密碼學演算法於 IC 卡上的具體實現

楊中皇
國立高雄第一科技大學資訊管理系

摘要

網際網路的的高速發展帶給我們極大的便利，但也隨之不斷產生網路安全的問題。一般網路資訊系統以帳號與通行碼方式來確認使用者身份，但易記的通行碼容易被揣測盜用，而設定較長的通行碼則使用者本身亦難以記住。IC 卡具有攜帶方便且難以仿造的優點，而使用內含加解密與數位簽章機制的 IC 卡將成為電子化社會的趨勢。

本文以內含 H8/300 八位元中央處理器的日立公司智慧卡晶片為例，說明如何於 IC 卡內部實現 AES、RSA、ESIGN 等加解密及數位簽章的密碼學演算法。由於 IC 卡內部的 RAM 記憶體少，且中央處理器速度有限，這使得密碼學演算法於 IC 卡上高速實現的成為關鍵性技術，也是建立我國通資訊基礎建設安全機制中重要的一環。

關鍵詞：*IC 卡，智慧卡，密碼學，數位簽章，電子簽章，數位信封，AES， RSA，ESIGN。*

壹、 前言

傳統上，密碼學(cryptography) [1] 之技術主要用於軍事與外交方面，研究成果也常因所謂的國家安全因素，而無法公開。然而隨著網際網路(Internet)與全球資訊網(WWW)的極速成長，網路安全問題也一再浮現出來。早期的網路安全措施多是倚賴封閉式的網路與設計細節的保密，並未在系統設計時便確實考量其安全性。然而隨著上網人口的增加，新聞群組(newsgroup)與電子佈告欄系統(bulletin board system)的風靡，使得全球性駭客族可以立即交換資訊、分享經驗。也因此近年來電腦網路被入侵、資料被竄改的情事在國內外不斷地發生。

一般網路系統以帳號與通行碼(password)方式來確認使用者身份，但通行碼有其缺點，意即使用者若為了增加其安全性而設定長度較長之通行碼，以防止他人在旁窺伺盜用，相對的使用者本身亦難以記住密碼，甚或有人將其寫下以便使用，如此便會失去保密意義；相反的，若使用者為了便於熟記，而使用如生日、電話號碼・・・等易記之通行碼，則相對的亦容易被有心人士利用各種方式揣測。有心人士甚至只要利用簡單的網路封包掃瞄工具，就可在傳輸途中截取使用者帳號與通行碼，所以使用內含密碼學機制的低成本硬體裝置如 IC 卡來提供身份確認等網路安全功能有其必要性。

使用密碼學技術對網路提供最佳的安全防禦，目前微軟公司(Microsoft)與網景公司(Netscape)的網頁瀏覽器(browser)也都配備這種技術。密碼系統中如果加密金鑰與解密金鑰兩者中有一者可以公開，則稱之公開金鑰密碼系統 (public-key cryptosystems)，其中最著名的系統為 RSA [2]。如果加密金鑰與解密金鑰兩者皆須保密，則稱之為私密金鑰密碼系統(private-key cryptosystems) 。私密金鑰密碼系統又可區分成串流密碼系統(stream cipher)與區塊密碼系統(block cipher)兩種。 前者具備較理想的數學分析理論，對密碼系統的強度有測量的標準，而大多數軍事與外交用的通信保密器多採用此種串流密碼系統。但由於涉及國家安全，軍事與外交串流密碼系統的設計理論多被視為國家機密，其精髓並不見於公開文獻。目前學術探討與坊間所用多是區塊密碼系統，其中最著名的系統為 DES (Data Encryption Standard)[3]。早期的密碼系統都是私密金鑰密碼系統，而公開金鑰密碼系統則是到 1976 年才被發明。

雖然 56 位元金鑰的 DES 為目前全世界使用最廣泛的加解密演算法，但隨著電子科技的發展與電腦運算速度的提升，設計破解 DES 的特殊硬體或以多部電腦合作破解 DES 的構想與實驗近幾年來一再被提出， 1999 年已有報導 DES 可在一天內被破解[4]。這也使得以 DES 為密碼演算法機制的系統安全性堪虞，而 2000 年 10 月美國政府機構 NIST 正式宣布選用 Rijndael[5]演算法作為 AES (Advanced Encryption Standard)，且於 2001 年成為美國聯邦資訊處理加密標準[6]，逐步取代 DES。但是世界上多數高科技國家，如美國、日本、法國，仍然對具備加解密保密功能的軟硬體裝置實施出口管制，或僅允許安全性較差之產品輸出。而國內進口的兼具保密功能的產品的安全性也一再令人質疑，行政院於民國九十年一月十七日第二七一八次院會通過「建立我國通資訊基礎建設安全機制計畫」建立國家資通安全整體防護體系目標，所以我們確實有必要建立本土化的資訊安全技術。

我國政府積極推行 IC 卡已有十年以上的歷史，IC 卡[7-9]在業界的應用也日趨廣泛。例如：被金融界視為塑膠貨幣的金融 IC 卡、中華電信推出之 IC 通話卡、以及今年即將開始使用的健保 IC 卡・・・等，美國國防部也開始全面使用有公開密碼學機制的 CAC IC 卡 (Common Access Card)[10]。由以上實例可知 IC 卡的應用日趨重要，而且 IC 卡與目前常用的磁條卡比較起來具有難以偽造與記憶體容量較高的優點。然而從我們多年來使用國外廠商所提供 IC 卡開發應用系統的經驗中，常見的問題便是廠商技術支援不足及價格偏高。而且廠商通常僅能提供內含較不安全的密碼學演算法的 IC 卡，例如 56 位元 DES (Data Encryption Standard) [3]，而多半無法提供如 Triple-DES [11]、AES [5]、1024 或 2048 位元 RSA 加解密等，或於使用上有所限制。這使得攸關 IC 卡安全性的密碼學演算法自行研發實現日行重要。

本文主要目的便是在介紹如何具體實現 IC 卡內部密碼學演算法，以使政府大力推行 IC 卡時能掌握關鍵性的技術。近年來新一代密碼學演算法一一浮現，除了美國 NIST 的 AES 計畫，歐洲也正在進行 NESSIE [12]計畫，日本也有類似的 CRYPTREC [13]計畫，皆在評估新一代的密碼學演算法。本文除了介紹 AES 的實現外，我們也介紹 RSA 及 ESIGN [14]兩種適合做電子簽章的演算法具體實現於 IC 卡。

IC 卡讀卡機的介面目前有 RS-232、USB、PCMCIA、3.5 吋磁片等，隨著 Windows2000 的內建 PC/SC 讀卡機驅動程式庫，同時 IC 卡讀卡機的價格也日益便宜，這使得 IC 卡越來越適合網路安全應用。

貳、八位元 H8/300 的 IC 卡

我們以日立公司 H8/3113 智慧卡(內含中央處理器的 IC 卡)晶片介紹 AES、RSA、ESIGN 等密碼學演算法的實現，在此我們簡單描述這智慧卡晶片的特性。H8/3113(晶片 IC 內部結構如圖 1)提供八位元 H8/300 [15]中央處理器，32K 位元組(byte)唯讀記憶體(ROM)，16K 位元組電流可消除可程式唯讀記憶體(EEPROM)，2K 位元組一般用途隨機存取記憶體(RAM)，以及 1024 位元乘法運算加速器(coprocessor)。圖 2 是日立公司發展工具的硬體實體圖，軟體可在 Windows 環境下作業。

晶片的 ROM 記憶體區域是用來儲存作業系統程式和儲存密碼運算法程式或其他固定的應用程式；EEPROM 記憶體區域是用來儲存個人化的資料(如私密金鑰、公開金鑰憑證)或其他易更動的應用程式；RAM 記憶體區域則是供暫時性運算變數資料儲存用。H8/3113 中央處理器僅是八位元，RAM 僅有 2K 位元組（另有 0.5K 位元組供乘法運算加速器用），

程式大小亦有限制；這使得晶片程式的開發必須以組合語言為主，以便有效地利用智慧卡晶片內部有限的資源。

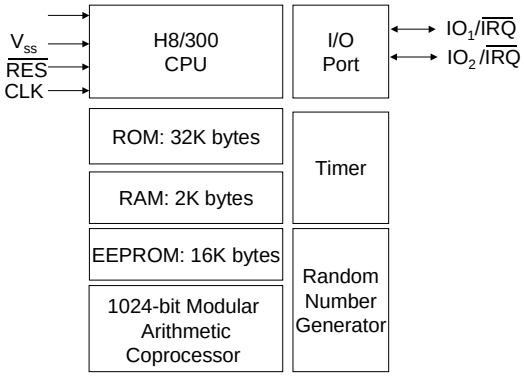

圖 1：H8/3113 智慧卡晶片內部結構圖

圖 2：H8/3113 智慧卡發展工具

1992 年美國政府正在發展 DSS (Digital Signature Standard) [16]數位簽章演算法時，曾以內含 H8/300 八位元中央處理器的智慧卡晶片(當時 RAM 僅有 256 位元組，ROM 有 10K 位元組，EEPROM 有 8K 位元組，也不具備加速器)評估 512 位元 DSS 與 RSA 的效率[17] (雖然目前對 RSA 密碼系統的金鑰長度要求 1024 位元，但十年前 512 位元已經夠安全)。假設 H8/300 晶片內部時鐘假設為 5MHz，表一說明 DSS 由於數位簽章較費時，所以宜採用 pre-computation 事先計算的方式，先將與亂數有關的計算處理，而 RSA 不適用此種方式。同年日本 NTT 也在同一智慧卡晶片評估其 ESIGN 演算法的效率[18]，結果列於表一，因為 576 位元 ESIGN 簽章可以快速算出，所以不需要進行 pre-computation。

表一：1992 年三種不同數位簽章演算法於 H8/300 智慧卡的效率

公開金鑰演算法	512-bit DSS	512-bit RSA	576-bit ESIGN
Pre-computation	28 秒	不適用	不需要
數位簽章產生時間	0.05 秒	25 秒	0.45 秒
數位簽章檢驗時間	56 秒	5 秒	0.27 秒

參、AES實現

NIST 評選比利時密碼學家所發明的 Rijndael [5]為 AES 時考慮了安全性、效能、效率、容易實做及有彈性。AES 的明文固定為 128 位元，金鑰則可為 128、192、或 256 位元。AES 每回合是由四個可逆的轉換所組成，稱作層(layers)。我們實現的 128 位元金

鑰 AES 的加密過程約如圖 3 所示。128 位元加解密金鑰先經過金鑰安排(key schedule)予以擴張計算出 11 組 128 位元的回合鑰(round key)。明文則經 AddRoundKey（須用到回合鑰)、ByteSubstitution、RotateRow、及 MixColumn 等層的轉換。

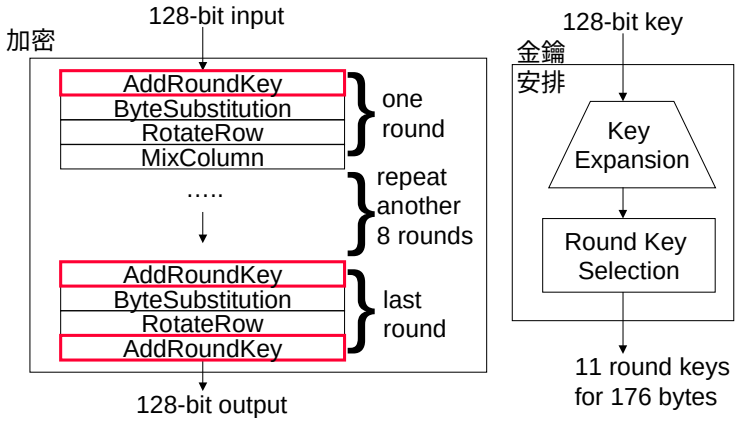

圖 3：128 位元金鑰之 AES 加密

我們只探討 128 位元(16 位元組)長度之 AES 加解密金鑰，因為這最有可能取代 DES 與 Triple-DES。AES 的金鑰安排在八位元平台則能有效地實現，因我們可以安排計算順序，減少記憶體的讀寫[19]。在評估演算法效率時，越大的程式及表格(佔用 ROM 區域)或越多的暫時變數(使用 RAM 區域)通常可加快執行速度；我們的組合語言程式與表格使用 ROM 記憶體 2.5K 位元組時，實現結果如表二所示，其中我們假設 H8/300 晶片內部時鐘假設為 5MHz。128 位元 AES 大約比我們過去於同一晶片實現的 DES 速度快約 4 倍，約比 Triple-DES 速度快約 10 倍。同時由於 128 位元金鑰之 AES 有 3.4×10^{38} 可能的金鑰選擇，遠較 56 位元 DES 有 7.2×10^{16} 可能的金鑰選擇為大，且截至目前為止並無有效的攻擊方式，故 AES 除了速度較快外也遠較 DES 安全。

表二：128 位元金鑰之 AES 於八位元 H8/300 智慧卡執行效率

金鑰安排時間	加密時間 (每個區塊)	加密處理速度
0.43 ms 1080 cycles	1.67 ms 4180 cycles	76 Kbits/s

肆、公開金鑰密碼學演算法 RSA、ESIGN 的實現

RSA公開金鑰密碼系統[2]是1977年美國麻省理工學院三位教授共同發明。在這系統中每個使用個體(entity)有一組成對的金鑰：公開金鑰與秘密金鑰。公開金鑰可以讓大眾知道它的內容，而秘密金鑰則要安全地保管(例如存放在IC卡內)。RSA密碼系統的安全原理是基於兩個大整數相乘得到乘積容易，而去分解兩個大整數的乘積則是很難(美國RSA公司甚至有總獎金超過美金六十萬元的有獎懸賞[20])。此系統的使用方式為：

1. 首先使用個體挑兩個大質數(例如 512 位元，可以用硬體或軟體產生)， P和 Q。
2. 再選個正整數 E滿足 GCD(E,(P-1)(Q-1))=1，意即 E與(P-1)(Q-1)的最大公因數為 1。
3. 利用歐基里得輾轉相除法(extended Euclidean algorithm)，計算出正整數 D,

滿足(符號 mod 代表模運算)

$$D \times E = 1 \bmod (P-1) \times (Q-1)$$

個體的公開金鑰為 *E* 與 *N* (= *P* x *Q*) ，可以到處公開，而私密金鑰為 *D* 與 *P* 及 *Q* 由自己保管。RSA 不僅能用在數位簽章，也適用在數位信封做金鑰加解密。假設 *H(M)*代表訊息 *M* 的單向雜湊函數(one-way hash function) [1] 值，那麼 RSA 數位簽章的產生與檢驗可簡單描述如下

(1) RSA 簽章產生：用私密金鑰 *D* 與公開金鑰 *N* 以下列公式計算簽章 *S*

$$S = H(M)^D \bmod N$$

(2) RSA 簽章檢驗:使用公開金鑰 *E* 與 *N* 去判斷 *H(M)*與 *S* 是否符合下列公式

$$H(M) \stackrel{?}{=} S^E \bmod N$$

實務上我們應使用中國剩餘定理(Chinese Remainder Theorem)來加速簽章的產生，此時除了金鑰 *N*、*P*、*Q*、*E*、*D* 之外，我們需有 *D* mod *P-1*、*D* mod *Q-1*、Q^{-1} mod *P*。這些數值將如表三儲存於 IC 卡的 EEPROM 區域供數位簽章的產生與檢驗用或提供加解密。

表三：RSA 相關金鑰儲存於 IC 卡 EEPROM 記憶體區域

N
E
D
P
Q
D mod *P - 1*
D mod *Q - 1*
Q^{-1} mod *P*

ESIGN (Efficient digital SIGNature 的縮寫)是日本 NTT 院士岡本龍明(Tatsuaki Okamoto)1990 年所創，公開金鑰為 *N* (= P^2Q) ，私密金鑰為 *P* 與 *Q*。ESIGN 與 DSS 相似，僅能做數位簽章之用，而無法如 RSA 做加解密用。ESIGN 的簽章屬於隨機數位簽章(randomized digital signature)[1]，亦即同一訊息有多個有效的數位簽章，而目前 ESIGN 已載於國際標準 ISO 14888-3[21]且獲選進入歐洲 NESSIE [12]計畫的第二階段。

(1) ESIGN 簽章產生：令 *k* 為私密金鑰 *P* 的位元長度，*R* 為介於 0 與 *P* x *Q* 之間的隨機亂數，*E* 為大於 4 的安全參數（例如 *E*=1024)，符號⌈ ⌉代表天花板函數(ceiling function) ，*H(M)*在此代表訊息 *M* 的單向雜湊函數值但長度為 *k-1* 位元，$(0\|H(M)\|0^{2k})$為長度為 *3k* 位元(先擺個零，串接 *k-1* 位元的 *h(m)*，再串接 *2k* 個零)。以下列公式計算出長度為 *3k* 位元的簽章 *S*。

$$W_0 = \left\lceil \frac{(0\|H(M)\|0^{2k}) - R^E \bmod N}{PQ} \right\rceil$$

$$T = W_0 \times \left(E \times R^{E-1}\right)^{-1} \bmod P$$

$$S = R + T \times PQ \bmod N$$

(2) ESIGN 簽章檢驗:使用公開金鑰 N 與 E 算出 $S^E \bmod N$，再判斷 $k+1$ 個最有效的位元(most significant bits) 是否與 $(0||H(M)||0)$ 相同。

$$(0\|H(M)\|0) \stackrel{?}{=} \left[S^E \bmod N\right]^{k+1}$$

中國剩餘定理無法適用於 ESIGN，而表四描述 ESIGN 金鑰 N、R、PQ 儲存於 IC 卡 EEPROM 的區域，供簽章的產生與檢驗。

表四：ESIGN 相關金鑰儲存於 IC 卡 EEPROM 記憶體區域

N
P
P x Q

RSA 與 ESIGN 都要用到模算術(modular arithmetic) [22]，包括模指數、模乘法、模反元素等運算。如表一所示，512 位元 RSA 於八位元 H8/300 智慧卡的簽章產生需耗時 25 秒，1024 位元 RSA 的簽章產生估計要 200 秒(其實在 1992 年多大數 IC 卡的 RAM 記憶體不超過 256 位元組，由於 RAM 容量太小，當時 1024 位元 RSA 的簽章無法在晶片內部產生或檢驗)，這已超過多數使用者可容忍的範圍。所以除非有乘法運算加速器(coprocessor)，否則 1024 位元 RSA 不適用於八位元智慧卡。近年來主要的智慧卡晶片供應商多已經提供內含乘法運算加速器的晶片，雖然價格稍貴，但能在低成本的八位元晶片上提供低於 0.5 秒鐘 1024 位元 RSA 簽章的功能。

大多數智慧卡晶片提供的乘法運算加速器指令為計算 Montgomery 乘積(product)[23,24] 。圖 4 介紹如何以 Montgomery 乘積來計算模指數，其中 n 代表 N 的位元組長度 ，而我們的模指數演算法是以一次處理兩個位元的指數為例。若要一次處理更多位元的指數，則需要更多的先行計算時間，同時也會用到更多的 RAM 記憶體。

已知: M, E, N $\quad 0 \le M < N < R = 256^n$

假設指數 e 的二進制表示為 $e = e_{2t-1}e_{2(t-1)}\cdots e_{2j+1}e_{2j}\cdot e_{2j-1}\cdots\cdots e_1e_0$，

MONT(A, B) 代表 Montgomery 乘積，MONT(A, B)= $A * B * R^{-1} \bmod N$

求: C, $\quad C = M^E \mod N$

步驟 1. 先行計算 $\underline{M} = M * R \bmod N$

先行計算 $\underline{M}^3$ = MONT(MONT ($\underline{M}$, $\underline{M}$), $\underline{M}$)

設定 flag = 0

設定 $C = R \bmod N$

步驟 2. for i =t -1 to 0 do if flag = 1, then $C \leftarrow$ MONT(C, C)

case ($e_{2i+1}e_{2i}$)

'00': if flag = 1 do $C \leftarrow$ MONT(C, C)

'01': if flag = 1 do $C \leftarrow$ MONT(C, C); $C \leftarrow$ MONT(C, $\underline{M}$)
else $C \leftarrow$ MONT(C, $\underline{M}$); flag $\leftarrow$ 1

'10': if flag = 1 do $C \leftarrow$ MONT(C, $\underline{M}$); $C \leftarrow$ MONT(C, C)
else $C \leftarrow$ MONT($\underline{M}$, $\underline{M}$); flag $\leftarrow$ 1

'11': if flag = 1 do $C \leftarrow$ MONT(C, C); $C \leftarrow$ MONT(C, $\underline{M}^3$)
else $C \leftarrow$ MONT(C, $\underline{M}^3$); flag $\leftarrow$ 1

步驟 3. $C \leftarrow$ MONT(C, 1)

If the most-significant bit of C is one, then $C \leftarrow C + N$

圖 4：適用於八位元智慧卡晶片的模指數(modular exponentiation)演算法

大多數公開金鑰密碼學演算法都會用到模乘法， $A * B \bmod N$。典型的模乘法是先計算兩數的乘積 $A * B$，然後再進行除法算出餘數。但在智慧卡晶片的 RAM 有限的環境下，我們必須有所修改，盡量少用 RAM。圖 5 以八位元中央處理器為例，說明如何在沒有乘法運算加速器的智慧卡晶片進行高速模乘法的計算，我們假設 A、B、N 皆是位元組長度為 n，且 N 的最有效位元(most significant bit)為 1。Q_h 在步驟 5 的值僅可能為 0、1、或 2，所以我們在步驟 6 用減法來加快速度。在步驟 7，同樣為加快速度，我們用兩個位元組的比較來取代除法。表五則是我們在智慧卡發展系統上執行圖 5 模乘法的效率。

已知: $A, B, N,\ A=\sum_{j=0}^{n-1} A_j 2^{8j},\ B=\sum_{j=0}^{n-1} B_j 2^{8j},\ N=\sum_{j=0}^{n-1} N_j 2^{8j}$

$0 \le A, B < N,\ \frac{2^8}{2} \le N_{n-1} \le 2^8 - 1$

求: $A * B \bmod N$

Step 1. $C \leftarrow 0$ (C 的位元組長度為 n+2)
$i \leftarrow n$ -1

Step 2. $C \leftarrow 2^8 * C + A * B_i$

Step 3. $Q_h \leftarrow \left\lfloor \frac{2^8 C_{n+1} + C_n}{N_{n-1} + 1} \right\rfloor$

Step 4. If Q_h = 0, then goto Step 9
else $C \leftarrow C$ - $2^8 * Q_h * N$

Step 5. $Q_h \leftarrow \left\lfloor \frac{2^8 C_{n+1} + C_n}{N_{n-1} + 1} \right\rfloor$

Step 6. If Q_h = 1, then $C \leftarrow C$ - $2^8 * N$
else if Q_h = 2, then $C \leftarrow C$ - $2^8 * 2N$

Step 7. If $2^8 C_n + C_{n-1} \ge (N_{n-1}+1)(2^8-1)$, then $C \leftarrow C$ - $(2^8\text{-}1)N$

Step 8. $i \leftarrow i - 1$
If $i \geqq 0$, then goto Step 2

Step 9. If $C < N$, then return $C = \sum_{j=0}^{n-1} C_j 2^{8j}$

Step 10. $Q_l \leftarrow \left\lfloor \frac{2^8 C_n + C_{n-1}}{N_{n-1} + 1} \right\rfloor$

Step 11. $C \leftarrow C$ - $Q_l * N$

Step 12. If $C < N$, then return $C = \sum_{j=0}^{n-1} C_j 2^{8j}$

Step 13. $C \leftarrow C$ - N
Goto Step 12

圖 5：適用於八位元智慧卡晶片的模乘法 (modular multiplication)演算法

表五：模乘法於八位元 H8/300 智慧卡執行效率

576 位元	90 毫秒(ms)
768 位元	150 毫秒(ms)
1152 位元	360 毫秒(ms)

模反元素在 RSA 的私密金鑰 D 計算與 ESIGN 的數位簽章產生時需要用到，模反元素的演算法一般用 Lehmer 的方法[22]，但同樣為了節省 RAM 的因素，我們採用圖 6 的演算法，在智慧卡的實際效率則歸納在表六。

已知: A, N
求: T 滿足 $T * A \equiv 1 \mod N$

Step 1. $C \leftarrow N$
$D \leftarrow A$
$X' \leftarrow 0$
$X \leftarrow 1$
$counter \leftarrow 0$

Step 2. If D is more than on byte, then $Q \leftarrow \left\lfloor \frac{C}{D_{top}+1} \right\rfloor$
(D_{top} is the non-zero most-significant digit of D)
else $Q \leftarrow \left\lfloor \frac{C}{D} \right\rfloor$
If $Q = 0$, then $Q \leftarrow 1$
C (new) $\leftarrow D$ (old)
D (new) $\leftarrow C$ (old) - $Q * D$ (old)

Step 3. $D = 0$, then goto Step 5

Step 4. X' (new) $\leftarrow X$ (old)
X (new) $\leftarrow X'$ (old) + $Q * X$ (old)
If ($C \leqq D$), then swap C with D, swap X with X'
else $counter \leftarrow counter$ +1
Goto Step 2

Step 5. If (counter = 1 mod 2), then return $N - X$
else return X

圖 6：適用於智慧卡的模反元素(modular inverse)演算法

表六：模反元素於八位元 H8/300 智慧卡執行效率

192 位元	120 毫秒
256 位元	200 毫秒
384 位元	470 毫秒

ESIGN 可在沒有乘法運算加速器的情況下，於八位元智慧卡內高速產生 1024 位元以上數位簽章。表七為我們利用圖 6 與圖 7 的模乘法與模反元素演算法於 H8/300 智慧卡硬體執行的結果，ROM 的大小為 2.8K 位元組，而全部 RAM 的佔用區域為 466 位元組。

表七：1152 位元金鑰之 ESIGN 於八位元 H8/300 智慧卡執行效率

Pre-computation	6.0 秒
數位簽章產生時間(E = 1024)	0.15 秒
數位簽章檢驗時間(E = 1024)	3.7 秒

伍、結論

難以偽造的 IC 卡的使用是電子化社會的趨勢，一張 IC 卡不僅可用做身份識別，也可提供門禁管制、電子公文、網路報稅等功能。這使得 IC 卡內部密碼學演算法的實現日行重要。本文介紹如何在 IC 卡內部實現包含數位簽章的密碼學演算法，以使政府大力推行 IC 卡時能掌握關鍵性技術，我們也期待未來產官學能以更嚴謹的態度訂定 IC 卡規範。

參考文獻

[1] A. J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography* (CRC Press Series on Discrete Mathematics and Its Applications), 1996.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, Feb. 1978, Vol. 21, No. 2, pp. 120-126; "Cryptographic Communications System and Method", U.S. Patent #4,405,829, Sep. 1983.

[3] National Institute of Standards and Technology, "Data Encryption Standard", Federal Information Processing Standard, *FIPS PUB 46-2*, December 1993. 中華民國國家標準 CNS X5011 "數據保密(加/解密)運算法", 1983年公佈。

[4] "Cracking DES code all in a day's work for security experts", http://www.cnn.com/TECH/computing/9901/21/descrack.idg/index.html 或見 "RSA Code-Breaking Contest Again Won by Distributed.Net and Electronic Frontier Foundation (EFF) - DES Challenge III Broken in Record 22 Hours", http://www.rsasecurity.com/news/pr/990119-1.html.

[5] J. Demen and V. Rijmen, "The Rijndael Block Cipher", Document version 2, March 1999, http://www.esat.kuleuven.ac.be/~rijmen/rijndael.

[6] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)", Federal Information Processing Standard, *FIPS PUB 197*, November 26, 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[7] 邱榮輝，認識IC卡，*資訊安全通訊*，第四卷第三期，民國87年6月，頁18~30。

[8] ISO 7816 Part 1 to 10: *Identification Cards – Integrated Circuit(s) Cards with Contacts*, 1987 to 2000.

[9] W. Rankl and W. Effing, *Smart Card Handbook*, 2nd edition, John Wiley & Sons, 2000.

[10] "DoD Issues Time-Saving Common Access Cards", Oct. 2000. http://www.defenselink.mil/news/Oct2000/n10102000_200010107.html.

[11] National Institute of Standards and Technology, Federal Information Processing Standard (FIPS) 46-3, Data Encryption Standard, October 25, 1999, http://csrc.nist.gov/ fips/fips46-3.pdf.

[12] "New European Schemes for Signatures, Integrity, and Encryption (NESSIE)," http://www.cosic.esat.kuleuven.ac.be/nessie/.
[13] "Call for Cryptographic Techniques", IPA, June 2000, http://www.ipa.go.jp/security/enc/CRYPTREC/index-e.html.
[14] Tatsuaki Okamoto, "A Fast Signature Scheme Based on Congruential Polynomial Operations", *IEEE Trans. Info. Theory*, Vol. 36, pp. 47-53, January 1990. 參見 http://info.isl.ntt.co.jp/esign/ 或 https://www.cosic.esat.kuleuven.ac.be/nessie/updatedPhase2Specs/esign/esign5.pdf
[15] *Hitachi Single-Chip Microcomputer H8/3113 Hardware Manual*, Hitachi Ltd., 1998. 參見 http://www.hitachisemiconductor.com/sic/jsp/japan/eng/solutions/ic_cards_solution/image/english/h8_3114e.pdf.
[16] National Institute of Standards and Technology, *Digital Signature Standard (DSS)*, Federal Information Processing Standard, FIPS 186, May 1994.
[17] ISO/TC68/SC21會議資料，1992年2月。
[18] Chung-Huang Yang, "Modular Arithmetic Algorithms for Smart Cards," *IEICE Technical Report on Information Security* (日本電子情報通信學會資訊安全技術報告), ISEC92-16, August, 1992.
[19] 楊中皇、蔡金鳳，新一代對稱式密碼學演算法於智慧卡上的具體實現，*第十一屆全國資訊安全會議*，2001，頁199-206。
[20] The RSA Challenge Numbers, http://www.rsasecurity.com/rsalabs/challenges/factoring/numbers.html.
[21] ISO/IEC 14888-3, Information technology -- Security techniques -- Digital Signatures with Appendix - Part 3: Certificate-Based Mechanisms, Dec. 1998.
[22] Donald E. Knuth, *The Art of Computer Programming - Seminumerical Algorithms*, Vol. 2, Section 4.3, Addison-Wesley, 1981.
[23] Peter L. Montgomery, "Modular Multiplication without Trial Division," Mathematics of Computation, pp. 519-521, 1985.
[24] Stephen R. Dusse and Burton S. Kaliski Jr., "A Cryptographic Library for the Motorola DSP56000," *Proc. Eurocrypt'90*, pp. 230-244, 1990.