A Secure and Practical CRT-Based RSA to Resist Side Channel Attacks*

ChangKyun Kim[1], JaeCheol Ha[2**], Sung-Hyun Kim[3], Seokyu Kim[3], Sung-Ming Yen[4], and SangJae Moon[1]

[1] School of Electrical Engineering and Computer Science, Kyungpook National Univ., Daegu, 702-701, Korea
dreamps@m80.knu.ac.kr and sjmoon@knu.ac.kr

[2] Division of Information Science, Korea Nazarene Univ., Cheonan, Choongnam, 330-718, Korea
jcha@kornu.ac.kr

[3] System LSI Division, Samsung Electronics Co., Ltd., Korea
{teri_kim,ceokyu.kim}@samsung.com

[4] Dept of Computer Science and Information Engineering National Central Univ., Chung-Li, Taiwan 320, R.O.C.
yensm@csie.ncu.edu.tw

Abstract. A secure and practical CRT-based RSA signature scheme is proposed against side channel attacks, including power analysis attack, timing attack, and fault analysis attack. The performance advantage obtained over other existing countermeasures is demonstrated. To prevent from fault attack, the proposed countermeasure employs a fault diffusion concept which is to spread the fault into the correct term during the recombination process by using CRT. This new countermeasure is also secure against differential power attack by using the message random blinding technique on RSA with CRT.

Keywords: CRT, DPA, Fault attack, RSA, Side channel attack, Smart card.

1 Introduction

We focus our attention on the CRT-based RSA signature [7, 8]. Recently, this system may be vulnerable to fault analysis attacks [2, 4–6, 14] and the power analysis attack [1]. We introduce several attacks mainly based on two reasonable assumptions. Firstly, an adversary can insert a random fault during the computation of a signature and get a fault output. He tries to find a secret prime p or q in RSA with CRT. Secondly, he can input the chosen messages directly to

* This research has been supported by University IT Research Center Project.
** The second author was also supported in part by Korea Nazarene University research fund.

its system for power attacks. These assumptions have been widely used in many attacks for several cryptosystems.

To prevent from side-channel attacks including the fault analysis attack and the timing attack, some countermeasures by using fault detection or fault tolerance have been reported in many papers [11–13]. However, they suffer from some disadvantages such as computational load, production of undetectable error, or less compatibleness with existing systems. Moreover, Boer *et al.* reported that they can be broken by a differential power analysis attack [1].

In this paper, the main purpose is to present a countermeasure against the existing two fault attacks. The core idea is that a fault induced during a term computation processing spread the fault over another term in recombination using CRT. To prevent from DPA attack, we employ the message random blinding technique on this system. Also, the proposed countermeasure is a more efficient and robust method than existing countermeasures, and is strong against some side channel attacks.

2 Preliminary

2.1 The CRT-Based RSA System

Consider the RSA system [9] without the Chinese remainder theorem (CRT). Let $N = p \cdot q$ be a product of two prime integers each $|N|/2$ bits long. To sign a given message m using RSA system computes $S = m_d \bmod N$, where d is a secret key of signer. To succeed in detecting the secret key d from public information, an attacker tries to find p and q by factoring N in prime factors.

The main computational operation of signing using RSA is the modular exponentiation of a message, m. The RSA without CRT scheme has more computational loads than other signature schemes, DSS and ECDSA. So, the CRT algorithm is used to compute more effectively the signature $S = m^d \bmod N$. The RSA with CRT algorithm was proposed to speed up the original RSA signature or decryption computation [7, 8]. In the RSA with CRT, they first computed $S_p = m^{d_p} \bmod p$ and $S_q = m^{d_q} \bmod q$, where $d_p = d \bmod (p-1)$ and $d_q = d \bmod (q-1)$. Then the signature, S, can be computed by the following recombination algorithm which is often called Gauss's algorithm.

$$S = (S_p \cdot q \cdot (q^{-1} \bmod p) + S_q \cdot p \cdot (p^{-1} \bmod q)) \bmod N$$

where both $q^{-1} \bmod p$ and $p^{-1} \bmod q$ can be pre-computed to reduce the computational loads.

The computing time of S_p requires 1/8 the time of computing S. Thus, computing both S_p and S_q takes 1/4 the time to compute S directly. The RSA with CRT is about four times faster than direct exponentiation algorithm. This is why RSA with CRT is widely adopted as an implementation method in processors with limited computing power such as smart cards.

2.2 Vulnerability to Fault Attack

The RSA system with CRT which has been naively implemented is vulnerable to fault attacks. A fault cryptanalysis which has lately attracted attention is a method of intentionally causing faults to occur in hardware or software during operation of a smart card or processor and detecting particular secret information using faulty output.

The fault attack-I proposed by Boneh *et al.* is based on a theory that when either S_p or S_q is not a correct value during signature generation based on RSA with CRT, the N can be factored in prime factors using two signatures (one is correct signature and the other is faulty signature) with respect to a same message [2]. It is assumed that a fault occurs in S_p during computation of the signature, which results in faulty signature $\widehat{S}$ due to a faulty $\widehat{S}_p$ and a correct S_q.

$$\widehat{S} = (\widehat{S}_p \cdot q \cdot (q^{-1} \bmod p) + S_q \cdot p \cdot (p^{-1} \bmod q)) \bmod N$$

Then, the computation of

$$gcd((S - \widehat{S}), N) = q$$

will give the secret prime q, so N is easily factored.

The attack II proposed by Lenstra allows fault cryptanalysis to be accomplished using only one faulty signature [6]. It is also assumed that a fault occurs under the same fault as above, that is,

$$S_p \neq \widehat{S}_p \bmod p.$$

Then, this fault enables to factor N by computing

$$gcd((\widehat{S}^e - m) \bmod N, N) = q$$

where e is the public exponent used to verify the signature S. Consequently, $(\widehat{S}^e - m)$ is not a multiple of p but is a multiple of q. Accordingly, the secret prime q is detected. The secret prime p can be also detected from a faulty signature on a faulty $\widehat{S}_q$ and a correct S_q.

2.3 Shamir's Countermeasure and Its Improvement

In order to protect against such fault attacks, the following various algorithms have been developed. Since these fault attacks are based on the assumption that a fault has occurred in S_p or S_q, it is checked to find whether a fault has occurred during generation of a signature by computing S_p or S_q two times. However, this approach requires a large amount of computation. Moreover, in the case of a system having a permanent fault, there is no way of verifying whether a fault has occurred during generation of a signature.

In another approach, an original message $m = S^e \bmod N$ is recovered through signature verification with respect to a signature value, and it is checked to find

whether a fault has occurred. In this way, if a large number, public exponent e or modulus N, is used for verification, then a large amount of computation is required. Therefore, it increases the computational load when compared to the previous countermeasure and the naively implemented scheme.

Shamir proposed a simple countermeasure against these fault attacks [11, 12]. According to this method, a random prime r will be selected. After computing $p' = p \cdot r$ and $q' = q \cdot r$, the following two values are computed

$$S_p{}' = m^{d_p{}'} \bmod p'$$
$$S_q{}' = m^{d_q{}'} \bmod q'$$

where $d_p{}' = d \bmod (p-1)\cdot(r-1)$ and $d_q{}' = d \bmod (q-1)\cdot(r-1)$. Then we check whether $S_p{}' \equiv S_q{}' \bmod r$, and if the checking is correct then it is determined that no fault has occurred in generation of a signature. In this case a signature is generated by computing

$$S = (S_p \cdot q \cdot (q^{-1} \bmod p) + S_q \cdot p \cdot (p^{-1} \bmod q)) \bmod N$$

where $S_p = S_p{}' \bmod p$ and $S_q = S_q{}' \bmod q$.

However, the method proposed by Shamir has the following problems. First, a probability of a fault that cannot be theoretically detected is $1/r$. Here, if a large r is selected, the probability of a fault that cannot be detected can be reduced. However, operating efficiency decreases because a modular computation on large modulus must be performed. In contrast, if a small r is selected, operating efficiency increases, but a probability of a faulty occurrence that cannot be detected increases.

Second, since the size of modulus is extended from $|p|$ or $|q|$ to $|p \cdot r|$ or $|q \cdot r|$, respectively, this method is not compatible with existing systems such as smart cards or general purpose processors. Moreover, two exponents, $d_p{}'$ and $d_q{}'$, increased by the random number r increases the computational load by about r bits compared with the naively implemented scheme.

3 A New Countermeasure Against Side Channel Attacks

This section presents the secure and practical CRT-based RSA signature scheme against side channel attacks, including the timing attack, differential power analysis (DPA), simple power analysis (SPA), and fault attack. This scheme employs three techniques to protect from side channel attacks. First, to prevent from SPA, this protocol employs a dummy operation. Second, to prevent from a timing attack and DPA, a randomization of the message and key is employed. Finally, to prevent from fault attack, this protocol proposes the fault diffusion concept which is to diffuse fault of an abnormal computed term into a normal computed term.

3.1 The Proposed RSA with CRT

To solve the above problems of the previous countermeasure, the first target is to present a digital signature method which does not require any additional parameters, thereby allowing the method to be compatible with existing systems, and providing protection against fault cryptanalysis.

We consider that existing fault cryptanalysis is based on the fact that a fault occurs in either S_p or S_q. Our main idea is to extend a fault throughout generation of a signature even if the fault occurs in only one of S_p and S_q to prohibit an attacker from deriving a formula which can attack secret primes. For example, when a fault occurs only in the S_p, the fault is induced into the other terms to protect the secret prime q. We present a novel countermeasure to resist against side channel attacks.

Input: $m,\ d,\ p,\ q,\ N$
Output: S.

1. Compute $S_p = m^{d_p} \bmod p$ and $S_q = m^{d_q} \bmod q$ with $Exp(\ )$, where $d_p = d \bmod p-1$ and $d_q = d \bmod q-1$.
2. Compute $T_p = (m - {S_p}^{e_p}) \bmod p$ and $T_q = (m - {S_q}^{e_q}) \bmod q$, where $e_p = e \bmod p-1$ and $e_q = e \bmod q-1$.
3. Compute $T = T_p \oplus T_q$ using XOR operation $\oplus$.
4. Compute the signature S using enhanced Gauss's algorithm. $S = (S_p \cdot (q \oplus T) \cdot (q^{-1} \bmod p) + S_q \cdot (p \oplus T) \cdot (p^{-1} \bmod q)) \bmod N$.
5. Check both $S \equiv S_p \bmod p$ and $S \equiv S_q \bmod q$. If these are true then send out the computed signature S.

Fig. 1. Proposed RSA with CRT immune to side channel attacks

In the above computation of the figure 1, when S_p and S_q are normally generated, as $T_p = (m - {S_p}^{e_p}) \bmod p = 0$ and $T_q = (m - {S_q}^{e_q}) \bmod q = 0$, a correct signature is generated. However, when a faulty $\widehat{S}_p$ and a normal S_q are generated, a checking value $T_p = (m - {S_p}^{e_p}) \bmod p$ is not zero. This value T_p includes at least one non-zero bit. As a result, $(S_p \cdot (q^{-1} \bmod p)$ is multiplied not by q but by another value $(q \oplus T)$ and $S_q \cdot (p^{-1} \bmod q)$ is not by p but by $(p \oplus T)$. Therefore the fault occurring in S_p is spread to the term including S_q. In the case where the attacker uses a fault attack, it is assumed that a correct signature S and a fault signature $\widehat{S}$ are as follows.

$$S = (S_p \cdot q \cdot (q^{-1} \bmod p) + S_q \cdot p \cdot (p^{-1} \bmod q)) \bmod N$$

$$\widehat{S} = (\widehat{S}_p \cdot (q \oplus \widehat{T}) \cdot (q^{-1} \bmod p) + S_q \cdot (p \oplus \widehat{T}) \cdot (p^{-1} \bmod q)) \bmod N$$

which results in faulty signature $\widehat{S}$ due to a faulty $\widehat{S}_p$ and a correct S_q. Then, a formula used by the attacker is not valid, as shown.

$$S - \widehat{S} \neq u_q(S_p - \widehat{S}_p)$$

where we can assume that u_q is a multiple of q. Therefore the attacker cannot obtain the secret prime q using $gcd((S - \widehat{S}), N)$ in fault attack-I.

In case fault attack-II, the attacker calculate whether $(\widehat{S}^e - m) \bmod N$ is multiple of q. Applying fault $\widehat{S}_p$ or $\widehat{S}_q$ will give a faulty signature $\widehat{S}$, so $(\widehat{S}^e - m)$ which becomes neither a multiple of p nor a multiple of q. As a result, the attacker cannot eventually take any secret prime by computing $gcd((S - \widehat{S}) \bmod N, N)$.

Finally, we suppose that a computational fault and memory access fault is induced when computing signature S in step 4 by using the enhanced Gauss's algorithm [7]. Even though both S_p and S_q are correct in step 1, which generates $T = 0$, when an attacker inserts a fault such as S_p, S_q, p, q, $q^{-1} \bmod p$ or $p^{-1} \bmod q$ during recombining in step 4 then computes a faulty signature. This fault signature can give a secret key to an attacker, so the checking in step 5 can detect a fault generated during recombination computation.

In the above computation, however, if S_p and S_q in step 1 are generally computed without countermeasure against power attacks, then an adversary can obtain the secret prime by using a DPA attack that uses byte-wise hypotheses on the remainder after the modular reduction with one of the primes [1]. This attack uses special data called Modular reduction on equidistant data (MRED). Therefore, the message m should be blinded by a random number r. The detailed process of this algorithm is described in the figure 2.

Input: $m,\ d_p,\ p$ (or $d_q,\ q$)
Output: S_p (or S_q).

1. Randomly choose a number r.
2. $C = m \cdot r \bmod p$
3. $Temp[0] = r^{-1} \bmod p$ and $Temp[1] = m \cdot r^{-1} \bmod p$
4. for $i = n - 1$ downto 0{
4.1 $C = C^2 \bmod p$
4.2 $C = C \cdot Temp[d_{p_i}] \bmod p$
}
5. $S_p = C \cdot Temp[0] \bmod p$
6. Return(S_p)

Fig. 2. Exponentiation algorithm immune to DPA, SPA, Timing attack: $Exp(\ )$

Intermediate results during the exponentiation algorithm, $Exp(\ )$, is always a value multiplied by r, and moreover, $Temp[1]$ substitutively employed message m which is also blinded by r^{-1}. Therefore, MRED attack and timing attack don't work when $m' = m \cdot r^{-1} \bmod p$ or $m' = m \cdot r^{-1} \bmod q$.

In the exponentiation of $S_p = m^{d_p} \bmod p$, since this processing is depend on the secret value d_p, an attacker can find the secret value d_p from a measured power consumption signal by using SPA. But figure 2 shows a countermeasure to SPA attack, where the instructions conducted during a cryptographic algorithm

do not depend on the data being processed. It is similar to Coron's simple SPA countermeasure [3].

3.2 Consideration on permanent fault attack

In the permanent attack in the figure 1, we assume that some parameters are permanently corrupted by the attacker. Firstly, we consider that p is damaged and it becomes fault value $\widehat{p}$. In this case the signature $\widehat{S}$ is computed as follow.

$$\widehat{S} = (S_p \cdot (q \oplus T) \cdot (q^{-1} \bmod p) + S_q \cdot (\widehat{p} \oplus T) \cdot (p^{-1} \bmod q)) \bmod (\widehat{p} \cdot q)$$

Therefore, it can be verified that the step 5 well works, that is, $\widehat{S} \neq S_p \bmod \widehat{p}$ because $q \cdot (q^{-1} \bmod p) \neq 1 \bmod \widehat{p}$. This is similar with the case of permanent fault q.

Secondly, we assume that a permanent fault induced on
$q^{-1} \bmod p$. The fault signature $\widehat{S}$ is computed as follow.

$$\widehat{S} = (S_p \cdot (q \oplus T) \cdot (\widehat{q^{-1} \bmod p}) + S_q \cdot (p \oplus T) \cdot (p^{-1} \bmod q)) \bmod (p \cdot q)$$

It can be detected in step 5, that is, $\widehat{S} \neq S_p \bmod p$ because $q \cdot (\widehat{q^{-1} \bmod p}) \neq 1 \bmod p$.

Thirdly, a permanent fault on d will cause faults both $\widehat{S}_p$ and $\widehat{S}_q$. Therefore fault attack is in vain. However, a permanent fault on d_p will cause a fault $\widehat{S}_p$ but a correct S_q. By this fault, temporary value T_p is not zero, so the signature is represented as follow.

$$\widehat{S} = (\widehat{S}_p \cdot (q \oplus \widehat{T}) \cdot (q^{-1} \bmod p) + S_q \cdot (p \oplus \widehat{T}) \cdot (p^{-1} \bmod q)) \bmod (p \cdot q)$$

This means that fault $\widehat{S}_p$ is spread over two terms and detected in $\widehat{S} \neq \widehat{S}_p \bmod p$ because $(p \oplus \widehat{T}) \cdot (p^{-1} \bmod q)) \neq 0 \bmod p$.

3.3 Performance of the Proposed RSA with CRT

In the proposed RSA signature scheme in the figure 1, new system parameters are not needed. Furthermore, the computation in step 1 and 2 can be computed in parallel if two processors are used. In real implementation, some temporary registers are needed to store e_p, e_q, T_p, T_q, and T.

We compare the performance as computational time compared with the conventional RSA system with CRT. Additional computational loads are occurred in step 2, 3, and 5. However computational time for step 3 and 5 is minor due to its simple operation. In step 2, we need two modular exponentiations using public exponent e. If this public key is very small, then additional computational time is negligible. However, the worse performance of our method is the case using a long integer having similar length with d. It can be cleared that the computational time will take about twice as much compared to the original RSA with CRT. Even in this worst case, our method computes about two times faster than direct RSA system without CRT. Furthermore, this protocol can avoid the disadvantages of producing an undetectable error such as Shamir's method.

4 Concluding Remarks

We proposed a secure and practical implementation of CRT-based RSA signature to resist side channel attacks. The basic idea of this scheme is the message random blinding technique and the fault diffusion concept. Also, this scheme does not need any additional system parameter. Additional computational overhead necessary to prevent from side channel attacks is negligible when compared with the conventional RSA with CRT. Furthermore, in order to speed up the modular exponentiation algorithm, combining either the m-ary or the sliding window techniques to $Exp(\ )$ can be possible.

References

1. Bert den Boer, K. Lemke, and G. Wieke, "A DPA attack against the modular reduction within a CRT implementation of RSA," *Proc. of Cryptographic Hardware and Embedded Systems*, LNCS 2523, pp. 228–243, Springer-Verlag, 2003.
2. D. Boneh, R.A. DeMillo, and R.J. Liption, "One the important of checking cryptographic protocols for faults," *Advances in Cryptology – EUROCRYPT '97*, LNCS 1233, pp. 37–51, Springer-Verlag, 1997.
3. J. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," *Proc. of Cryptographic Hardware and Embedded Systems*, LNCS 1717, pp. 292–302, Springer-Verlag, 1999.
4. M. Joye, A.K. Lenstra, and J.-J. Quisquater, "Chinese remaindering based cryptosystems in the presence of faults," *Journal of Cryptology*, vol. 12, no. 4, pp. 241–245, 1999.
5. M. Joye, J.-J. Quisquater, F. Bao, and R.H. Deng, "RSA-type signatures in the presence of transient faults," *Proc. of Cryptography and Coding*, LNCS 1355, pp. 155–160, Springer-Verlag, 1997.
6. A.K. Lenstra, "Memo on RSA signature generation in the presence of faults," September 1996.
7. A.J. Menezes, P.C.van Oorchot, and S.A. Vanstone. *Handbook of applied cryptography*. CRC Press, 1997.
8. J.-J. Quisquater and C. Couvreur, "Fast decipherment algorithm for RSA public key cryptosystem," *Electronics Letters*, vol. 18, no. 21, pp. 905–907, 1982.
9. R.L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystem," *Commun. of ACM*, vol. 21, no. 2, pp. 120–126, 1978.
10. W. Schindler, "A timing attack against RSA with the Chinese remainder theorem," *Proc. of Cryptographic Hardware and Embedded Systems*, LNCS 1717, pp. 292–302, Springer-Verlag, 1999.
11. A. Shamir, "How to check modular exponentiation," Presented at the rump session of *EUROCRYPT '97*, Konstanz, Germany, May 1997.
12. A. Shamir, "Method and apparatus for protecting public key schemes from timing and fault attacks," United States Patent 5991415, November 23, 1999.
13. S.M. Yen, S.J. Kim, S.G. Lim, and S.J. Moon, "RSA speedup with residue number system immune against hardware fault cryptanalysis," *Proc. of Information Security and Cryptology*, LNCS 2288, pp. 397–413, Springer-Verlag, 2002.
14. S.M Yen, S.J. Moon, and J.C. Ha, "Permanent fault attack on the parameters of RSA with CRT," *Proc. of Information Security and Privacy – ACISP '03*, LNCS 2727, pp. 285–296, Springer-Verlag, 2003.