers and Security*, Vol. 22, No. 2, pp. 160-166.

[5]D. Chaum (1981), "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", *Communications of the ACM*, Vol. 24, No. 2, pp. 84-88.

[6] D. Chaum (1983), "Blind Signatures for Untraceable Payments", *Advances in Cryptology - CRYPTO'82,* Springer-Verlag , pp. 199-203.

[7] C. I. Fan, W. K. Chen and Y. S. Yeh , "Date Attachable Electronic Cash", *Computer Communications*, Vol. 23, No. 4, pp. 425-428, 2000.

[8] International Organization for Standardization (2007/05/30), http://www.iso.org/iso/en/CombinedQueryResult.CombinedQueryResult?queryString=7816.

[9] W. S. Juang (1999), C .L. Lei, "Partially Blind Threshold Signatures Based on Discrete Logarithm", *Computer Communication,* Vol. 22, No. 1, pp.73-86.

[10] W. S. Juang, "A Practical Anonymous Off-line Multi-authority Payment Scheme (2005)," *Electronic Commerce Research and Applications*, Vol. 4, No. 3, pp. 240-249 .

[11] W. S. Juang (2007), "D-Cash: A Flexible Pre-paid E-cash Scheme for Date-attachment", *Electronic Commerce Research and Applications*, Vol. 6, No. 1, pp. 74-80.

[12] A. Lenstra, E. Tromer, A. Shamir, W. Kortsmit, B. Dodson, J. Hughes and P. Leyland, "Factoring Estimates for A 1024-bit RSA Modulus" (2003), *Advances in Cryptology-ASIACRYPT 2003*, Springer, Berlin , Vol. 2894, pp. 55-74.

[13] NIST FIPS PUB 186-2 (2001), "Digital Signature Standard, National Institute of Standards and Technology", *US Department of Commerce*.

[14] NIST FIPS PUB 180-2 (2004), "Secure Hash Standard, National Institute of Standards and Technology", *US Department of Commerce*.

[15] K. Nyberg and R.A. Rueppel (1995), "Message Recovery for Signature Schemes Based on The Discrete Logarithm Problem", *Advances in Cryptology-Eurocrypt'94*, Springer, Berlin, Vol. 950, pp. 182-193.

[16] S. Pohlig, and M. Hellman (1978), "An Improved Algorithm for Computing Logarithms over GFp and Its Cryptographic Significance", *IEEE Transactions on Information Theory*, IT-24, pp. 106-110.

[17] R. Rivest and A. Shamir (1997), “PayWord and MicroMint: Two Simple Micropayment Schemes”, *Security Protocols*, Vol. 1189, pp. 69-87.

[18] R. Rivest, A. Shamir, and L. Adleman (1978), “A Method for Obtaining Digital Signatures and Public Key Cryptosystems”, *Communications of the ACM*, Vol. 21, No. 2, pp.120-126.