行政院國家科學委員會補助專題研究計畫成果報告

※ ※
※ **寬 頻 分 碼 多 重 進 接 無 線 通 訊 之 加 解 密 系 統** ※
※ **Cryptosystems for Wide-band CDMA Wireless** ※
※ **Communications** ※
※ ※

計畫類別：□個別型計畫 ✓整合型計畫

計畫編號：NSCー89ー2219ーEー009ー016

執行期間：88 年 8 月 1 日至 89 年 7 月 31 日

計畫主持人：王聖智

執行單位： 國立交通大學電子研究所

中 華 民 國 八十九 年 十一 月 日

行政院國家科學委員會專題研究計畫成果報告
計畫名稱：寬頻分碼多重進接無線通訊之加解密系統
Cryptosystems for Wide-band CDMA Wireless Communications
計畫編號：NSC-89-2219-E-009-016
執行期限：88 年 8 月 1 日至 89 年 7 月 31 日

主持人：王聖智 (交通大學電子工程系副教授)
計畫參與人員：蔡維昌、連璧賢、柯瑞泰 (交通大學電子所研究生)

一、中文摘要

本年度計畫的結果進度包含三個部分：RSA **的硬體架構設計、利用** DSP **系統模擬實現** RSA **架構以及進行** Elliptic Curve **密碼系統演算法之研究並針對在** Elliptic Curve **中主要運算** - finite field multiplication **，提出一個新的演算法架構，以加快** Elliptic Curve **密碼系統的運算速度。**

關鍵詞：密碼學，RSA **密碼系統，橢圓曲線密碼系統，公匙密碼系統**

Abstract

This report includes three parts： the RSA hardware architecture design, the simulation of RSA hardware architecture on a DSP system, and a new architecture design for the major computation – finite field multiplication in Elliptic Curve Cryptosystems.

Keywords ： Cryptography, RSA Cryptosystem, Elliptic Curve Cryptosystem, Public-Key Cryptosystem

二、緣由與目的：

無線通訊上資料的安全問題已經逐漸的受到重視，為了避免被截聽，故於通訊過程中加上資料的加解密，才能確保安全。

本計畫的目標是針對 RSA **公匙密碼系統和橢圓曲線密碼系統，研究兩者的演算法，找出合適於** WCDMA **的方法並加以實現。除了公匙密碼系統的演算法之外，亦將進行關於架構的設計、速度、和低功率等方面之研究，並將討論與其它** WCDMA **設計區塊之整合與界面之處理。**

三、結果與討論

(1) RSA 的硬體架構設計

我們嘗試運用 Systolic Array **之架構來實現一個** RSA **加解密系統，此架構中最主要的運算就是乘法冪，此處我們採用的是以** Montgomery Algorithm **為主體的實現方式。利用兩個主要的觀念：**Partitioning **及** Pairing-off**，實現硬體上之加速乘法冪的運算。**

原始實現乘法冪的每個乘法運算區塊如下，A**、**B**分別為乘數和被乘數，**N**為冪數，**R**為乘法冪的餘數，每一次的區塊運算中的每一個位元都和前一次區塊的前一個位元有關連性：**

$$A=\sum_{i=0}^{m-1}a_i2^i, B=\sum_{i=0}^{m-1}b_i2^i, N=\sum_{i=0}^{m-1}n_i2^i, q_i=(R_i)_0\oplus a_ib_0$$

$$(R_i)_{j-1}+2(Carry1_i)_j+2(Carry2_i)_j=$$
$$a_ib_j+q_in_j+R(_{i-1})_j+(Carry1_i)_{j-1}+(Carry2_i)_{j-1}$$

以下分別是我們所提出，改進乘法冪運算的兩個設計。

(i) Partitioning

將原始的式子分解成兩式，再利用 pre-computation **的技巧計算初始值，使一個** clock cycle **中的運算減少。**

$$A=\sum_{i=0}^{m-1}a_i2^i, B=\sum_{i=0}^{m-1}b_i2^i, N=\sum_{i=0}^{m-1}n_i2^i, q_i=(R_i)_0\oplus a_ib_0$$

$$\begin{cases}(P_i)_j+2(CP_i)_j=a_ib_j+q_in_j+(CP_i)_{j-1}\\(R_i)_{j-1}+2(CR_i)_j=(P_i)_j+(R_{i-1})_j+2(CR_i)_{j-1}\end{cases}\quad\begin{cases}(CP_i)_j=(Carry1_i)_j\\(CR_i)_j=(Carry2_i)_j\end{cases}$$

若是考慮了 pre-computation **後，以上的式子將可轉換成：**

$$q_i=(R_{i-2})_1\oplus(P_{i-1})_1\oplus(CR_{i-1})_0\oplus a_ib_0$$
$$(P_i)_0=(R_{i-2})_1\oplus(P_{i-1})_1\oplus(CR_{i-1})_0$$
$$(CP_i)_0=a_ib_0\bullet\overline{((R_{i-2})_1\oplus(P_{i-1})_1\oplus(CR_{i-1})_0)}$$

在 Fig.1 **中，我們運用模型呈現出** partitioning **的實際情況。**

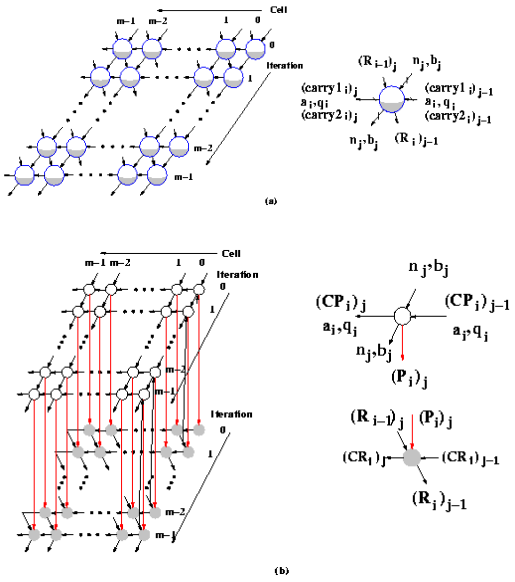

Fig.1. Data dependency of Montgomery's algorithm (a) original architecture (b) partitioned architecture

我們將用這個架構平行運算以充份利用每個硬體而不會浪費等待資料的時間。但是從產生 $(R_i)_j$ **這個位元到下一個位元** $(R_{i+1})_j$ **之間仍然需要一個** clock cycle **的等待時間，也就是** one clock cycle gap problem**。雖然** partitioning **的架構不會解決這個問題，但是它會使每一個** clock cycle **所需的運算減少，而加快了運算的速度。**

Fig.2 **為** partitioning **運算乘法冪的時序圖，而** Fig.3 **為** Partitioning **的** Systolic Array **架構。因為每一個基本的運算架構都是固定的，所以我們得到** Fig.4 (b)**中所示的** 1-D **的** systolic array**。**

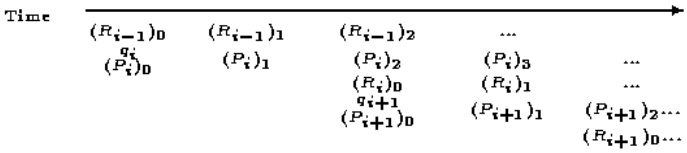

Fig. 2. Timing sequence of $(P_k)_i$ and $(R_k)_i$ after removing some data dependencies

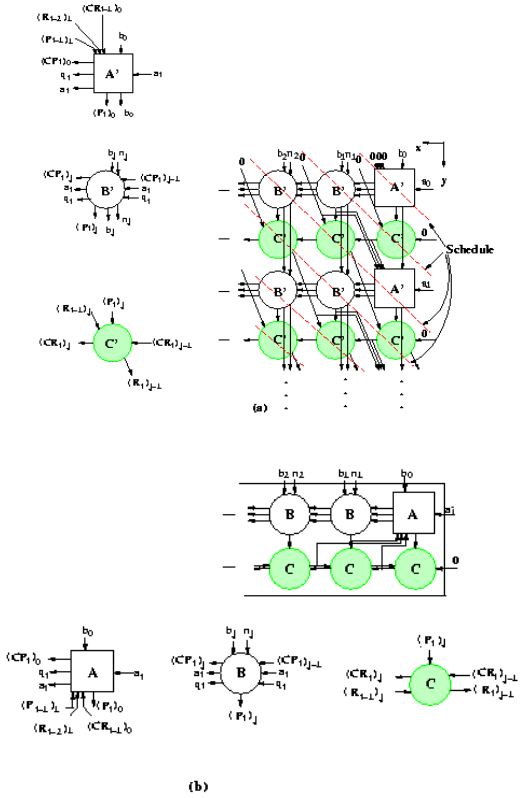

Fig. 3. Two-dimensional and one-dimensional structures of the double –layer systolic array

(ii) Pairing-off

如果將兩個位元配對成位元組，平行處理乘法冪的運算，也就是在一個 clock cycle **中，同時平行處理多個位元組。雖然位元組之間有** dependency**，但是因為每個位元組在一個** clock cycle **後半段的運算需要來自較低位元組前半段**

的運算結果，所以整體而言，是不需要多餘的延遲時間等待的，所以速度上不會受影響。而且這樣的設計在計算前後高低的不同位元時，是不需要等待一個 clock cycle，所以不會有 one clock cycle gap problem。以下是合併配對的數學式。

$$4(CP_i)_j + 2(P_i)_{2j} + (P_i)_{2j-1} =$$
$$2a_ib_{2j} + 2q_in_{2j} + a_in_{2j-1} + q_in_{2j-1} + (CP_i)_{j-1}$$

$$4(CR_i)_{j-1} + 2(R_i)_{2j-2} + (R_i)_{2j-3} =$$
$$2(R_{i-1})_{2j-1} + (P_i)_{2j-1} + (R_{i-1})_{2j-2} + (P_i)_{2j-2} + (CR_i)_{j-2}$$

Fig.4 就是 pairing-off 的架構圖。可以看到 clock cycle k 時可運算得到$(R_i)_j$的值，而在 clock cycle k+1 時，就可接由運算而得到$(R_{i+1})_j$的結果，不需要等待一個 clock period 的時間。

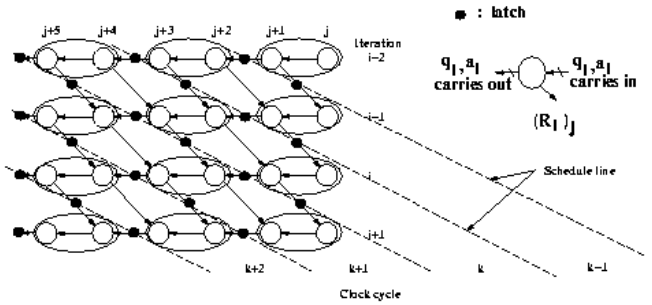

Fig. 4. Pairing-off style of the non-interlaced architecture

(2) DSP 系統模擬實現 RSA 架構

在此部份，目前我們已經熟悉 TI 的 DSP 發展系統，了解此系統上之硬體規格架構及使用介面。由於此系統中之硬體運算資源有限，加上所使用之字元長度(word length)為 32 bits，目前我們正嘗試修改我們的運算架構，以便於在 DSP 系統上模擬出 RSA 系統中整個加解密的過程。

基本上，我們在模擬整個 RSA 公匙密碼系統的標準時，必須先產生兩個強大的質數 p，q 作為秘密金匙，還要產生兩個數 e，d，使 $ed = 1 \bmod (p-1)(q-1)$，分別作為加、解密的公開金匙以及秘密金匙，而 RSA 的標準，是針對一組 512 – bit 的資料加、解密，以確保安全性。所以我們在 DSP 系統上設計時，也是試著以 512 – bit 的大小為一組資料處理，所以在 DSP 系統上，我們需要 16 個 words 大小來處理一組資料，所以 words 之間的溢位、借位問題也是我們主要的考量問題。

(3) Elliptic Curve 密碼系統演算法之研究發展

Elliptic Curve Cryptosystem (ECC) 一般是架構在 F_q 或 F_2^m 上，而 F_q 主要是作 modulo q 的運算，若是運用 homogenous 座標系，可消除在 ECC 中所需要的除法運算，而只需進行乘法的運算。因此，一個 ECC 系統之實現上，主要的運算將是架構在 F_q 或是 F_2^m 上的乘法冪運算。目前我們已完成整個 ECC 系統之研究，正著手以 Systolic Array 的架構來設計實現。

在 Elliptic curve cryptosystem (ECC)中，我們採用了在有限域 $GF(2^m)$ 上 non-supersingular Elliptic curve 曲線，而有限域中的有意義的數值是定義在曲線的有限點上，而曲線的定義如下：

$$E: y^2 + xy = x^3 + ax^2 + b$$

where a, $b \in GF(2^m)$ and $b \neq 0$

如果 $P = (x_1, y_1)$ ， $Q = (x_2, y_2) \neq P$ 且 $P, Q \in E$ ，則定義兩點 P, Q 的加法， $P + Q = (x_3, y_3) \in E$，而它們的數學運算關係如下：

$$x_3 = \begin{cases} (\frac{y_1 + y_2}{x_1 + x_2})^2 + \frac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + a & P \neq Q \\ x_1^2 + \frac{b}{x_1^2} & P = Q \end{cases}$$ ，and

$$y_3 = \begin{cases} (\frac{y_1 + y_2}{x_1 + x_2})(x_1 + x_3) + x_3 + y_1 & P \neq Q \\ x_1^2 + (x_1 + \frac{y_1}{x_1})x_3 + x_3 & P = Q \end{cases}$$

而在 $GF(2^m)$ 上的加法、乘法等運算的基底，我們採用了 standard basis representation ，而兩個數 $A(\alpha)$ 和 $B(\alpha)$ 均定義是屬於有限域

$GF(2^m)$ 中，數學表示式如下：

$$A(\sim)=\sum_{i=0}^{m-1}a_i\sim^i=a_{m-1}\sim^{m-1}+a_{m-2}a^{m-2}+\ldots+a_1\sim+a_0 \quad \text{and}$$

$$B(\sim)=\sum_{i=0}^{m-1}b_i\sim^i=b_{m-1}\sim^{m-1}+b_{m-2}a^{m-2}+\ldots+b_1\sim+b_0$$

而在有限域 $GF(2^m)$ 上的加法表示式如下：

$$S(\sim)=A(\sim)+B(\sim)=\sum_{i=0}^{m-1}s_i\sim^i$$

where $s_i = a_i \oplus b_i$

而乘法冪的演算法如下面所示，主要是移位 (shift) 和求餘數的運算 (mod)：

Multiplication Algorithm over $GF(2^m)$ by using Modulus Operation

$R^0(\sim)=0;$

for i = 1 to m

$$R^i(\sim)=(R^{i-1}(\sim)\sim+a_{m-i}B(\sim))(\bmod F(\sim))$$

end

$$P(\sim)=R^m(\sim)$$

where a_j is the jth coefficient of $A(\alpha)$, $R^i(\alpha)=\sum_{j=0}^{m-1}r_j^i\sim^j$ and $a_{m-i}B(\sim)=\sum_{j=0}^{m-1}(a_{m-i}b_j)\sim^j$

在這個乘法的演算法中，可以發現最主要的運算就是 $R^i(\sim)=(R^{i-1}(\sim)\sim+a_{m-i}B(\sim))(\bmod F(\sim))$。經由化簡我們可以得到在第 i 個 iteration 中的第 j 個位元的運算是 $r_j^i=r_{m-1}^{i-1}f_j\oplus r_{j-1}^{i-1}\oplus a_{m-i}b_j$ ，而以 systolic Array 的架構來實現這一個 finite field multiplication。

針對這個架構在 systolic Array 上的 finite field multiplication 動作，我們仿照先前在 RSA 中的設計，套用了 partitioning 和 pairing-off，來加速這個運算。partitioning 是將 finite field multiplication 的運算分解，減少 clock period ，式子如下。

$$p_j^i=r_{m-1}^{i-1}\oplus a_{m-i}b_j$$

$$r_j^i=r_{j-1}^{i-1}\oplus p_j^i$$

pairing-off 是將 partitioning 後結果，合併沒有關聯性的部分，拿掉多餘的 Flip-Flop，並且作最佳化的處理。結合這兩個部分，我們預期將可以加速 finite field multiplication 的運算速度，目前，此設計之硬體實現正在進行中。

四、計畫結果自評

在 RSA 的硬體設計中，我們針對 1024-bit 的 partitioning 架構設計，可以得到一個 cycle time 只需要 2.0ns，面積約為 240 K 個電晶體大小的實際硬體結果。而 1024-bit 的 paring-off 架構的運算，經由估計得到的一個 cycle time 為 2.7ns ，面積約為 209K 個電晶體的大小的硬體結果。我們設計出這兩種硬體架構經由模擬測試結果顯示，在速度及硬體面積上，都較現有的其他架構為佳。

用 DSP 系統模擬實現 RSA 的架構目前尚未完成。

另外，針對在 Elliptic Curve 密碼系統中主要運算 – finite field multiplication 的運算，我們已經提出一個演算法，而正要實現和驗證。

五、參考文獻

[1] W. Diffie and M. Hellman, “New Directions in Cryptography”, IEEE Transactions on Information Theory, vol. IT-22, pp. 644-654, November 1976.

[2] R.Rivest, A. Shamir and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” Communications of the ACM, vol. 21, pp. 120-126, February 1978.

[3] D.Denning, “Cryptography and data security”, Addison-Wesley, 1982

[4] S. Bandyopadhyay and A. Sengupta, “Algorithms for multiplication in Galois field for implementation using systolic arrays”, IEE Proceedings, Vol. 135, Pt. E, No. 6, pp.336-340. November, 1988.