SCIS 2003 The 2003 Symposium on Cryptography and Information Security
Hamamatsu, Japan, Jan.26-29, 2003
The Institute of Electronics, Information and Communication Engineers

On the Performance of ESIGN and RSA on IC Cards

楊 中皇 [1] 森田 光 [2] 岡本 龍明 [3]
Chung-Huang Yang Hikaru Morita Tatsuaki Okamoto

Abstract The advancement of information and communications technology, especially the Internet, has created an opportunity to improve the administrative efficiency and service quality in governments of many nations. However, due to the lack of communication security services, sensitive documents could not be transmitted securely over open networks using off-the-shelf software. Digital signature is by far one of the most important cryptographic techniques used in the e-government and e-commerce applications. It provides authentication of senders or receivers (they are who they claim to be) and offers non-repudiation of transmission (senders can't deny their digital signature in the signed documents and the document cannot be altered in transmission without being detected). This paper presents our efforts in the implementation of digital signature algorithms on IC cards. We evaluated the performance of well-known ESIGN and RSA digital signature algorithms on the Hitachi H8/300 8-bit IC card chips.

Keyword IC card, digital signature, PKI, ESIGN, RSA

1 Introduction

IC cards [1-2] are much more difficult to duplicate than magnetic strip cards and cryptographic functions can be implemented inside these cards. With substantial cost reductions and the ability to handle multiple applications on a single card, the IC cards are about to enter a period of the rapid growth. An individual bearing a single IC card will be able to electronically and securely interact with several servers or service provides. As a consequence, an entirely new type of commercial and educational landscape is being created. However, IC card itself has limited computing power and memory capacity. This makes it a difficult job to efficiently implement the cryptographic functions inside IC card.

One of the e-government's [3] objectives is to facilitate the exchange and integration of information between different agencies and the Internet is being used as the communication channel to exchange information between all sectors of society. However, due to the lack of communication security services and the export control of U.S.A. and many nations, sensitive information could not be securely transferred between and within governmental agencies over the Internet using off-the-shelf software. Here, the security services [4] mean data confidentiality, authentication, access control, data integrity, and non-repudiation.

Cryptography [5-6] is the only practical means for providing security services over an insecure channel such as Internet. The increasing use of electronic means of data communications, coupled with the growth of computer usage, has extended the need to protect information. Considerable progress has been made in the techniques for encryption, authentication, and fending off attacks from intruders over the last decade. Nevertheless, the impose of export controls on those computer software or hardware devices has precluded the use of secure products or has made such imported products very expensive.

In the public-key cryptography [4] or public-key scheme, each entity has a pair of public key and corresponding private key; private key shall be kept secretly at every entity and could be individually stored at the built-in EEPROM area of the IC card while the public key is openly available to each other entity. Public-key infrastructures (PKI) [6] are comprised of supporting services that are needed for using public-key technologies on a large scale and are widely adopted in the e-government projects worldwide. Digital signature is a core technique in the PKI.

Digital signature [5-6] is by far one of the most important cryptographic techniques employed in the e-government and e-commerce applications. A digital signature algorithm allows an entity to use its private key to electronically sign a message and generate a signature that is dependent on both message itself and the entity's

[1]国立高雄師範大学, National Kaohsiung Normal University , 116, Ho Ping First Road, Kaohsiung 802, Taiwan, R.O.C. http://www.crypto.idv.tw/
[2]ＮＴＴサービスインテグレーション基盤研究所 〒180-8585 東京都武蔵野市緑町3-9-11、本館604, NTT Service Integration Laboratories, Main Bldg. 604, 3-9-11 Midori-cho, Musashino-Shi, Tokyo 180-8585, JAPAN
[3]ＮＴＴ情報流通プラットフォーム研究所 〒239-0847 神奈川県横須賀市光の丘1-1, NTT Information Sharing Platform Laboratories, 1-1 Hikarinooka, Yokosuka-Shi, Kanagawa 239-0847, JAPAN

private-key information. The computed signature is then attached to the message and sent with the message. The signature verification process could be performed by any receiving party and cannot be repudiated. Recipient could verify the signature by doing some computation involving the received message, the attached signature, and sender's public key. If the results properly hold in a predefined mathematical relation, the signature is accepted as genuine. Otherwise, the signature may be fraudulent or the message altered.

In this paper, we carry out the implementation of digital signature algorithms on IC cards. New arithmetic algorithms are proposed and implemented and we also evaluated the performance of well-knows ESIGN [7] and RSA [8] digital signature algorithms on the Hitachi H8/300 8-bit IC cards. In order to understand the cost-effectiveness of different IC cards, we evaluated digital signature schemes on the low-cost 8-bit chip without using the coprocessor.

2 The 8-bit IC Cards

The IC card device we adopted for implementing and evaluating digital signature algorithms is the Hitachi H8/3113 [9] IC card device. It contains an 8-bit H8/300 microprocessor, 32-Kbyte ROM, 16-Kbyte EEPROM, 2.5K-byte RAM, an arithmetic coprocessor, and a random number generator. Maximum external clock rate is 10 MHz.

We currently are also implementing digital signature algorithms on the Infineon SLE66CX322P [10] IC card device. It contains an 8051-compatible CPU, 134-Kbyte ROM, 32-Kbyte EEPROM, 4K-byte RAM, an arithmetic coprocessor, and a random number generator. Implementation results will be given in the future.

Assembly codes were written to evaluate the performance of ESIGN and RSA algorithms in the IC card IC. Hardware emulator, as shown in Figure 1, is used to emulate the performance.

Figure 1. Hitachi (left) and Infineon (right) IC card development tools

3 The ESIGN and RSA Digital Signature Algorithms

The RSA [8] digital signature scheme was first published in 1978. In this scheme, each entity first chooses two large prime numbers. Let's denote as P and Q, as *private key*. Then a public exponent E is chosen such that GCD(E,(P-1)(Q-1))=1 is satisfied, which means the greatest common divisor (GCD) of positive integer E with (P-1)(Q-1) equals to one. Now, we apply the extended Euclidean algorithm [10] to find a positive integer D satisfies

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

where *mod* denotes the modular arithmetic [11]. The *public key* of entity is E and N *(= P. Q)* while the private key is *D and P and Q;* private key might be stored inside IC cards and is needed for generating digital signature while public key is needed for verifying digital signature. In practice, a message digest algorithm, such as the NIST's SHA-2 [12], is used with the RSA algorithm for signature generation and signature verification. Therefore, instead of directly applying RSA digital signature scheme on a long message, we could efficiently sign on the message's hash value, signature $S = SHA(M)^D \bmod N$.

ESIGN (abbreviation for Efficient digital SIGNature) digital signature scheme was first proposed in 1990 and is now a part of the ISO 14888 international standard [13]. ESIGN is one of the crypto algorithms selected for the 2nd phase of the NESSIE (New European Schemes for Signatures, Integrity and Encryption) [14] project and also for the IEEE P1363a [15] standard. There are variants of ESIGN scheme with different security strength, from the classical digital signature algorithm to the provably secure against single occurrence chosen message attack (SO-CMA) and to the provably secure against chosen message attack (CMA) [9]. In this paper, we will use ESIGN to denote the core component of all ESIGN's variants where we are mainly concerned about the implementation issues. In the basic ESIGN scheme, each entity chooses two large prime numbers, let's also denote as P and Q but the *public key* of entity is P^2Q ($= N$) while the private key is P and Q. A public security parameter E (exponent) also is needed, $8 \le E$. To generate a signature on a message M; first we need a (encoding) random number R, $0 \le R \le PQ$-1. Let the bit length of P be k, then we compute the signature S by the following equations:

$$W_0 = \left\lceil \frac{\left(f \| 0^{2k}\right) - R^E \bmod N}{PQ} \right\rceil \quad (1)$$

$$T = W_0 \times \left(E \times R^{E-1}\right)^{-1} \bmod P \quad (2)$$

$$S = R + T \times PQ \quad (3)$$

where $\lceil x \rceil$ denotes the ceiling function for the largest integer less than x, f is a k-bit (hashed) message representative, and $\left(f \| 0^{2k}\right)$ is a $3k$-bit value obtained by putting 2k-bit zeroes as the least significant bits. Signature verification process involves the computation of S^E *mod* N and check for the equality of the most significant k bits of $S^E \bmod N$ with the message representative f.

4 The Implementation of Digital Signature Algorithms on 8-bit IC Cards

Multiple-precision modular arithmetic [11] is required at both ESIGN and RSA digital signature algorithms. However, most general-purpose IC cards feature limited RAM/ROM and slow 8-bit CPU that make them traditionally unsuitable for public-key cryptosystems or digital signature schemes aimed at real-time applications. While there have been enormous publications (see, for example, [11]) on modular arithmetic algorithms, few of them considering the IC card environment where RAM area is usually only 2K bytes or less and this represents a major implementation constrain.

In the following, we describe three modular arithmetic algorithms: modular exponentiation, modular multiplication, and modular inverse, which are required for implementing ESIGN and RSA digital signature scheme on the H8/300 general-purpose IC cards. Figure 2 shows the modular exponentiation algorithm. We scan 2 bits of exponent at once in Figure 2 and more performance improvement can be achieved if more bits are processing at once but more program ROM size and data RAM size would be required.

Given: M, E, N $0 \le M < N < R = 256^n$
Find: $C, C = M^E \mod N$

Solution:

Let PROD(A, B) indicate the operation of modular multiplication, PROD(A, B)= $A * B \mod N$

Assume that binary presentation of exponent E is $e_{2t-1}e_{2(t-1)}\cdots e_{2j+1}e_{2j}e_{2j-1}\cdots e_1e_0$,

Step 1. Pre-compute M^3 = PROD(PROD(M,M), M)
Initialize flag←0

Step 2. for i = t -1 to 0 do
case ($e_{2i+1}e_{2i}$)
'00': if flag = 1 do
C ←PROD(C, C); C←PROD (C, C)
'01': if flag = 1 do
C ←PROD (C, C); C ←PROD (C, M)
else
C ←PROD (C, M); flag←1
'10': if flag = 1 do
C ←PROD (C, M); C←PROD (C, C)
else
C←PROD (C, M);
C←PROD (C, C); flag←1
'11': if flag = 1 do
C ←PROD (C, C); C ←PROD (C, M^3)
else
C←PROD (C, M^3); flag←1

Figure 2. Modular exponentiation algorithm for 8-bit IC cards

The modular multiplication algorithm is shown in Figure 3. Instead of using conventional method of multiplication following by division approach, a *lookahead determination* technique [16] was developed so as to reduce the RAM usage while it still provides excellent performance.

Given: A, B, N, $A = \sum_{j=0}^{n-1} A_j 2^{8j}$, $B = \sum_{j=0}^{n-1} B_j 2^{8j}$,
$N = \sum_{j=0}^{n-1} N_j 2^{8j}$ $0 \le A, B < N$, $\frac{2^8}{2} \le N_{n-1} \le 2^8 - 1$
Find: C = A * B mod N

Solution:

Step 1. C ← 0 (Byte length of C is n+2)
$i \leftarrow n$ -1

Step 2. C ← 2^8 C + A * B_i

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

Step 4. If q = 0, then goto Step 9
else C ← C - $2^8 * q$

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

Step 6. If q = 1, then
C ← C - 2^8 * N
else if q = 2, then C← C - 2^9 * N

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

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

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

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

Step 11. C ← C - q * N

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

Step 13. C ← C - N, Goto Step 12

Figure 3. Modular multiplication algorithm [16] for 8-bit IC cards

The proposed modular inverse algorithm, see Figure 4, is based on the extended Euclidean algorithm, but we adopt *approximation division* [10] instead of direct long division (see Step 2). Our experiences indicate that such an approach provides a 10 to 20 times performance improvement over the original extended Euclidean algorithm.

Given: A, N
Find: T such that $T * A \equiv 1 \bmod N$

Solution:

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 $Q \leftarrow 1$
$C(\text{new}) \leftarrow D(\text{old})$
$D(\text{new}) \leftarrow C(\text{old}) - Q * D(\text{old})$

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

Step 4. $X'(\text{new}) \leftarrow X(\text{old})$
$X(\text{new}) \leftarrow X'(\text{old}) + Q * X(\text{old})$
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

Figure 4. Modular inverse algorithm for 8-bit IC cards

5 Implementation of the Digital Signature Algorithms

The ESIGN and RSA algorithms were implemented on Hitachi's H8/3113 IC cards running at internal 5 MHz. Assembly codes for modular arithmetic algorithms were written and emulated using Hitachi's hardware development system. Performance of our modular multiplication and modular inverse algorithms are summarized in Table 1 and Table 2.

Table 1. Performance of modular multiplication on H8/300 8-bit IC cards running at 5MHz

modular multiplication	
576 bits	90 ms
768 bits	150 ms
1152 bits	360 ms

Table 2. Performance of modular inverse on H8/300 8-bit IC cards

modular inverse	
192 bits	120 ms
256 bits	200 ms
384 bits	470 ms

Assembly codes for ESIGN and RSA are then written. Total program size is about 3K bytes for each digital signature algorithm. To improve the performance of ESIGN, we revise the equation (2) into

$$T = W_0 \times E \times \left(E \times R^E\right)^{-1} \bmod P \qquad (4)$$

This way instead of performing $R^{E-1} \bmod P$, we perform $\left(R^E \bmod N\right) \bmod P$, where $R^E \bmod N$ is already computed at equation (1).

Besides, the implementation complexity (in terms of code size) of an algorithm is as important as the algorithm performance on the IC card environment. In other words, we have to carefully evaluate the tradeoff between program size and performance. Table 3 shows the performance of 1152-bit RSA with E=65537 and ESIGN with E=1024 on H8/300 CPU running at 5MHz. The total RAM usage in the implementation is 466 bytes.

Table 3. Performance of ESIGN and RSA on H8/300 8-bit IC cards

Digital Signature Algorithm	1152-bit RSA	1152-bit ESIGN
Pre-computation	No applicable	6.0 s
Signature Generation	155 s	0.15 s
Signature Verification	6.12 s	3.7 s

In the Figure 5, we gave the performance analysis of ESIGN on H8/300 IC card with different values of security parameter E, where *R^E mod N* denotes the operation of $R^E \bmod N$ in the equation (1), *R^E mod P* denotes the $\left(R^E \bmod N\right) \bmod P$ in the equation (4), INV is the modular inverse operation needed in the equation (3), and *Misc* denotes all other operations in ESIGN signature generation. By using fast modular inverse algorithm, the critical operation involved in the ESIGN algorithm is the operation of $R^E \bmod N$. Our results are somewhat different from the NESSIE's report on the 8051-based IC card Implementation of ESIGN [17].

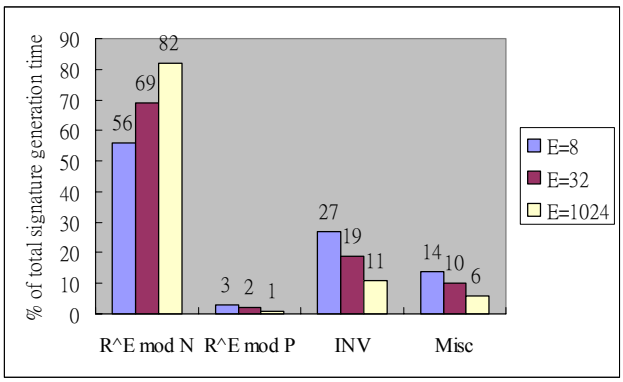

Figure 5. Performance analysis of ESIGN signature generation on H8/300 IC cards

For ESIGN, a pre-computation technique could be adopted to speed up signature generation. In this approach, we calculate two variables T_1 and T_2 in advance

that are dependent on random number R while is independent on message M.

$$T_1 = R^E \bmod N$$
$$T_2 = \left(E \times R^E\right)^{-1} \times R \mod P$$

Then when message M is available (for example, when a document needed to be signed), we generate signature S by the following equations,

$$W_0 = \left\lceil \frac{\left(f \| 0^{2k}\right) - T_1 \mod N}{PQ} \right\rceil$$
$$T = W_0 \times T_2 \mod P$$
$$S = R + T \times PQ$$

By pre-computation, ESIGN could generate 1152-bit signature in less than 0.5 second without using coprocessor in 8-bit IC cards.

6 Conclusions

Digital signature scheme is the core component of the public-key infrastructure (PKI) and is essential for the viability of e-government or e-commerce. In this paper, we have described our implementation efforts for 1152-bit ESIGN and RSA digital signature algorithms on general-purpose 8-bit IC cards. The results show that it takes less than 0.5 second to generate 1152-bit ESIGN signature on H8/3113 IC cards without a coprocessor while 1152-bit RSA signature takes more than 150 seconds.

We are currently in the process of improving our results and implementing ESIGN on the Infineon 8051-compatible IC cards. For ESIGN with large value of public exponent, say $1024 \leq E$, we might need to have a fast modular squaring algorithm and the use of the Chinese Remainder Theorem could give us further improvement.

References:

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

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

[3] N. Adam, et al, "E-Government: Human Centered Systems for Business Services," *The Proceedings of The First National Conference on Digital Government*, May 21-23, 2001.

[4] ISO 7498-2, "Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture," 1989.

[5] 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.

[6] W. Ford and M. S. Baum, *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption*, Prentice Hall, 1997.

[7] T. Okamoto, "A Fast Signature Scheme Based on Congruential Polynomial Operations", *IEEE Trans. Info. Theory*, Vol. 36, pp. 47-53, January 1990. See also http://info.isl.ntt.co.jp/esign/ or https://www.cosic.esat.kuleuven.ac.be/nessie/updatedPhase2Specs/esign/esignd-spec.pdf

[8] 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.

[9] *Hitachi Single-Chip Microcomputer H8/3113 Hardware Manual*, Hitachi Ltd., 1998.

[10] Technologies AG, *Security & Chip Card ICs SLE 66CX322P*, 2002. http://www.infineon.com/cmc_upload/documents/048/210/SPI_SLE66CX322P_1102.pdf

[11] D. E. Knuth, *The Art of Computer Programming - Seminumerical Algorithms*, Vol. 2, Section 4.3, 3rd edition, Addison-Wesley, 1998.

[12] National Institute of Standards and Technology, "Secure Hash Standard," *FIPS PUB 180-2*, August 2002. http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

[13] ISO/IEC 14888-3, Information technology - Security techniques - Digital Signatures with Appendix-Part 3: Certificate-Based Mechanisms, Dec. 1998.

[14] New European Schemes for Signatures, Integrity and Encryption (NESSIE), https://www.cosic.esat.kuleuven.ac.be/nessie/

[15] IEEE P1363a: Standard Specifications For Public Key Cryptography: Additional Techniques, http://grouper.ieee.org/groups/1363/P1363a/

[16] H. Morita and C. H. Yang, "A Modular-Multiplication Algorithm using Lookahead Determination," *IEICE Trans. Fundamentals,* Vol. E76-A, No. 1, January 1993, pp. 70-77.

[17] NESSIE, *Performance of Optimized Implementations of the NESSIE Primitives*, version 1.0, pp. 41-42, October 30, 2002. https://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/D21-v1.pdf