智慧卡下有效率具不可追蹤性之密碼認證機制

李鴻璋*、趙國全
淡江大學資訊管理學系
E-mail*:hclee@mail.im.tku.edu.tw

摘要

近年來，隨著智慧卡在身份敏感的登入認證系統之應用，有效率的互相認證與會議鑰匙產生協議是重要的安全議題。2010 年，Li 等研究提出了對 JCL 演算法的修正版，以滿足匿名性及不可追蹤性，加強了使用者身份的保護，但 Li 等及 JCL 方法中，在雙方通訊過程所伴隨的高溝通和運算成本，與智慧卡晶片處理器的處理限制不符。且為了達到不可追蹤性之安全水準，必須符合在 Li 等演算法認證資訊的設計條件下，降低了登入認證系統彈性，因此本論文針對 Li 等演算法加以改進，讓智慧卡登入認證系統在滿足不可追蹤性下，並具有更好的效能與系統設計彈性。

本論文提出一個以亂數方式產生動態密碼的演算法，結合了線性反饋位移暫存器(LFSR)的虛擬亂數(Pseudorandom)產生演算法之概念，將每回合產生具不可追蹤性之亂數當作登入密碼。在登入階段，以動態密碼來做為互相認證的因子，通過互相認證產生會議鑰匙減少了三分之一的通訊回合。

在效能評估上，透過認證過程所需消耗的溝通成本與運算成本，以及各階段實際執行時間進行比較。實驗結果顯示，本論文相較於 Li 等演算法，不僅同樣滿足匿名性和不可追蹤性，在登入階段減少了通訊回合數，降低溝通和運算成本，如此，可達到效率及效能兼具的智慧卡登入認證系統。

關鍵詞:智慧卡、互相認證、匿名性、動態密碼、線性反饋位移暫存器

1. 緒論

1.1 研究背景

網路環境下的遠端服務系統下，使用者需要遠端登入伺服器來要求服務，使用者身份認證就是相當重要的課題，傳統以靜態密碼為基礎的登入認證系統，可能存在離線字典猜測攻擊或重送攻擊等密碼猜測攻擊的風險，攻擊者通過身份認證，偽裝成合法使用者來達成會議鑰匙協議取得服務。為解決此種風險，動態密碼(OTP, One Time Password)在網路安全交易機制中身份認證的應用[4],[5],[7],[9]，相較於靜態密碼，具不可預測性，可有效認證訊息傳輸方的身份，防止攻擊者竊取認證資訊執行重送攻擊的風險，攻擊者無法通過下次的登入認證。

安全的身份認證方法一般會要求在下面三種特性中，選擇兩種的組合來進行認證，稱為二因子認證(2-Factor Authentication)[2],[13]，動態密碼即滿足第一及第二項特性。

身份認證需具備下列特性:

(1) 使用者所知的資訊，如帳號、密碼、PIN 碼。
(2) 使用者所擁有的物件，如金鑰、憑證、密碼產生器。
(3) 使用者所擁有的特徵，如生物特徵的指紋、五官、掌形。

動態密碼即每次使用時產生的密碼都不相同，故又稱一次性密碼，動態密碼是在離線狀態下使用虛擬亂數產生器(PRNG, Pseudorandom Number Generator)來產生[14]，因此必須和認證伺服器達到時間同步，於網路交易每次使用皆不同，並只限定一次有效。密碼以隨機方式產生，具備虛擬亂數的特性，理論上可透過這回合的密碼來產生下一回合的密碼，意即密碼間存在數學相關。

虛擬亂數產生器產生與真實亂數相近的亂數循環[14]，虛擬亂數循環的產生與輸入起始種子值或目前狀態相依，攻擊者可

能透過追蹤種子值來預測亂數值，存在安全性風險,因此為了產生接近真實的亂數，虛擬亂數產生器需具備密碼學的效力，目前已有許多符合密碼學的虛擬亂數產生演算法如 B.B.S.(Blum Blum Shub)、Fortuna 及 Mersenne twister，而後基於效能上的考量，密碼學的虛擬亂數產生演算法逐漸被硬體概念的亂數產生器所取代，如 LCG LCG(Linear Congruential Generators)、LFG (Lagged Fibonacci Generator)、LFSR(Linear Feedback Shift Registers)及 FCSR(Feedback with Carry Shift Registers)。本研究基於智慧卡下處理器限制的考量，選擇以 LFSR 做為動態密碼產生器[1],[10],[12],LFSR 已被證實能快速產生虛擬亂數，與本研究限制條件相符。

Li 等研究提出了滿足 JCL 演算法匿名性，並延伸不可追蹤性的登入認證系統[6], [11]，不可追蹤性為匿名性的延伸，更加強了使用者身份的保護，同時在密碼更換階段簡化了阻斷服務攻擊(Denial of Service)的驗證，安全的登入認證系統通常具有下列特性:

(1) 互相認證(Mutual Authentication): 使用者和認證伺服器通訊，雙方都必須確認訊息是來自合法的一方。

(2) 會議鑰匙協議(Session Key Agreement):認證雙方或多方必須協議出一把共同的祕密鑰匙，之後祕密通訊皆使用此把鑰匙加密，其他方無法竊取。

(3) 匿名性(Anonymity):只有認證伺服器知道登入系統的使用者身份，其他使用者 無法得知。

(4) 不可追蹤性(Untraceability):攻擊者不知使用者何時開始與認證伺服器進行通訊與進行幾次通訊，無法藉由認證過程中雙方通訊的訊息，來判斷使用者的身份。

除了上述特性，並能有效防止重送攻擊(Replay Attack)、內部攻擊(Inside Attack)、離線密碼猜測攻擊(Off-line Password-Guessing Attack) 及簡化阻斷服務攻擊(DOS)的驗證。

關於「不可追蹤性」此點，過去針對匿名性來隱藏使用者身份，雖在登入認證階段使用者身份資訊並沒有以明文方式傳輸，但使用者與認證伺服器的通訊規則，攻擊者可從其中找出關聯來判斷使用者身份，Li 等演算法改善 JCL 演算法[6],[11]，讓第三方無法得知哪一位使用者曾經和認證伺服器進行一次以上的通訊，讓通訊過程不會有洩漏身份的風險。

1.2 研究動機和目的

Li 等演算法為了達到「不可否認性」[11]，並簡化「阻斷服務攻擊」的驗證，必須在認證資訊的設計條件下，降低了登入認證系統彈性，在智慧卡處理器與認證伺服器的通訊過程，需要大量加解密及雜湊運算，伴隨過高的溝通及運算成本，對嵌入系統或手持裝置處理器產生過高的負載，使 smart card 處理器的限制在高效能應用上產生爭議。

本研究針對「不可追蹤性」進行深入探討，以期能就建構出一套高效能並符合安全特性的登入認證系統，讓合法使用者在登入系統要求服務時，不需額外安裝驅動程式及憑證存取,提供方便的使用方式，伺服器能快速認證使用者身份，故特設計此演算法，不僅符合網路安全交易機制且具有高效能的智慧卡應用。

本研究提出以動態密碼做身份認證的因子，在滿足「不可追蹤性」下，於註冊階段，選擇線性反饋位移暫存器(LFSR)演算法起始種子來設定起始狀態[1],[10],[12]。然後在第 i 次登入階段(i 從 1 開始)產生下一回合的動態密碼，認證伺服器確認無誤再產生新的動態密碼進行認證，使密碼訊息產生隨機性。

2. 文獻探討

2.1 JCL 演算法

JCL 演算法依據 Fan et al.演算法為基礎[3],[6]，提出了一套智慧卡下強化身份認

證及會議鑰匙協議的登入認證系統，滿足「匿名性」的要求，對使用者的身份做到保護，並提供以下安全特性:1)不需要密碼儲存的資料表;2)使用者可自由選擇密碼；3)無時間同步(time-synchronization)問題；4)防止重送攻擊(replay attack)；5)離線字典攻擊(off-li ne dictionary attack)；用橢圓曲線(Elliptic Curve)密碼標準產生會議鑰匙協議的因子，降低 Fan et al.演算法的溝通和運算成本。

JCL 演算法必需定義下列名詞:

- h(・):安全雜湊函數。
- E_s(・):透過伺服器密鑰做對稱式加解密。
- ||:字串連接。
- p:一個大的質數。
- E_p:在Z_p集合中的橢圓曲線方程式。
- G:序列的中心點。
- (x, P_s):橢圓曲線密碼標準的私鑰公鑰對。

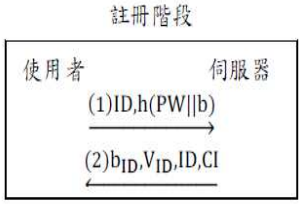
JCL 演算法總共分為五個階段(參數產生、註冊、前置運算、登入以及密碼更換階段)[6]，橢圓曲線方程式用來做身份認證的參數皆在參數產生階段由伺服器產生。註冊階段為登入認證伺服器要求服務的使用者做身份註冊，伺服器會發給使用者一智慧卡確認編號 CI 及以伺服器密鑰 s 加密的認證資訊b_{ID}供登入時使用。前置運算階段利用參數產生階段之參數，以橢圓曲線方程式運算來產生雙方通訊會議鑰匙之組成因子。登入階段使用者使用b_{ID}及使用共同密鑰V_{ID}加密前置運算階段產生參數傳輸給伺服器，雙方經由三回合的通訊產生會議鑰匙，做為以後秘密通訊之用。密碼更換階段使用者利用會議鑰匙來更換適合的密碼。

2.1.1 參數產生階段

此階段伺服器需產生橢圓曲線方程式之參數[8]，伺服器挑選一個大的質數 p，並挑選兩個參數$a, b \in Z_p$，a 和 b 需滿足$4a^3 + 27b^2 (mod\ p) \neq 0$，橢圓曲線方程式為$E_p: y^2 = x^3 + ax + b$。伺服器尋找序列的中心點G和亂數x來做私鑰，公鑰為$P_s = xG$，然後發佈參數P_s, p, E_p, G給使用者。

2.1.2 註冊階段

為使用者在登入認證伺服器前，使用者身份 ID 及選擇的密碼 PW 與一個亂數 b 進行雜湊運算，將$(ID, h(PW||b))$訊息傳送給認證伺服器，當伺服器收到後產生一智慧卡確認編號 CI，用來確認持有智慧卡使用者的身份，如果是新的使用者伺服器則設定 CI=1，若伺服器發現使用者智慧卡確認編號已有註冊過，則設定 CI=CI+1，伺服器將(ID, CI)儲存在認證資料表中。伺服器使用密鑰產生$b_{ID} = E_s(h(PW||b)||ID||CI||h(ID||CI||h(PW||b))$及$V_{ID} = h(ID||s||CI)$，將$(ID, CI, b_{ID}, V_{ID})$訊息傳送給使用者的智慧卡，將$(ID, CI, b_{ID}, V_{ID}, b)$儲存在智慧卡記憶體中。圖 1 為 JCL 演算法註冊階段流程圖。



(1) $ID, h(PW||b)$
(2) b_{ID}, V_{ID}, ID, CI
$b_{ID} = E_s(h(PW||b)||ID||CI||h(ID||CI||h(PW||b))$
$V_{ID} = h(ID||s||CI)$

圖 1 JCL 演算法註冊階段流程圖

2.1.3 前置運算階段

智慧卡選擇一亂數 r，計算$e = rG, c = rP_s (= rxG)$做為E_p下的點，將(e, c)兩點儲存在智慧卡中供登入階段使用。

2.1.4 登入階段

登入階段，使用者不用輸入密碼，只需再登入階段前完成前置運算階段即可。

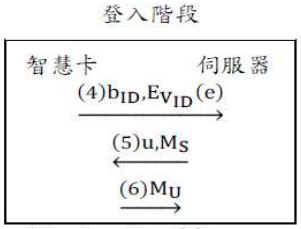
1 使用者使用智慧卡和認證伺服器通訊時，傳送b_{ID}及$E_{V_{ID}}(e)$給伺服器。

2 伺服器用密鑰s解密b_{ID}來獲得

$h(PW||b)||ID||CI||h(ID||CI||h(PW||b))$，用來和註冊資料表$(ID, CI)$比對確認使用者身份，計算$h(ID||CI||h(PW||b))$驗證是否為合法使用者及確保資料完整性，並使用$V_{ID}$解密$E_{V_{ID}}(e)$取得e，當通過伺服器認證，伺服器選擇一亂數u，計算$c = ex$和$M_S = h(c||u||V_{ID})$，將 u 及M_S傳送給智慧卡。

3 智慧卡收到 u 及M_S計算M_S是否等於$h(c||u||V_{ID})$確認是否為合法伺服器，如果通過認證，智慧卡計算$M_U = h(h(PW||b)||V_{ID}||c||u)$和會議鑰匙$k = h(V_{ID}||c||u)$，智慧卡傳送$M_U$給伺服器。

4 伺服器收到M_U計算M_U是否等於$h(h(PW||b)||V_{ID}||c||u)$，通過認證後伺服器計算會議鑰匙$k = h(V_{ID}||c||u)$，智慧卡和伺服器即可使用會議鑰匙來進行安全的秘密通訊。圖 2 為 JCL 演算法登入階段流程圖。

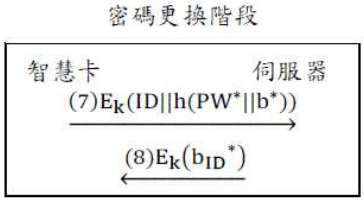


(4) $b_{ID}, E_{V_{ID}}(e)$
(5) u, M_S
$M_S = h(c||u||V_{ID})$
(6) M_U
$M_U = h(h(PW||b)||V_{ID}||c||u)$

圖 2 JCL 演算法登入階段流程圖

2.1.5 密碼更換階段

使用者想要執行更換密碼必須通過登入階段互相認證後會議鑰匙的產生，使用者想更換密碼 PW 和亂數 b，智慧卡使用會議鑰匙 k 計算$E_k(ID||h(PW^*||b^*))$傳送給伺服器，伺服器計算 $E_k(b_{ID}{}^*)$傳送給智慧卡，$b_{ID}{}^* = E_s(h(PW^*||b^*)||ID||CI||h(ID||CI||h(PW^*||b^*))$當智慧卡收到$E_k(b_{ID}{}^*)$用 k 解密得到$b_{ID}{}^*$，將$b_{ID}{}^*$及$b^*$儲存在智慧卡中並取代原先的$b_{ID}$和 b，讓智慧卡下一次登入時使用。圖 3 為 JCL 演算法密碼更換階段流程圖。



(7) $E_k(ID||h(PW^*||b^*))$
(8) $E_k(b_{ID}{}^*)$
$b_{ID}{}^* = E_s(h(PW^*||b^*)||ID||CI||h(ID||CI||h(PW^*||b^*))$

圖 3 JCL 演算法密碼更換階段流程圖

2.2 Li 等演算法

Li 等演算法同樣分為五個階段(參數產生、註冊、前置運算、登入、密碼更換階段)[11]，其中參數產生及前置運算兩階段同 JCL 演算法。Li 等演算法以 JCL 演算法為基礎，不僅繼承了滿足 JCL 演算法「匿名性」互相認證的會議鑰匙協議，更提供了「不可追蹤性」來加強使用者身份的保護，通訊過程第三方無法確認智慧卡與認證伺服器是否有一次以上的通訊，防止攻擊者藉由通訊過程的訊息來追蹤使用者身份，在密碼更換階段更提供「阻斷服務攻擊」驗證的機制，避免使用者下次登入因攻擊者的偽造資訊，造成互相認證的失敗。

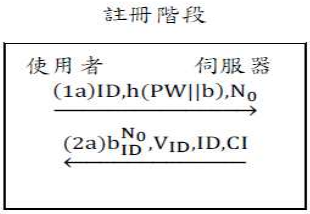
2.2.1 註冊階段

註冊階段除了使用者身份及密碼和亂數雜湊運算組成的訊息外在加上一固定亂數(nonce)N_0，將$(ID, h(PW||b), N_0)$訊息傳送給認證伺服器，當伺服器收到後同樣產生智慧卡確認編號 CI，將(ID, CI, N_0)儲存在註冊資料表中，並使用密鑰 s 產生$b_{ID}^{N_0}$和V_{ID}，$b_{ID}^{N_0}$會因N_0而具有隨機性，

$$b_{ID}^{N_0} = E_s((ID||CI||N_0)||(ID||CI||N_0) \oplus h(PW||b)||h((ID||CI||N_0)||((ID||CI||N_0) \oplus h(PW||b))))$$

伺服器將$(b_{ID}^{N_0}, V_{ID}, ID, CI)$訊息傳送給智慧卡，智慧卡記憶體儲存$(b_{ID}^{N_0}, V_{ID}, ID, CI, b)$。

圖 4 為 Li 等演算法註冊階段流程圖。



(1a) $ID, h(PW||b), N_0$

(2a) $b_{ID}^{N_0}, V_{ID}, ID, CI$

$$b_{ID}^{N_0} = E_s((ID||CI||N_0)||(ID||CI||N_0) \oplus h(PW||b)||h((ID||CI||N_0)||((ID||CI||N_0) \oplus h(PW||b))))$$

$$V_{ID} = h(ID||s||CI)$$

圖 4 Li 等演算法註冊階段流程圖

2.2.2 登入階段

1 智慧卡和認證伺服器通訊時，傳送$b_{ID}^{N_0}$及$E_{V_{ID}}(N_1||e)$給伺服器，N_1為一固定亂數。

2 伺服器收到訊息後，解密$b_{ID}^{N_0}$獲得$(ID||CI||N_0||h(PW||b)||h((ID||CI||N_0)||((ID||CI||N_0) \oplus h(PW||b)))$，與註冊資料表$(ID, CI, N_0)$比對並計算$h((ID||CI||N_0)||((ID||CI||N_0) \oplus h(PW||b)))$進行智慧卡身份認證及確保資料完整性，解密$E_{V_{ID}}(N_1||e)$取得$N_1$及 e，通過認證後，伺服器更新註冊資料表$(ID, CI, N_1)$。

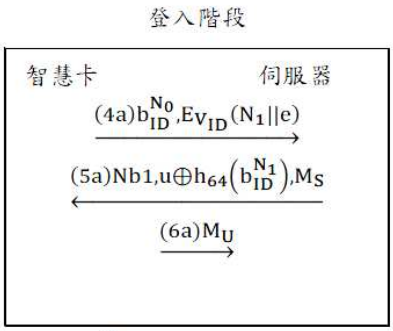
3 伺服器選擇一亂數 u，計算$c = ex$及$M_S = h(c||u||V_{ID})$，將$(Nb1, u \oplus h_{64}(b_{ID}^{N_1}), M_S)$訊息傳送給智慧卡，$h_{64}(b_{ID}^{N_1})$為$h(b_{ID}^{N_1})$前 64 位元。

$$b_{ID}^{N_1} = E_s((ID||CI||N_1)||\oplus h(PW||b)||h((ID||CI||N_1)||((ID||CI||N_1) \oplus h(PW||b))))$$

$$Nb1 = b_{ID}^{N_1} \oplus (h(N_1||e||1)||h(N_1||e||2)||h(N_1||e||3))$$

4 智慧卡收到訊息後，計算$Nb1 \oplus (h(N_1||e||1)||h(N_1||e||2)||h(N_1||e||3))$取得$b_{ID}^{N_1}$，透過$u \oplus h_{64}(b_{ID}^{N_1})$取得u，智慧卡即可計算$M_S = h(c||u||V_{ID})$來確證是否為合法伺服器，如通過認證，智慧卡更新記憶體用$b_{ID}^{N_1}$取代$b_{ID}^{N_0}$，並計算$M_U = h(h(PW||b)||V_{ID}||c||u)$傳送給伺服器。

5 伺服器收到M_U後，同樣計算M_U是否等於$h(h(PW||b)||V_{ID}||c||u)$，完成互相認證，產生智慧卡和伺服器中會議鑰匙$k = h(V_{ID}||c||u)$。圖 5 為 Li 等演算法登入階段流程圖。



(4a) $b_{ID}^{N_0}, E_{V_{ID}}(N_1||e)$

(5a) $Nb1, u \oplus h_{64}(b_{ID}^{N_1}), M_S$

$$b_{ID}^{N_1} = E_s((ID||CI||N_1)||(ID||CI||N_1) \oplus h(PW||b)||h((ID||CI||N_1)||((ID||CI||N_1) \oplus h(PW||b))))$$

$$Nb1 = b_{ID}^{N_1} \oplus (h(N_1||e||1)||h(N_1||e||2)||h(N_1||e||3))$$

$$M_S = h(c||u||V_{ID})$$

(6a) M_U

$$M_U = h(h(PW||b)||V_{ID}||c||u)$$

圖 5 Li 等演算法登入階段流程圖

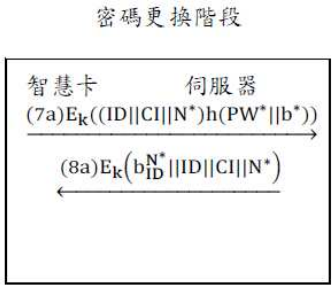
2.2.3 密碼更換階段

同 JCL 演算法以登入階段經互相認證產生的會議鑰匙，智慧卡選擇密碼PW^*和亂數b^*及一個新的固定亂數N^*，計算$E_k((ID||CI||N^*)h(PW^*||b^*))$傳送給伺服器，伺服器解密後獲得$(ID, CI, N^*, h(PW^*||b^*))$，更新註冊資料表$(ID, CI, N^*)$，伺服器計算$E_k(b_{ID}^{N^*}||ID||CI||N^*)$訊息傳送給智慧卡，

$$b_{ID}^{N^*} = E_s((ID||CI||N^*)||(ID||CI||N^*) \oplus h(PW^*||b^*)||h((ID||CI||N^*)||((ID||CI||N^*) \oplus h(PW^*||b^*))))$$

智慧卡解密訊息後更新記憶體$(b_{ID}^{N^*}, ID, CI, N^*)$。此方法驗證「阻斷服務攻擊」，避免攻擊者假冒伺服器發送垃圾訊息給智慧卡，智慧卡更新記憶體內容後，下次登入要求

服務導致認證失敗。圖 6 為 Li 等演算法密碼更換階段流程圖。



(7a) $E_k((ID||CI||N^*)h(PW^*||b^*))$

(8a) $E_k(b_{ID}^{N^*}||ID||CI||N^*)$

$$b_{ID}^{N^*} = E_k((ID||CI||N^*)||(ID||CI||N^*) \oplus h(PW^*||b^*)||h((ID||CI||N^*)||((ID||CI||N^*) \oplus h(PW^*||b^*))))$$

圖 6 Li 等演算法密碼更換階段流程圖

2.3 各相關文獻之探討

JCL 演算法和 Li 等演算法系統架構，JCL 演算法提出的登入身份認證系統，是以「匿名性」為基礎，在註冊階段以伺服器密鑰將使用者身份資訊加密，智慧卡與伺服器通訊過程，第三方無法得知使用者身份，卻無法滿足「不可追蹤性」，攻擊者可以透過智慧卡與伺服器通訊次數，來推測使用者身份，並存在「阻斷服務攻擊」的風險。Li 等演算法提出改良方法來滿足「不可追蹤性」，並達成「阻斷服務攻擊」的驗證，伺服器加密使用者資訊會加入一固定亂數，使加密的認證資訊皆不相同，通訊過程要求使用新的固定亂數加密產生新的認證資訊，做下一次登入時使用，攻擊者無法從相同的認證資訊，來判斷使用者身份，因此攻擊者無法仿冒認證資訊，導致登入更新智慧卡記憶體後登入認證失敗。

我們發現 Li 等演算法中認證資訊b_{ID}^{N}將使用者身份 ID 及 CI 加密於第一區塊密文，透過固定亂數 N 使的b_{ID}^{N}產生隨機化，並選擇 AES 加密演算法做對稱式加解密，並以密文區塊串接模式做處理，因為 AES 每個區塊處理長度為 128 bits，所以 Li 等演算法設計 ID 及 CI 長度皆為 32 bits，外加一個固定亂數 N 長度為 64 bits，來填滿b_{ID}^{N}第一處理長度，以達到不可追蹤性到第一區塊密文，這樣的設計比較沒有彈性，如果在 ID 及 CI 位元數增加或選取不同之加密法，如 triple-DES，可能存在攻擊者追蹤到第一區塊密文的風險。

3. 系統架構

3.1 系統概述

本論文主要以 Li 等中登入認證系統裡「不可追蹤性」的安全特性為基礎[11]，探討如何設計一個高效能且富彈性的登入認證系統。此系統主要以動態密碼作為認證因子[4],[5],[7],[9]，每次登入皆使用不同密碼做認證，減少完成互相認證所需回合數，不僅節省通訊過程的溝通及運算成本，更落實了「不可追蹤性」。

我們提出的登入認證系統必須定義下列名詞:

- P_N:線性反饋位移暫存器方程式。
- N_0:P_N方程式的種子。
- N_i:由P_N及N_0所產生序列中第 i 回合動態密碼值。

登入認證系統為滿足「不可追蹤性」，每一回合雙方通訊的認證資訊皆需具隨機性，攻擊者無法得知一位使用者與伺服器通訊次數，避免攻擊者透過智慧卡與伺服器通訊資訊來推測使用者身份。因此我們透過「線性反饋位移暫存器」做為動態密碼的產生器[1],[10],[12]，認證過程每回合使用P_N方程式產生新的動態密碼做為認證資訊，用來確認訊息是否來自合法的另一方，即可防止攻擊者連結認證資訊推測使用者身份的風險。

本論文之登入認證系統一共分為三個階段(註冊、第 i 次登入階段(i 從 1 開始)及動態密碼重置階段)，註冊階段使用者必須註冊P_N方程式起始的種子N_0設定密碼循環的起始狀態，讓P_N方程式產生起始動態密碼，伺服器會給定每位使用者一資料儲存索引 UI，並將使用者資訊 ID 及 UI 和下一回合動態密碼N_1加密產生$v_{ID}^{N_1}$傳給使用者，讓使用者登入過程確認智慧卡身份，第 i 次登入(i 從 1 開始)階段智慧卡傳遞

$v_{ID}^{N_{2i-1}}$及P_N方程式下一回合產生的動態密碼給伺服器，伺服器在傳輸P_N方程式的下一回合動態密碼給智慧卡，經過兩回合的通訊產生會議鑰匙。動態密碼重置階段使用者想更換P_N方程式的起始種子，選定新的種子N_0^*以透過互相認證產生的會議鑰匙加密，讓伺服器更新註冊資料表的使用者資訊 N，智慧卡更新記憶體中資訊v_{ID}^{N}。

3.2 線性反饋位移暫存器方程式

線性反饋位移暫存器(LFSR, Linear Feedback Shift Register)原是用以簡化隨機存取暫存器的定址[1],[10],[12]，延伸應用為虛擬亂數產生器(PRNG)，供密碼系統加密之用，線性反饋位移暫存器方程式以線性二進位 0 與 1 位元串流方式產生，證實能快速產生亂數，起始的方程式種子值輸入後，下一位的位元串流和前一位元相關，輸入位元則由最終位元和運算位元作XOR運算，只要得知目前的亂數，即可產生下一回合亂數，一段時間後即會重複起始種子值亂數，產生不可預測的亂數循環，輸入的種子為 n 位元位移暫存器，排除種子為 0，形成2^n-1種狀態的亂數循環。

本論文使用線性反饋位移暫存器來做動態密碼產生器，使用者只需選擇P_N方程式起始種子N_0，不可預測的特性讓認證雙方透過每回合產生的動態密碼做身份確認，只要確保P_N方程式種子N_0不被攻擊者竊取，即能滿足「不可追蹤性」，更簡化了認證過程降低通訊過程溝通和運算成本。為說明運作方式，圖 7 為一 16 位元線性反饋位移暫存器方程式範例。

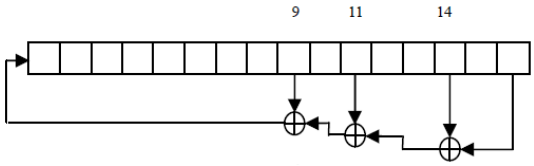


圖 7 16 位元線性反饋位移暫存器方程式

由圖 7 可知當我們一個 16 位元的線性反饋位移暫存器方程式，可產生$2^{16}-1$種狀態的亂數循環，選定位置第 9 及 11 和 14 的位元為運算位元，假設我們選定的起始種子為 517，二進位表示為 0000001000000101，即可產生下一回合產生的亂數以二進位表示為 0000000100000010，十進位為 258。

3.3 註冊階段

使用者必須先選定P_N方程式的起始種子N_0，將P_N和N_0傳遞給認證伺服器，伺服器收到訊息後給使用者一個資料儲存索引 UI，伺服器將(ID, UI, N_0)三個訊息儲存在註冊資料表，接著產生起始狀態下一回合動態密碼N_1，將使用者身份資訊用伺服器密鑰 s 加密產生$v_{ID}^{N_1}$，$v_{ID}^{N_1}=E_s(N_1||ID||UI)$，密文$v_{ID}^{N_1}$是以區塊密文密器加密，如果明文超過一區塊長度，對稱加密$E_s(\cdot)$則以密文區塊串接模式(CBC, Cipher Block Chaining)做處理，最後將 ID、UI 及$v_{ID}^{N_1}$傳送給使用者，使用者將$(ID, UI, v_{ID}^{N_1})$儲存在智慧卡中做登入階段的認證因子。圖 8 為本論文註冊階段流程圖。

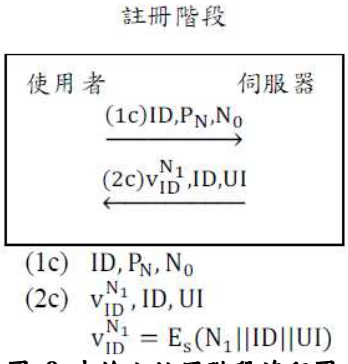


圖 8 本論文註冊階段流程圖

3.4 第 i 次登入階段(i 從 1 開始)

1 智慧卡與伺服器通訊，將註冊階段之$v_{ID}^{N_{2i-1}}$及雜湊運算$h(N_{2i}||v_{ID}^{N_{2i-1}})$傳送給認證伺服器，$N_{2i}$為$N_{2i-1}$下一回合動態密碼。

2 伺服期收到訊息時，利用密鑰 s 解密$v_{ID}^{N_{2i-1}}$取得用者身份資訊，計算$h(N_{2i}||v_{ID}^{N_{2i-1}})$確認訊息是否來自合法的使用者，並確保資料完整性。

3 通過認證後，伺服器使用P_N方程

式產生N_{2i}下一回合動態密碼N_{2i+1}，使用密鑰 s 將使用者身份資訊 ID 及N_{2i+1}做加密運算產生$v_{ID}^{N_{2i+1}}$，$v_{ID}^{N_{2i+1}} = E_s(N_{2i+1}||ID||UI)$，將$v_{ID}^{N_{2i+1}} \oplus h(N_{2i+1})$與雜湊運算$h(h(N_{2i+1})||v_{ID}^{N_{2i+1}})$傳送給智慧卡，伺服器計算會議鑰匙$h(N_{2i} \oplus N_{2i+1} \oplus v_{ID}^{N_{2i+1}})$。

4 智慧卡收到訊息，N_{2i+1}智慧卡產生N_{2i+1}計算$h(h(N_{2i+1})||v_{ID}^{N_{2i+1}})$確認訊息是否來自合法的伺服器。

5 通過認證後，智慧卡更新記憶體用$v_{ID}^{N_{2i+1}}$取代原先的$v_{ID}^{N_{2i-1}}$，下次第 i 次登入階段(i 從 1 開始)使用則使用$v_{ID}^{N_{2i+1}}$做認證，通過互相認證即產生通訊的會議鑰匙$h(N_{2i} \oplus N_{2i+1} \oplus v_{ID}^{N_{2i+1}})$。圖 9 為本論文第 i 次登入階段(i 從 1 開始)流程圖。

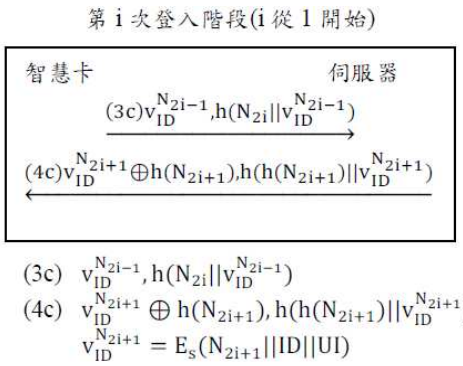


圖 9 本論文第 i 次登入階段 (i 從 1 開始)流程圖

3.5 動態密碼重置階段

動態密碼重置階段為使用者更換P_N方程式的起始種子N_0，產生不同循環的動態密碼，智慧卡選定新的N_0為N_0^*，使用會議鑰匙 k 將N_0^*及使用者身份資訊加密$E_k(N_0^*||ID||UI)$傳送給認證伺服器，伺服器收到訊息後使用 k 解密取得N_0^*，更新註冊資料表，以N_0^*取代N_i，接著使用密鑰 s 將N_0^*下一回合動態密碼N_1^*和使用者身份資訊加密產生$v_{ID}^{N_1^*}$，$v_{ID}^{N_1^*} = E_s(N_1^*||ID||UI)$，伺服器再使用 k 將$v_{ID}^{N_1^*}$及$N_1^*$加密傳送$E_k(v_{ID}^{N_1^*}||ID||UI||N_1^*)$給智慧卡，智慧卡用 k 解密後確認訊息$N_1^*$是否正確，通過認證以$v_{ID}^{N_1^*}$更新記憶體取代$v_{ID}^{N}$，下一次便使用$v_{ID}^{N_1^*}$做登入認證。圖 10 為本論文動態密碼重置階段流程圖。

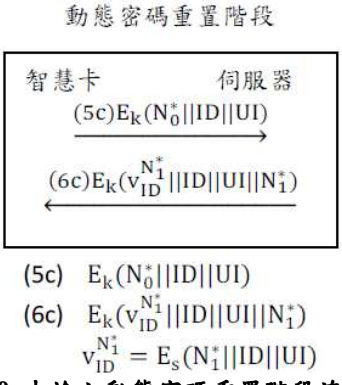


圖 10 本論文動態密碼重置階段流程圖

3.6 本論文安全性分析

本論文在滿足 Li 等演算法「匿名性」及「不可追蹤性」等安全特性下，針對溝通與運算成本以及認證資訊的限制加以改進，發展出一套具高效能與安全性且富彈性的登入認證系統。根據 1.1 節所述安全的登入認證系統應具有之特性，以下針對這些安全特性加以分析探討。

3.6.1 互相認證(Mutual Authentication)

會議鑰匙的產生必須通過第 i 次登入階段(i 從 1 開始)的互相認證，智慧卡和伺服器雙方必須確認訊息是來自合法的一方，本論文提出以動態密碼作為認證因子，登入階段當伺服器收到智慧卡傳送$h(N_{2i}||v_{ID}^{N_{2i-1}})$後，計算$h(N_{2i}||v_{ID}^{N_{2i-1}})$是否相同，若相同表示訊息是由合法的智慧卡傳送，因伺服器可透過P_N方程式產生下一回合動態密碼N_{2i}來做認證，伺服器傳送$h(h(N_{2i+1})||v_{ID}^{N_{2i+1}})$給智慧卡，智慧卡收到訊息後，智慧卡產生下一回合動態密碼N_{2i+1}計算$h(h(N_{2i+1})||v_{ID}^{N_{2i+1}})$是否相同，若

相同則通過互相認證，雙方即同意通訊之會議鑰匙。

3.6.2 會議鑰匙協議(Session Key Agreement)

會議鑰匙的產生必須保證在通過互相認證下以秘密方式共享，現今大部分的會議鑰匙協議必須滿足下列特性，會議鑰匙在通訊過程必須是唯一，每回合進行認證產生之會議鑰匙必須獨立，即上一次進行通訊之會議鑰匙並不會影響此次進行通訊的會議鑰匙。

本論文提出以動態密碼來做為組成會議鑰匙的因子，使用 XOR 做運算產生會議鑰匙$h(N_{2i} \oplus N_{2i+1} \oplus v_{ID}^{N_{2i+1}})$，$N_{2i}$和$N_{2i+1}$為智慧卡和伺服器在此次第 i 次登入階段(i 從 1 開始)分別由兩端產生的動態密碼，雙方產生的動態密碼皆為唯一，因此每回合產生的動態密碼具唯一性。動態密碼皆透過雜湊運算做處理，攻擊者無法取得真正的動態密碼，假設攻擊者不當取得上一次通訊之會議鑰匙，動態密碼為P_N方程式產生之虛擬亂數循環，因此即使上次通訊之會議鑰匙被竊取，下一次進行通訊之會議鑰匙也會因動態密碼產生變動，攻擊者更無法偽造會議鑰匙。

3.6.3 匿名性(Anonymity)

「匿名性」必須符合通訊過程只有伺服器知道智慧卡身份，第三方無法得知，也就是說使用者身份不能以明文方式傳送，本論文使用伺服器密鑰 s 加密使用者身份 ID 產生$v_{ID}^{N_{2i-1}}$，假如通訊過程攻擊者竊取$v_{ID}^{N_{2i-1}}$，想解密$v_{ID}^{N_{2i-1}}$取得使用者身份資訊 ID，沒有伺服器密鑰 s 即無法完成，故滿足「匿名性」要求。

3.6.4 不可追蹤性(Untraceability)

「不可追蹤性」為加強匿名性的安全特性，攻擊者無法藉由通訊過程的連結資訊來猜測使用者資訊，即攻擊者無法確認伺服器與哪位使用者通訊一次以上。我們選擇使用區塊加密來進行對稱加密操作，假如明文超出一個區塊長度，我們使用 CBC 模式來做處理，每個區塊明文加密後，皆與前一區塊密文做 XOR 運算結合，每一區塊密文與前一區塊相依，可以確認相同區塊長度的明文，會因不一樣的加密資訊產生不一樣的密文。

假設進行兩次第 i 次登入階段(i 從 1 開始)的通訊過程，分別的認證資訊如下，

authentication session1=

$$(v_{ID}^{N_{2i-1}}, h(N_{2i}||v_{ID}^{N_{2i-1}}), v_{ID}^{N_{2i+1}} \oplus h(N_{2i+1}), h(h(N_{2i+1})||v_{ID}^{N_{2i+1}}))$$

authentication session2=

$$(v_{ID}^{\bar{N}_{2i-1}}, h(\bar{N}_{2i}||v_{ID}^{\bar{N}_{2i-1}}), v_{ID}^{\bar{N}_{2i+1}} \oplus h(\bar{N}_{2i+1}), h(h(\bar{N}_{2i+1})||v_{ID}^{\bar{N}_{2i+1}}))$$

因動態密碼N_i產生的亂數循環，使得區塊密文$E_s(\cdot)$皆不相同，$v_{ID}^{N_{2i-1}}$與$v_{ID}^{\bar{N}_{2i-1}}$互相獨立，$v_{ID}^{N_{2i+1}}$與$v_{ID}^{\bar{N}_{2i+1}}$同樣也是獨立，使的每次與伺服器認證資訊皆不同，$h(\cdot)$為密碼學安全的雜湊函數，$h(N_{2i}||v_{ID}^{N_{2i-1}})$與$h(\bar{N}_{2i}||v_{ID}^{\bar{N}_{2i-1}})$，$h(N_{2i+1})$與$h(\bar{N}_{2i+1})$，$h(h(N_{2i+1})||v_{ID}^{N_{2i+1}})$和$h(h(\bar{N}_{2i+1})||v_{ID}^{\bar{N}_{2i+1}})$同樣也是獨立。

假設攻擊者竊取到兩次通訊密文$v_{ID}^{N_{2i-1}}$及$v_{ID}^{\bar{N}_{2i-1}}$，兩則訊息透露出一些關聯，攻擊者可能連結兩則訊息來判斷使用者身份，$v_{ID}^{N_{2i-1}}$由伺服器密鑰加密產生，攻擊者可能透過區塊密文做連結，我們將區塊密文加密動態密碼 N，每一區塊密文通訊時皆具有隨機性，攻擊者無法得知P_N方程式起始種子，即無法推測出動態密碼N_i的亂數循環，通訊過程動態密碼經由雜湊運算處理，攻擊者無法得知真正的動態密碼。總結因 authentication session1 和 authentication session2 互相獨立，攻擊者無法確認兩次登入認證過程有相同的訊息，動態密碼重置階段(5c)及(6c)也是相同，故滿足「不可追蹤性」。

3.6.5 驗證阻斷服務攻擊 (Denial of Service)

在動態密碼重置階段當智慧卡傳送訊

息(5c)給認證伺服器，使用者想更換動態密碼產生方程式P_N的起始種子，選擇新的起始種子N_0^*產生不同的動態密碼循環，伺服器傳送(6c)給智慧卡更新記憶體的認證資訊$v_{ID}^{N_1^*}$，如果攻擊者更改訊息(6c)為$E_{k'}(m)$，$|k'| = |k|$且$|m| = \left|v_{ID}^{N_1^*}\right|$，$k'$和 m 為攻擊者冒充伺服器傳送(6′c)給智慧卡，智慧卡解密訊息後更新記憶體以 m 取代v_{ID}^{N}，造成下次登入系統時認證失敗。本論文可透過(6c)訊息中的動態密碼N_1^*來判斷訊息是否為伺服器傳送的訊息，通過認證才更新智慧卡記憶體，保持一置性，可簡化「阻斷服務攻擊」的驗證，減輕智慧卡的負載。

3.6.6 防止離線密碼猜測攻擊(Off-Line Passowrd-Guessing Attack)

離線密碼猜測攻擊分為兩種，第一種為攻擊者取得密碼產生規則，並嘗試從規則中取得密碼，第二種為智慧卡遺失，攻擊者可由智慧卡中取出認證資訊來進行登入認證，因此訊息需擁有足夠資訊來認證密碼猜測，才能防止離線密碼猜測攻擊。在第 i 次登入階段(i 從 1 開始)，智慧卡傳送訊息(3c)給認證伺服器，認證資訊為$(v_{ID}^{N_{2i-1}}, h(N_{2i}||v_{ID}^{N_{2i-1}}))$，當攻擊者攔截訊息(3c)，攻擊者無法從$h(N_{2i}||v_{ID}^{N_{2i-1}})$中取得動態密碼$N_{2i}$，本論文將動態密碼透過伺服器密鑰加密，若攻擊者攔截$v_{ID}^{N_{2i-1}}$，只有伺服器能解密取得N_{2i-1}，因此攻擊者無法推測下一回合動態密碼N_{2i}來做雜湊運算$h(N_{2i}||v_{ID}^{N_{2i-1}})$，攻擊者無法登入做認證，即能防止「離線密碼猜測攻擊」。

3.6.7 防止重送攻擊(Replay Attack)

重送攻擊為攻擊者攔截智慧卡登入階段傳送訊息，並將訊息重新傳送給伺服器即可通過認證而入侵系統，伺服器則誤認攻擊者為合法使用者。本論文使用動態密碼來防止重送攻擊，智慧卡每次第 i 次登入階段(i 從 1 開始)皆會計算下一回合的動態密碼N_{2i}，使每次的登入認證訊息$h(N_{2i}||v_{ID}^{N_{2i-1}})$皆不同，第二及第三個會議鑰匙組成因子$N_{2i+1}$與$v_{ID}^{N_{2i+1}}$由伺服器產生，故能有效防止「重送攻擊」。

3.6.8 防止內部攻擊(Inside Attack)

內部攻擊為當使用者密碼在註冊階段被伺服器所擁有，在登入階段使用者必須忽略存在伺服器的密碼，即不能以註冊密碼當認證資訊以防止內部攻擊。本論文用動態密碼做為認證資訊，註冊階段伺服器僅儲存起始種子，在第 i 次登入階段(i 從 1 開始)以下一回合動態密碼N_{2i}做雜湊運算$h(N_{2i}||v_{ID}^{N_{2i-1}})$當認證資訊，與註冊階段起始動態密碼不同，認證過程也會因動態密碼循環使每回合產生不同之認證資訊，故滿足不可追蹤性。

3.6.9 成本因素

本論文以動態密碼作為認證因子，每次登入皆使用不同動態密碼做認證，在第 i 次登入階段(i 從 1 開始)完成互相認證只需兩回合，不僅簡化了認證程序，更大幅節省通訊過程的溝通及運算成本。

最後，根據以上之安全性分析，將所探討的 JCL 及 Li 等與本論文所提出的演算法，彙整成以下的比較表。表 1 為本論文及相關演算法登入認證系統安全性比較表。

表 1 本論文及相關演算法
登入認證系統安全性比較表

	本論文	JCL	Li 等
互相認證	✓	✓	✓
鑰匙協議	✓	✓	✓
匿名性	✓	✓	✓
不可追蹤性	✓	×	✓*
驗證阻斷服務攻擊	✓	×	✓
防止內部攻擊	✓	✓	✓
防止離線密碼猜測	✓	✓	✓
防止重送攻擊	✓	✓	✓
成本因素	低	中	高

* Li 等方法需調整對稱加密器及 ID 和 UI 之值

4. 研究結果

此章節我們將針對 JCL 及 Li 等與本論文所提出之演算法作效能評估，以溝通與運算成本及各階段實際執行時間為準則。我們假設 ID 及 UI 長度皆為 32 bits，每回合線性反饋位移暫存器方程式P_N的種子 N 長度為 32 bits，安全的雜湊演算法h(・)選擇以 SHA-1 做處理，訊息摘要長度為 160 bits，對稱式加密演算法E_s(・)選擇以 AES (Advanced Encryption Standard)處理，區塊密文的長度為 128 bits，本論文選擇以動態密碼作為認證因子，相較於 JCL 與 Li 等演算法以橢圓曲線做認證因子的運算，更能節省溝通及運算成本，實際執行實際也較少，降低了智慧卡的處理量。

4.1 溝通及儲存成本

溝通成本我們以下列要素做比較，密碼長度(PL)，智慧卡的儲存成本(CC)，每位使用者在認證伺服器的儲存成本(SC)，登入階段的智慧卡與伺服器傳輸訊息的總溝通成本(LC)，密碼更換階段智慧卡與伺服器傳輸訊息的總溝通成本(PC)。

本論文以動態密碼做認證，因線性反饋位移暫存器方程式P_N種子 N 為 32 bits，每回合產生之動態密碼取 64 bits，註冊階段智慧卡儲存參數$(ID, UI, v_{ID}^{N_1})$，儲存成本為 32+32+128=192 bits，$v_{ID}^{N_1}$以一個區塊加密，每位使用者在伺服器的儲存參數為(ID, UI, N_0)，儲存成本為 32+32+32=96 bits，第 i 次登入階段(i 從 1 開始)的雙方傳輸訊息為$(v_{ID}^{N_{2i-1}},\ h(N_{2i}||v_{ID}^{N_{2i-1}}),\ v_{ID}^{N_{2i+1}} \oplus h(N_{2i+1}), h(h(N_{2i+1})||v_{ID}^{N_{2i+1}})$，溝通成本為 128+160+160+160=608 bits，動態密碼重置階段雙方傳輸訊息$(E_k(N_0^*||ID||UI), E_k(v_{ID}^{N_1^*}||ID||UI||N_1^*))$，溝通成本為 128+256=384 bits，$E_k(N_0^*||ID||UI)$以一個區塊加密，$E_k(v_{ID}^{N_1^*}||ID||UI||N_1^*)$以兩個區塊加密。

JCL 演算法區塊密文E_s(・)及雜湊運算h(・)訊息摘要長度皆為 128bits，系統密碼經雜湊運算h(PW||b)，長度為 128 bits，智慧卡儲存資訊$(ID, CI, b_{ID}, V_{ID}, b)$，儲存成本 32+32+384+128+64=640 bits，伺服器儲存資訊(ID, CI)，儲存成本為 32+32=64 bits，登入階段雙方傳輸訊息為$(b_{ID}, E_{V_{ID}}(e), u, M_S, M_U)$，溝通成本為 384+384+64+128+128=1088 bits，b_{ID}及$E_{V_{ID}}(e)$皆以三個區塊加密，密碼更換階段雙方傳輸訊息為$(E_k(ID||h(PW^*||b^*)),\ E_k(b_{ID}{}^*))$，溝通成本為 384+512=896 bits，$E_k(ID||h(PW^*||b^*))$以三個區塊加密，$E_k(b_{ID}{}^*)$以四個區塊加密。

Li 等區塊密文E_s(・)及雜湊運算h(・)輸出長度等同 JCL，系統密碼h(PW||b)長度同樣為 128 bits，智慧卡儲存資訊$(b_{ID}^{N_0}, V_{ID}, ID, CI,\ b)$，儲存成本為 384+128+32+32+64 =640 bits，$b_{ID}^{N_0}$以三個區塊加密，伺服器儲存資訊(ID, CI, N_0)，儲存成本為 32+32+64=128 bits，登入階段雙方傳輸訊息為$(b_{ID}^{N_0}, E_{V_{ID}}(N_1||e), Nb1, u \oplus h_{64}(b_{ID}^{N_1}), M_S, M_U)$，溝通成本為 384+ 512+384+128+64+128+128=1728 bits，$b_{ID}^{N_0}$以三個區塊加密，$E_{V_{ID}}(N_1||e)$以四個區塊加密，密碼更換階段雙方傳輸訊息為$(E_k((ID||CI||N^*)||\ h(PW^*||b^*)),\ E_k(b_{ID}^{N^*}||ID||CI||N^*))$，溝通成本為 384+640=1024 bits，$E_k((ID||CI||N^*)||\ h(PW^*||b^*))$使用三個區塊加密，$E_k(b_{ID}^{N^*}||ID||CI||N^*)$使用五個區塊加密。最後，根據以上所探討的 JCL 及 Li 等與本論文所提出的演算法之儲存成本及溝通成本，彙整成以下的比較表。表 2 為本論文與相關演算法登入認證系統溝通成本及儲存成本比較表。

表 2 本論文及相關演算法
登入認證系統溝通成本及儲存成本比較表

	PL	CC	SC	LC	PC
本論文	64 bits	192 bits	96 bits	608 bits	384 bits
JCL	128 bits	640 bits	64 bits	1088 bits	896 bits
Li 等	128 bits	640 bits	128 bits	1728 bits	1024 bits

PL:密碼長度(bits);
CC:智慧卡儲存成本(bits);
SC:伺服器儲存成本(bits);
LC:登入階段總溝通成本(bits);

PC:密碼更換階段總溝通成本(bits);

4.2 計算成本

計算成本我們以下列要素做比較，登入階段通訊回合數(LN)，註冊階段使用者計算成本(RU)，註冊階段伺服器計算成本(RS)，登入階段使用者計算成本(LU)，登入階段伺服器計算成本(LS)，密碼更換階段使用者計算成本(PU)，密碼更換階段伺服器計算成本(PS)。並使用下列單位做衡量，雜湊運算(h)，每個區塊的對稱式加解密(s)，動態密碼產生序列(p)，橢圓曲線運算(m)。

本論文因使用動態密碼做認證因子，因此在登入階段雙方透過動態密碼的循環特性來判斷是否為真實的另一方，只需要兩回合的通訊即能完成互相認證，註冊階段使用者透過線性反饋位移暫存器P_N產生起始密碼，故計算成本為 1p，伺服器產生下一回合動態密碼N_1，進行一個區塊的對稱式加密$v_{ID}^{N_1}$，計算成本為 1p+1s，第 i 次登入階段(i 從 1 開始)智慧卡要產生上次第 i 次登入階段(i 從 1 開始)結束回合的下一回合動態密碼N_{2i}，將N_{2i}做雜湊後$h(N_{2i}||v_{ID}^{N_{2i-1}})$傳輸給伺服器，計算$h(N_{2i+1})$取得$v_{ID}^{N_{2i+1}}$，確認伺服器傳輸訊息$h(h(N_{2i+1})||v_{ID}^{N_{2i+1}})$的動態密碼循環是否正確，故計算成本為 2p+3h，伺服器端必須解密$v_{ID}^{N_{2i-1}}$判斷使用者身份，並確認智慧卡傳輸訊息的動態密碼循環$h(N_{2i}||v_{ID}^{N_{2i-1}})$是否正確，產生下一回合動態密碼$N_{2i+1}$，將$N_{2i+1}$重新進行一個區塊的對稱式加密$v_{ID}^{N_{2i+1}}$，對$N_{2i+1}$進行雜湊運算 $h(N_{2i+1})$，最後把 $h(N_{2i+1})$及$v_{ID}^{N_{2i+1}}$做雜湊運算$h(h(N_{2i+1})||v_{ID}^{N_{2i+1}})$，故計算成本為 2p+3h+2s，動態密碼重置階段使用者傳輸新的起始種子N_0^*，使用會議鑰匙 k 加密$E_k(N_0^*||ID||UI)$，解密伺服器傳輸訊息$E_k(v_{ID}^{N_1^*}||ID||UI||N_1^*)$並驗證$N_1^*$是否正確，故計算成本為 2p+3s，伺服器收到訊息$E_k(N_0^*||ID||UI)$後解密，將起始動種子下一回合動態密碼N_1^*加密產生$v_{ID}^{N_1^*}$，最後使用會議鑰匙進行兩個區塊加密$E_k(v_{ID}^{N_1^*}||ID||UI||N_1^*)$，故計算成本為 1p+4s。

JCL 演算法使用橢圓曲線做認證因子，計算單位為 m，登入階段需三回合的通訊來完成互相認證，註冊階段使用者的計算成本為 1h，伺服器的計算成本為 2h+3s，登入階段使用者計算成本為 3h+3s，伺服器計算成本為 1m+4h+6s，密碼更換階段使用者計算成本為 1h+5s，伺服器計算成本為 1h+5s。

Li 等登入階段同樣需三回合的通訊來完成互相認證，註冊階段使用者的計算成本為 1h，伺服器的計算成本為 2h+3s，登入階段使用者計算成本為 8h+4s，伺服器計算成本為 1m+10h+10s，密碼更換階段使用者計算成本為 1h+6s，伺服器計算成本為 1h+9s。根據以上所探討的 JCL 與 Li 等及本論文所提出的演算法之各計算成本，彙整成以下的比較表。表 3 為本論文及相關演算法登入認證系統計算成本比較表。

表 3 本論文及相關演算法
登入認證系統計算成本比較表

	LN	RU	RS	LU	LS	PU	PS
本論文	2	1p	1p +1s	2p +3h	2p+3h +2s	2p +3s	1p +4s
JCL	3	1h	2h +s	3h +3s	1m+4h +6s	1h +5s	1h +5s
Li 等	3	1h	2h +3s	8h +4s	1m+10 h+10s	1h +6s	1h +9s

LN:登入階段通訊回合數;
RU:註冊階段使用者計算成本;
RS:註冊階段伺服器計算成本;
LU:登入階段使用者計算成本;
LS:登入階段伺服器計算成本;
PU:密碼更換階段使用者計算成本;
PS:密碼更換階段伺服器計算成本;
h:雜湊運算;
s:每個區塊的對稱式加解密;
p:動態密碼產生序列;
m:橢圓曲線運算;

4.3 執行時間

以下針對各演算法之各階段實際執行時間進行實驗，本研究的實驗環境為 Intel Core2 Duo processor P8600 2.4G 雙核心 CPU 處理速度，供後續研究者一個參考依據。

我們選擇以 JAVA 語言來做各階段執

行時間的實驗,因 JAVA 平台提供強大安全防護功能的標準化 API,如加解密演算法、訊息摘要、橢圓曲線等相關密碼學套件，方便我們快速針對各階段實際執行時間的實驗。

我們必須對執行單位做參數的設定:

- s:我們選擇以 AES 演算法來做對稱式加解密,設定 CBC 處理模式,每個區塊為 128 bits，鑰匙長度同樣取 128 bits，並使用 PKCS5 填充機制來提供必要的填充字元。
- h:我們選擇以 SHA1 演算法做訊息摘要的處理，並透過位元串流方式來提供雜湊運算資料。
- m:以 JAVA 語言提供的 Security 介面中的 EllipticCurve 類別，產生橢圓曲線中心點 ECpoint 及 ECprivate Key，再透過 SecureRandom 類別產生一 Pseudorandom，將三個參數相乘產生認證因子。
- p:LFSR 的虛擬亂數產生器，我們輸入 32 位元的種子值，以位移暫存器的概念,逐一位元進行處理，輸出為 64 位元亂數。

根據以上執行單位針對各單位執行時間彙整成下表。表 4 為本論文及相關演算法登入認證系統單位執行時間比較表。

表 4 本論文及相關演算法登入認證系統單位執行時間比較表

計算單位	演算法	執行時間
h(SHA-1)	本論文	0.02ms
	JCL	0.02ms
	Li 等	0.02ms
s(AES)	本論文	0.028ms
	JCL	0.028ms
	Li 等	0.028ms
m (Elliptic Curve)	JCL	0.137ms
	Li 等	0.137ms
p(LFSR)	本論文	0.017ms

h(SHA-1):雜湊運算;
s(AES):每個區塊的對稱式加解密;
p(LFSR):動態密碼產生序列;
m(Elliptic Curve):橢圓曲線運算;

最後我們以單位執行時間，來做各階段實際執行時間的實驗，彙整成下列比較表。表 5 為本論文及相關演算法登入認證系統各階段執行時間比較表。

表 5 本論文及相關演算法登入認證系統各階段執行時間比較表

	RT	LT	PT
本論文	0.062ms	0.244ms	0.247ms
JCL	0.144ms	0.529ms	0.32ms
Li 等	0.144ms	0.889ms	0.46ms

RT:註冊階段執行時間;
LT:登入階段執行時間;
PT:密碼更換階段執行時間;

總結以上溝通成本與運算成本及各階段實際執行時間的評估，本論文提出之演算法在同樣滿足安全特性下，相較於 JCL 及 Li 等更低，降低嵌入式裝置的晶片處理器運算負載量，智慧卡運作效能更佳，延伸了系統在實務上的應用。

5. 結論

本論文使用以線性反饋位移暫存器(LFSR)產生虛擬亂數循環的動態密碼，使用者只需選定方程式起始種子，即能以動態密碼作為登入階段的認證因子，因動態密碼的隨機特性，使每回合的認證因子皆不同，讓訊息接收方透過動態密碼的循環特性，認證是否為合法的訊息傳遞方，在做互相認證及會議鑰匙協議上不僅同時滿足 Li 等演算法中「匿名性」延伸的「不可追蹤性」，認證過程相較於 Li 等演算法更具彈性，攻擊者無法藉由通訊過程的連結資訊來猜測使用者資訊，即攻擊者無法確認伺服器與哪位使用者通訊一次以上，在網路環境中加強了使用者身份的保護。

總結針對溝通成本與運算成本及各階段實際執行時間進行比較，本論文提出的智慧卡下的密碼認證系統相較於 Li 等演算法效率較高。以效能而言，認證過程中低溝通與運算成本及實際執行時間，以安全的雜湊函數及對稱加密，使智慧卡環境下低運算的負載量,與真實的應用更加符合，以效率而言,有效防止離線密碼猜測攻擊、重送攻擊、內部攻擊、及簡化阻斷服務攻擊的驗證，達到效率及效能兼具的智慧卡登入認證系統。

參考文獻

[1] Alfke,P.,“Efficient Shift Registers,LFSR Counters,and Long Pseudo-Random Sequence Generators,” Xilinx Application Note,Jul.1996.

[2] Das, M.L., “Two-Factor User Authentication in Wireless Sensor Networks,” IEEE, Trans. On. Wire. Commun.,vol.15,no.3,pp.1086- 1090, Mar.2009.

[3] Fan,C., Y.Chan, and Zhang, Z., “ Robust remote authentication scheme with smart cards,” Comput.Secur., vol.24,no.8,pp.618-628,Nov.2005.

[4] He, D. and Chan,S., “A Secure and Light weight User Authentication Scheme with Anonymity for the Global Mobility Network,”IEEE,13th International Conference on Network-Based Information Systems,2010.

[5] Huang,H., Liu, S., and Chen, H., “Designing a new mutual authentication scheme based on nonce and smart cards,” IEEE, Computer Sociel, International Symposium on Parallel and Distributed Processing with Applications, Sep. 2010.

[6] Juang, W., Chen, S., and Liaw,H., “Robust and efficient password-authenticated key agreement using smart cards,”IEEE, Trans.Ind.Electron., vol.15,no.6, pp. 2551-2556,Jun.2008.

[7] Li, B., Hu,S., and Liu, Y. , “A Practical One-Time Password Authentication Implement on Internet,” IEEE International Conference on Wireless, Mobile and Multimedia Networks, 2006.

[8] Lauter,K., “The advantages of elliptic curve cryptography for wireless security,” Wireless Commun., vol.11, no.1, pp.62-67, Feb.2004.

[9] Luo,S., Hu,J., and Chen,Z., “An Identity- Based One-Time Password Scheme with Anonymous Authentication,” IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing,2009.

[10] Liang, W. and Jing, L. “A Cryptographic Algorithm Based on Linear Feedback Shift Register,”IEEE International Conference on Computer Application and System Modeling, pp.526-529, 2010.

[11] Li, X., Qiu, W., Zheng,D., Chen, K., and Li, J., “Anonymity Enhancement on Robust and Efficient Password-Authenticated Key Agreement Using Smart Cards,” IEEE, Trans. Ind. Electron., Vol.57, no.2, pp.793-800, Feb. 2010.

[12] Pasalic, E.,“On Guess and Determine Cryptanalysis of LFSR-Based Stream Ciphers” IEEE Trans. on. Infor. Theo., vol.55, no.7, Jul. 2009.

[13] Pu, Q., “An Improved Two-factor Authentication Protocol,” Second International Conference on MultiMedia and Information Technology.2010.

[14] WIKIPEDIA., Pseudorandom number generator. http://en.wikipedia.org/wiki/Pseudorandom_ number_generator, accessed 2011/2/1.