IC カード用剰余演算アルゴリズム
Modular Arithmetic Algorithms for Smart Cards

楊 中皇
Chung-Huang Yang

NTT 情報通信網研究所
NTT Network Information Systems Laboratories

Abstract

Most general-purpose smart cards feature limited RAM/ROM and slow CPU which make them traditionally unsuitable for public-key cryptosystems or signature schemes aimed at real-time applications. The modular arithmetic algorithms for ESIGN on a general-purpose smart card (the Hitachi H8/310) are described here. Our implementation results show that the computation time for 768-bit modular multiplication is approximately 150 ms, for 256-bit modular inverse is 200ms, and 768-bit ESIGN signature generation is computed in less than 0.8 seconds on the H8/310 running at 5 MHz.

1. Introduction

Most general-purpose smart cards[1] feature limited RAM/ROM and slow CPU which make them traditionally unsuitable for public-key cryptosystems or signature schemes aimed at real-time applications. While there has been enormous publications (see, for example, [2]) on modular arithmetic algorithms, few of them considering the smart card environment. Here we describe the modular arithmetic algorithms for ESIGN[3] on a general-purpose smart card (the Hitachi H8/310)[4]. Computational steps in the signature generation phase of ESIGN are as follow:

$$W = \left\lceil \frac{H(M) - X^K \mod N}{PQ} \right\rceil \tag{1}$$

$$Y = X \times W \times (KX^K)^{-1} \mod P \tag{2}$$

$$S = X + Y \times PQ \tag{3}$$

where $\lceil\ \rceil$ denote the integer ceiling function, P and Q are private keys (say, 256 bits each), N $(= P^2 \times Q)$ is the public key, H is a one-way hash function, X is a random

number, $K \geq 4$ is the system parameter (power of two), and S is the signature for message M.

The point, however, is not in the performance alone. Referring to Fig. 1, we are given a 96-byte operand X on RAM and required to compute X^2 and put the result on RAM. Conventional method of multiplication following by division approach is infeasible since total amount of the built-in RAM is only 256 bytes (the H8/310).

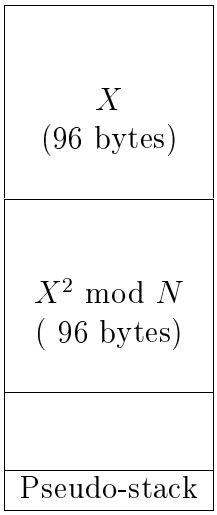

Figure 1. 256-byte Internal RAM Usage

Besides, the implementation complexity (in terms of code size) of an algorithm is as important as the algorithm performance on the smart card environment. In other words, we have to carefully evaluate the tradeoff between program size and performance.

2. The Algorithms

2.1 Addition and Subtraction

The standard right-to-left $O(n)$ algorithms are used.

2.2 Multiplication and Division

The conventional $O(n^2)$ methods are used.

2.3 Modular Multiplication

This is the most time-consuming routine for ESIGN signature generation and verification tasks and special attention is on tuning both algorithm and implementation performance. Our algorithm is shown in Fig. 2 and a thorough discuss is given in [5].

2.4 Modular Inverse

The proposed algorithm, see Fig. 3, like the extended Euclidean algorithm on which it is based, used approximation division instead of direct long division (see Step 2).

Given: A, B, N of n-byte positive integers , $A = \sum_{j=0}^{n-1} A_j 2^{8j}$, $B = \sum_{j=0}^{n-1} B_j 2^{8j}$
$N = \sum_{j=0}^{n-1} N_j 2^{8j}$, $0 \le A,\ B \le N$, $\frac{2^8}{2} \le N_{n-1} \le 2^8 - 1$

Find: $A \times B \bmod N$

Step 1. $C \leftarrow 0$ (C is a variable of $n+2$ byte)
$i \leftarrow n-1$

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

Step 3. $Q_h \leftarrow \left\lfloor \frac{2^8 C_{n+1} + C_n}{N_{n-1} + 1} \right\rfloor$, $0 \le Q_h \le 2^8 - 1$

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

Step 5. $Q_h \leftarrow \left\lfloor \frac{2^8 C_{n+1} + C_n}{N_{n-1} + 1} \right\rfloor$, $0 \le Q_h \le 2$

Step 6. If $Q_h = 1$ then $C \leftarrow C - 2^8 \times N$
If $Q_h = 2$ then $C \leftarrow C - 2^8 Q_h \times N$

Step 7. $Q_h \leftarrow \left\lfloor \frac{2^8 C_{n+1} + C_n}{N_{n-1} + 1} \right\rfloor$, $0 \le Q_h \le 1$

Step 8. If $Q_h = 1$ then $C \leftarrow C - 2^8 \times N$

Step 9. $Q_l \leftarrow \left\lfloor \frac{2^8 C_n + C_{n-1}}{N_{n-1} + 1} \right\rfloor$, $0 \le Q_l \le 2^8 - 1$

Step 10. If $Q_l = 2^8 - 1$ then $C \leftarrow C - Q_l \times N$

Step 11. $i \leftarrow i - 1$
If $i \ge 0$ then goto Step 2

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

Step 13. $Q_l \leftarrow \left\lfloor \frac{2^8 C_n + C_{n-1}}{N_{n-1} + 1} \right\rfloor$

Step 14. $C \leftarrow C - Q_l \times N$

Step 15. If $C < N$ then return $C = \sum_{j=0}^{n-1} C_j 2^{8j}$

Step 16. $C \leftarrow C - N$
Goto Step 15

Figure 2. The Modular Multiplication Algorithm for the H8/310

Given: Multi-precision integers A and P, $0 < A < P$, P is a prime

Find: A^{-1} *mod* P

Step 1. $C \leftarrow P$
$D \leftarrow A$
$X' \leftarrow 0$
$X \leftarrow 1$
counter $\leftarrow 0$ (*counter* is a single-byte variable)

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

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

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

Step 5. If (*counter* $\equiv 1 \mod 2$), then return $P - X$
else return X

Figure 3. The Modular Inverse Algorithm for the H8/310

Surprisingly, our experiences indicate such approach provides a 10 to 20 fold performance improvement over the original extended Euclidean algorithm.

3. Implementation

The above algorithms were implemented on Hitachi's H8/310 8-bit single-chip microcomputer running at 5 MHz. Total program size is about 3 kilo-bytes and the performance of our algorithms for ESIGN is summarized in Table 1.

	Size of Public Key N	
	576 bits	768 bits
modular multiplication	90 ms (576 bits)	150 ms (768 bits)
modular inverse	120 ms (192 bits)	200 ms (256 bits)
signature generation	450 ms	750 ms
signature verification	270 ms	460 ms

Table 1. ESIGN execution time, $K = 8$, on the H8/310

References:

1. Louis Claude Guillou, Michel Ugon, and Jean-Jacques Quisquater, "The Smart Card: A Standardized Security Device Dedicated to Public Cryptology," in *Contemporary Cryptology: The Science of Information Integrity*, (Ed.) Gustavus J. Simmons, pp. 561-614, IEEE Press, 1992.

2. Donald E. Knuth, *The Art of Computer Programming - Seminumerical Algorithms*, Vol 2, Section 4.3, Addision-Westley, 1981.

3. Tatsuaki Okamoto, "A Fast Signature Scheme Based on Congruential Polynomial Operations," *IEEE Trans. Info. Theory*, Vol. 36, pp. 47-53, January 1990.

4. *Hitachi Single-Chip Microcomputer H8/310 Hardware Manual*, Hitachi Ltd., 1990.

5. Hikaru Morita and Chung-Huang Yang, "Lookahead Determination for Modular-Multiplication," to appear in *IEICE Trans.*, Vol. E76-A, No. 1, January 1993. See also, C.H. Yang and H. Morita, "An Efficient Modular-Multiplication Algorithm for Smart-Card Software Implementation," *IEICE Japan*, Technical Report, ISEC91-58, January 1992.