情報保護學會論文誌
第19卷 第6號, 2009. 12

연산자 조작 공격과 피연산자 조작 공격에 대한 기존 CRT-RSA Scheme의 안전성 분석*

허 순 행,† 이 형 섭, 이 현 승, 최 동 현, 원 동 호, 김 승 주‡
성균관대학교 정보보호연구소

The Security Analysis of Previous CRT-RSA Scheme on Modified Opcode and Operand Attack*

Soonhaeng Hur,† Hyungsub Lee, Hyunseung Rhee, Donghyun Choi,
Dongho Won, Seungjoo Kim‡
Information Security Group, Sungkyunkwan University

요 약

CRT-RSA의 사용이 대중화됨에 따라, CRT-RSA에 대한 보안 또한 중요 이슈가 되었다. 1996년, Bellcore 연구원들에 의해 CRT-RSA가 오류 주입 공격에 취약하다고 밝혀진 이래로, 많은 대응책들이 제안되었다. 첫 번째 대응책은 1999년 Shamir에 의해 제안되었으며, Shamir의 대응책은 오류 검사 기법에 기반을 두고 있다. Shamir의 대응책이 소개된 이후, 오류 검사 기법을 사용하는 많은 대응책들이 제안되었다. 그러나 Shamir의 대응책은 2001년 Joey 등에 의하여 피연산자 조작 공격에 취약함이 밝혀졌으며, 오류 검사 기법 또한 2003년 Yen 등에 의하여 연산자 조작 공격에 취약하다고 알려졌다. 이에 Yen 등은 오류 검사 기법을 사용하지 않고 오류 확산 기법을 사용하여 새로운 대응책을 제안하였으나, Yen 등이 제안한 대응책 또한 2007년에 Yen과 Kim에 의하여 안전하지 않음이 밝혀졌다. 최근에는 Kim 등이 Yen 등의 대응책을 보완한 새로운 대응책을 제안하였으며, Ha 등 또한 오류 확산 기법을 사용한 대응책을 제안하였다. 그러나 Kim 등과 Ha 등이 제안한 대응책들을 포함한 기존 대응책들은 연산자 조작 공격에 대해서는 안전성이 증명되지 않았기 때문에 본 논문에서는 피연산자 조작 공격은 물론, 연산자 조작 공격도 고려하여 지금까지 제안된 대응책들의 안전성을 분석할 것이다.

ABSTRACT

As the use of RSA based on chinese remainder theorem(CRT-RSA) is being generalized, the security of CRT-RSA has been important. Since Bellcore researchers introduced the fault attacks on CRT-RSA, various countermeasures have been proposed. In 1999, Shamir firstly proposed a countermeasure using checking procedure. After Shamir's countermeasure was introduced, various countermeasures based on checking procedure have been proposed. However, Shamir's countermeasure was known to be vulnerable to the modified operand attack by Joey et al. in 2001, and the checking procedure was known to be vulnerable to the modified opcode attack by Yen et al. in 2003. Yen et al. proposed a new countermeasure without checking procedure, but their countermeasure was known to be also vulnerable to the modified operand attack by Yen and Kim in 2007. In this paper, we point out that pre, but countermeasures were vulnerable to the modified operand attack or the modified opcode attack.

Keywords: CRT-RSA, opcode, operand, fault attack

접수일(2009년 6월 18일), 게재확정일(2009년 10월 30일)
* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음. (NIPA -2009-(C1090-0902-0016))

† 주저자, shhur@security.re.kr
‡ 교신저자, skim@security.re.kr

I. 서 론

1996년 Kocher에 의해 부 채널 공격(side-channel attack)이 소개되기 전까지, 암호 알고리즘의 안전성은 이론상으로만 논의되고 있었다. 그러나 실제 임베디드(embeded) 장치에 구현된 암호 알고리즘이 부 채널 공격에 취약함이 알려진 이래로, 오류 주입 공격(fault attacks), 전력 분석 공격(power analysis attacks) 등의 여러 부 채널 공격 방법들이 등장하였다. 그 중에서 1997년에 Bellcore 연구원들에 의하여 소개된 오류 주입 공격은 강력한 부 채널 공격 방법 중 하나이다[1-3].

오류 주입 공격의 기본 아이디어는 임베디드 장치의 암호화 연산 과정에 오류를 주입하여 잘못된 연산 결과를 얻어낸 후, 그 값을 이용하여 비밀 키(p 또는 q)를 추측하는 방법이다. 이러한 오류 주입 공격은 연산자(opcode) 조작 공격과 피연산자(operand) 조작 공격의 두 가지 종류로 나누어진다. 암호 알고리즘이 임베디드 장치에 탑재될 때에는 어셈블리(assembly) 코드 형태로 변환되어 탑재되며, 어셈블리 코드는 연산자와 피연산자로 구성되어있기 때문이다. 이러한 어셈블리 코드의 연산자 부분에 오류를 발생시켜, 해당 어셈블리 명령어(instruction)가 수행되지 않고, 다음 명령어가 바로 수행되도록 만드는 공격이 연산자 조작 공격이다[4]. 그리고 피연산자 조작 공격은 어셈블리 코드의 피연산자 부분에 오류를 발생시켜, 해당 어셈블리 명령어의 수행 결과가 정상적인 값이 아닌 잘못된 값이 되도록 만드는 공격 방법이다[5].

본 논문에서는 현재 가장 널리 사용되고 있는 중국인의 잉여 정리를 사용한 RSA(CRT based RSA, CRT-RSA)에 대한 오류 주입 공격을 고려할 것이다. 지금까지 알려진 CRT-RSA에 대한 오류 주입 공격의 대응책으로는 Shamir, Joye 등, Aumuller 등, Yen 등, Blomer 등, Ciet 등, Giraud, Kim 등, Boscher 등이 제안한 대응책들이 있으며, 최근에는 Kim 등과 Ha 등이 새로운 대응책을 제안하였다.

따라서 본 논문의 2장에서 연산자 조작 공격과 피연산자 조작 공격에 대하여 기존 대응책들의 안전성을 간단히 분석하고, 3장에서는 최근에 제안된 Kim 등과 Ha 등이 제안한 대응책의 안전성을 분석할 것이며, 4장에서 결론을 맺을 것이다.

II. 연산자 조작 공격과 피연산자 조작 공격에 대한 기존 대응책들의 안전성 분석

CRT-RSA의 오류 주입 공격에 대한 첫 번째 대응책은 1999년, Shamir에 의해 발표되었다[6]. Shamir는 서명 값을 생성한 뒤, 오류 발생 여부를 검사하는 오류 검사 기법을 사용하였다. 그러나 Shamir의 대응책은 2001년 Joey 등에 의하여 피연산자 조작 공격에 취약함이 밝혀졌다[7]. 이에 Joye 등은 Shamir의 대응책을 개선한 새로운 오류 검사 기법을 제안하였으며, 한 편 Aumuller 등 또한 Joye 등이 제안한 것과는 다른 방법을 사용하여 대응책을 제안하였다[8]. 그러나 Joey 등과 Aumuller 등이 제안한 대응책들은 2003년 Yen 등에 의해 모두 연산자 조작 공격에 취약함이 발표되었다[9]. 그리고 Yen 등은 오류 검사 기법 대신 오류 확산 기법을 사용한 대응책을 제안하였으나, Yen 등이 제안한 대응책 또한 피연산자 조작 공격에 대하여 취약점을 가지고 있었다[10]. 2003년에는 Blomer 등 또한 Yen 등이 제안한 방식과는 다른 방법을 사용하여 대응책을 제안하였으나[11], Blomer 등의 대응책은 Wagner에 의해 피연산자 조작 공격에 취약함이 발표되었다[12]. 2005년에는 Ciet 등이 오류 확산 기법을 사용한 대응책을[13], Giraud가 Montgomery Ladomr 알고리즘을 사용한 대응책을 각각 제안하였다[14]. 그러나 Ciet 등이 제안한 대응책은 Kim과 Quisquater에 의해 연산자 조작 공격에 취약함이 알려졌으며[15], 또한 Ciet 등의 대응책은 Berzati 등에 의해 피연산자 조작 공격에도 취약함이 발표되었다[16]. 그리고 Giraud가 제안한 대응책은 Yen 등에 의하여 연산자 조작 공격에 취약하다고 알려졌다[17]. 최근에는 Kim과 Quisquater가 Giraud와 Ciet의 대응책을 수정하여 새로운 대응책을 제안하였지만[15], 수정된 Giraud의 대응책은 연산자 조작 공격과 피연산자 조작 공격에 취약하고, 수정된 Ciet의 대응책은 연산자 조작 공격에 취약함이 Ha 등에 의해 발표되었다[18]. 마지막으로 2007년에는 Boscher 등이 오류 주입 공격과 전력 분석 공격을 동시에 고려하여 새로운 대응책을 제안하였으나[19], Boscher 등에 제안한 대응책은 Kwon 등에 의하여 연산자 조작 공격에 취약함이 알려졌다[20].

[표 1]과 같이 Yen 등이 제안한 대응책을 제외한 대부분의 대응책들은 연산자 조작 공격에 취약하다. 대부분의 대응책들이 오류 검사 기법을 사용하고 있거

[표 1] 기존 대응책들의 안전성 분석

기존 대응책	안전성	
	피연산자 조작 공격	연산자 조작 공격
Shamir. (1999)	X	X
Joye et al. (2001)	X	X
Aumuller et al. (2003)	X	X
Yen et al. (2003)	X	O
Blomer et al. (2003)	X	X
Ciet et al. (2005)	X	X
Giraud (2005)	O	X
Kim et al.:modify Ciet (2007)	O	X
Kim et al.:modify Giraud (2007)	X	X
Boscher et al. (2007)	X	X

나, 피연산자 조작 공격에 대해서만 고려했기 때문이다. 반면 Yen 등의 대응책은 오류 검사 기법을 사용하지 않고 오류 확산 기법을 사용함으로써 연산자 조작 공격에 안전함을 보였지만, [10]에서 제안한 하드웨어 오류 공격(hardware fault attack)에 취약하다는 단점이 있었다. 하드웨어 오류 공격은 공격자가 능동적으로 공격대상인 암호 장치에 의도적인 오류를 주입하는 공격으로써, 공격자가 물리적인 방법으로 하드웨어 장치에 의도적인 오류를 주입할 수 있다는 가정 사항을 포함한다. 이에 따라 최근 Kim 등은 하드웨어 오류 공격에 취약하다는 단점을 보완한 대응책을 제안하였으며, Ha 등 또한 오류 확산 기법을 사용한 새로운 대응책을 제안하였다. Kim 등과 Ha 등이 제안한 대응책은 다음과 같다.

2.1 Kim 등이 제안한 대응책

Kim 등은 k_q 연산 과정에서 피연산자 조작 공격이 발생했을 때를 대비하여, $\tilde{m}=m\wedge(S_q^{e_r} \bmod q)++k_q \cdot q$ 또는 $\hat{m}=((S_p^{e_r} \bmod p)+k_p \cdot p)\wedge((S_q^{e_r} \bmod q)+k_q \cdot q)$와 같이 기존 $\hat{m}$, $\tilde{m}$ 연산에 원본 메시지 m과의 AND 연산을 추가하였다. 이는 AND 연산의 특징 중 $m\wedge m=m$과 $m\wedge x=R$(R은 랜덤 값)을 이용한 것이다. Kim 등이 제안한 대응책은 [표 2, 3]과 같다.

Kim 등이 제안한 대응책에 따르면, 만약 Step 1~2 과정에서 피연산자 조작 공격이 발생하여 S_p, S_q, k_p, k_q와 같은 파라미터 값이 랜덤 값이 되는 경우, Step 3의 AND 연산 결과가 랜덤 값이 되므로, $\hat{m}$ 또는 $\tilde{m}$의 값도 랜덤 값이 된다. 따라서 $S=CRT(S_p,S_q)\cdot \tilde{m}^r \bmod N$ 또는 $S=CRT(S_p,S_q)\cdot \hat{m}^r \bmod N$ 의 연산 결과가 랜덤 값이 되기 때문에 Kim 등은 제안한 대응책이 피연산자 조작 공격에 안전하다고 주장했다[21].

[표 2] Kim 등이 제안한 CRT-1 protocol

Input : message $m\in Z_n$ Output : signature $S=m^d \bmod N$
1. Compute $k_p=\lfloor m/p \rfloor$, $k_q=\lfloor m/q \rfloor$ 2. Compute $S_p=m^{d_r \bmod (p-1)} \bmod p$ $S_q=\hat{m}^{d_r \bmod (q-1)} \bmod q$ where $\hat{m}=((S_p^{e_r} \bmod p)+k_p \cdot p) \bmod q$ 3. Compute $S=CRT(S_p,S_q)\cdot \tilde{m}^r \bmod N$ where $\tilde{m}=m\wedge(S_q^{e_r} \bmod q+k_q \cdot q)$ 4. Output S

[표 3] Kim 등이 제안한 CRT-2 protocol

Input : message $m\in Z_n$ Output : signature $S=m^d \bmod N$
1. Compute $k_p=\lfloor m/p \rfloor$, $k_q=\lfloor m/q \rfloor$ 2. Compute $S_p=m^{d_r \bmod (p-1)} \bmod p$ $S_q=m^{d_r \bmod (q-1)} \bmod q$ 3. Compute $S=CRT(S_p,S_q)\cdot \hat{m}^r \bmod N$ where, $\hat{m}=((S_p^{e_r} \bmod p)+k_p \cdot p)\wedge((S_q^{e_r} \bmod q)+k_q \cdot q)$ 4. Output S

2.2 Ha 등이 제안한 대응책

Ha 등은 랜덤 수 r_1, r_2와 $e_p=d_p^{-1} \bmod r_1$, $e_q=d_q^{-1} \bmod r_2$를 사용하여 중간 값 T를 계산하고, 중간 값 T와 랜덤 수 R을 이용하여 CRT 결합 연산에 오류를 확산시키는 기법을 사용하였다. Ha 등이 제안한 대응책은 [표 4]와 같다.

Ha 등이 제안한 대응책에 따르면, 정상 상태일 경우에는 $T=0$, $c=1$이 되므로 Step 6에서 정상 서명 값이 출력된다. 그러나 만약 Step 1~4 과정에서 피연산자 조작 공격이 발생한다면, 중간 값 T는 0이 아닌 랜덤 값이 되므로, Step 4의 결과 값과 c 값이 랜덤 값이

[표 4] Ha 등이 제안한 대응책

Input : message $m, d_p, d_q, e_p, e_q, p^{-1}, q^{-1}, r_1, r_2$ Output : signature $S = m^d \bmod N$
1. Compute $S_{pr} = m^{d_p} \bmod pr_1,\ S_{qr} = m^{d_q} \bmod qr_2$ 2. Compute $T_p = (m - S_{pr}^{e_p}) \bmod r_1,\ T_q = (m - S_{qr}^{e_q}) \bmod r_2$ 3. Compute $T = (T_p \oplus T_q),\ T = T \bullet (R \oplus T)$ where R is a random number 4. Compute $S = (S_p \bullet (q \oplus T) \bullet q^{-1}) + (S_q \bullet (p \oplus T) \bullet p^{-1}) + R \bmod N$ where $S_p = S_{pr} \bmod p,\ S_q = S_{qr} \bmod q$ 5. Compute $c = ((S - S_{pr} + R)) \bmod p \oplus (S - (S_{qr} + R)) \bmod q) R + 1$ 6. Compute $S = (S - R)^c \bmod N$ 7. Output S

된다. 또한 Step 5에서 피연산자 조작 공격이 발생할 경우에도 $c \neq 1$이 된다. 따라서 Step 1~5 과정에서 피연산자 조작 공격이 발생한다면 $c \neq 1$이 되므로, Step 6에서 오류 서명 값이 생성된다. 그러므로 Ha 등은 제안한 대응책이 피연산자 조작 공격에 안전하다고 주장했다[18].

III. 연산자 조작 공격에 대한 Kim 등의 기법과 Ha 등의 기법의 취약점 분석

2장에서 살펴본 바와 같이 Kim 등과 Ha 등이 제안한 대응책들은 모두 피연산자 조작 공격에는 안전하다. 그러나 위 대응책들은 연산자 조작 공격에 대해서는 전혀 고려되지 않았기 때문에 다음과 같은 취약점이 존재한다.

3.1 Kim 등이 제안한 기법의 취약점 분석

Kim 등은 피연산자 조작 공격에 대응하기 위하여, 기존 $\hat{m}$, $\tilde{m}$ 연산에 원본 메시지 m과의 AND 연산을 추가하였으나, 위와 같이 AND 연산을 단독으로 사용할 경우 다음과 같이 연산자 조작 공격에 취약하게 된다.

Kim 등이 제안한 $\tilde{m} = m \wedge (S_q^{e_r} \bmod q) + k_q \bullet q$ 연산을 어셈블리 코드로 간단히 나타내면 다음과 같다.

Step 1. MOV eax, DWORD PTR [$(S_q^{e_r} \bmod q) + k_q \bullet q$)
Step 2. MOV ebx, DWORD PTR [m]
Step 3. AND eax, ebx
Step 4. MOV DWORD PTR [$\tilde{m}$], eax

위 어셈블리 코드에 따르면 Step 1에서 eax에 $(S_q^{e_r} \bmod q) + k_q \bullet q)$의 결과 값이 저장되고 Step 2에서 ebx에 m 값이 저장된다. 그리고 Step 3에서 eax 값과 ebx 값의 AND 연산이 수행되고 그 결과 값은 다시 eax에 저장된다. 만약 Step 3에 해당하는 어셈블리 명령어의 연산자 부분이 손상되어 동작하지 않는다면, eax에는 Step 1의 수행 결과인 $(S_q^{e_r} \bmod q) + k_q \bullet q)$ 값이 그대로 저장된 채, Step 4로 넘어가게 된다. 즉, Step 1~4 과정의 결과 값은 $\tilde{m} = (S_q^{e_r} \bmod q) + k_q \bullet q$가 되므로, 기존 Yen 등이 제안한 대응책과 같은 취약점을 갖게 된다. 다시 말해서, Kim 등이 제안한 기법 역시 하드웨어 오류 공격에 취약하게 되는 것이다.

또한 Kim 등에 제안한 기법의 $\hat{m} = ((S_p^{e_r} \bmod p) + k_p \bullet p) \wedge ((S_q^{e_r} \bmod q) + k_q \bullet q)$ 연산도 연산자 조작 공격에 취약하다. $\hat{m} = ((S_p^{e_r} \bmod p) + k_p \bullet p) \wedge ((S_q^{e_r} \bmod q) + k_q \bullet q)$ 연산을 간단한 어셈블리 코드로 나타내면 다음과 같다.

Step 1. MOV eax, DWORD PTR [$(S_p^{e_r} \bmod p) + k_p \bullet p$]
Step 2. MOV ebx, DWORD PTR [$(S_q^{e_r} \bmod q) + k_q$] $\bullet q$
Step 3. AND eax, ebx
Step 4. MOV DWORD PTR [$\hat{m}$], eax

위 어셈블리 코드는 Step 1에 eax에 $(S_p^{e_r} \bmod p) + k_p \bullet p$의 결과 값을, Step 2에 ebx에 $(S_q^{e_r} \bmod q) + k_q \bullet q$의 결과 값을 각각 저장 한 뒤, Step 3에서 AND 연산을 수행하고 그 결과 값을 eax에 저장하는 코드이다. 따라서 Step 3의 AND 명령어의 연산자가 손상되었을 경우, eax에는 Step 1에서 저장된 $(S_p^{e_r} \bmod p) + k_p \bullet p$이 그대로 남아있는 상태에서 Step 4로 넘어가게 된다. 따라서 $\hat{m}$에는 $(S_p^{e_r} \bmod p) + k_p \bullet p$ 값이 저장되므로, 해당 기법 역시 하드웨어 오류 공격에 취약하게 된다.

3.2 Ha 등이 제안한 기법의 취약점 분석

Ha 등이 제안한 대응책의 기본 구조는 다음과 같다[18].

Step 1. S_p, S_{pr}과 S_q, S_{qr} 계산
Step 2. CRT 재결합 알고리즘을 사용하여 서명 값 S를 계산
Step 3. 오류 검사 기법 (where $S_p = S_{pr} \bmod p$, $S_q = S_{qr} \bmod q$)을 이용하여 오류 발생 유무 확인
Step 4. 서명 값 출력

Ha 등이 제안한 대응책뿐만 아니라, [표 1]에 나타난 대다수의 대응책들도 위와 비슷한 4단계 구조를 가지고 있다[15]. 만약 Step 3과 같은 오류 검사 기법에서 연산자 조작 공격이 발생한다면, Step 1 또는 2에서 오류가 발생했다 할지라도 Step 3이 수행되지 않고 Step 4가 수행되기 때문에, 공격자는 Step 1 또는 2에 피연산자 조작 공격을 수행하거나, Step 3에 연산자 조작 공격을 수행하여 Step 4에서 잘못된 서명 값 S'을 얻을 수 있고, $GCD(S'^e - m, N)$을 계산하여 비밀 값 q를 계산할 수 있다.

IV. 결 론

CRT-RSA에 대한 오류 주입 공격에 대응하기 위하여 많은 대응책들이 제안되었지만, 지금까지 알려진 대부분의 대응책들은 오류 검사 기법을 사용하고 있거나 피연산자 조작 공격만을 고려하였기 때문에 연산자 조작 공격에 취약하다. 본 논문에서는 최근에 발표된 Kim 등과 Ha 등의 대응책을 포함하여 현재까지 발표된 대응책을 분석하였으며, 이러한 대응책들은 대부분 오류 확산을 위해 특정 연산 과정을 포함하고 있었다. 그러나 오류 확산을 위한 연산 과정이 피연산자 조작 공격이나 연산자 조작 공격에 취약함이 드러났다. 따라서 안전한 CRT-RSA를 위해서는 피연산자 조작 공격과 연산자 조작 공격에 모두 안전한 새로운 대응책 개발이 시급하다. 이에 대한 한 가지 방법으로 오류 확산 연산 과정을 어셈블리 코드로 보았을 때, 상위 Step에서 오류가 발생할 경우 하위 Step의 수행 결과에 관계없이 임의의 값이 나오도록 하는 방법이 있을 수 있다. 본 논문은 피연산자 조작 공격과 연산자 조작 공격에 모두 안전한 대응책을 개발하는데 기여할 수 있을 것으로 기대된다.

참 고 문 헌

[1] Bellcore Press Release, "New threat model breaks cypto codes," 1996.

[2] D. Boneh, R.A. DeMillo, and R.J. Lipton, "On the important of checking cryptographic protocols for faultsm," EUROCRYPT'97, LNCS 1233, pp. 37-51, 1997.

[3] A. Lenstra, "Memo on RSA signature generation in the presence of faults," manuscript, Sep. 1996.

[4] L.G. Pierson, P.L. Campbell, J.M. Eldridge, P.J. Robertson, T.D. Tarman, and E.L. Witzke, "Secure computing using cryptographic assurance of execution correctness," 38th Annual 2004 International Carnahan Conference, pp. 239-246, Oct. 2004.

[5] S. Singh and M. Hill, "Fault-Tolerant Method and Means for managing Access to an Initial Program Load Stored in Read-Only Memory or the Like," US Paten 5832005, Nov. 1998.

[6] A. Shamir, "How to Check Modular Exponentiation," presented at the rump session of EUROCRYPT'97, Konstanz, 11-15th, May 1997.

[7] M. Joye, P. Pailler, and S.M. Yen, "Secure evaluation of modular functions," International Workshop on Cryptology and Network Security 2001, pp. 227-229, Sep. 2001.

[8] C. Aumuller, P. Bier, W. Fischer, P. Hofreiter, and J.P. Seifert, "Fault atacks on RSA with CRT: Concrete results and practical countermeasures," Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2002, LNCS 2523, pp. 260-275, 2003.

[9] S.M. Yen, S.J. Kim, S.G. Lim, and S.J. Moon, "RSA speedup with Chinese remainder theorem immune against hardware fault cryptanalysis," IEEE Trans. On Computers Special issue on CHES, vol. 52, no. 4, pp. 461-472, Apr. 2003.

[10] S.M. Yen, D.R. Kim, and S.J. Moon, "Cryptanalysis of Two Protocols for RSA with CRT Based on Fault Infection," FDTC 2006, LNCS 4236, pp. 53-61, 2006.

[11] J. Blomer, M. Otto, and J.P. Seifert, "A new

CRT-RSA algorithm secure against Bellcore attacks," 10th ACM Conference on Computer and Communications Security, pp. 311-320, Oct. 2003.

[12] D. Wagner, "Cryptanalysis of a provably secure CRT-RSA algorithm," 11th ACM Conference on Computers and Communications Security, pp. 92-97, Oct. 2004.

[13] M. Ciet and M. Joye, "Practical fault countermeasures for Chinese remaindering based RSA," Fault Diagnosis and Tolerance in Cryptography-FDTC'05, pp. 124-131, Sep. 2005.

[14] C. Giraud, "Fault resistant RSA implementation," Fault Diagnosis and Tolerance in Cryptography-FDTC 2005, pp. 142-151, Nov. 2005.

[15] C. Kim and J.J. Quisquaterm, "Fault Attacks for CRT based RSA: new Attacks, new Results and new Countermeasures," Workshop in Information Security Theory and Practices 2007: Smart Cards, Mobile and Ubiquitous Computing Systems-WISTP 2007, LNCS 4462, pp. 215-228, 2007.

[16] A. Berzati, C. Canovas, and L. Goubin, "(In)Security against fault injection attacks for CRT-RSA implementations," 5th workshop on fault diagnosis and tolerance in cryptography, pp. 101-107, Aug. 2008.

[17] S.M. Yen, L.C. Ko, S.J. Moon, and J.C. Ha, "Relative Doubling attack against Montgomery Ladder," International Conference on Information Security and Cyptography ICISC'05, LNCS 3935, pp. 117-128, 2006.

[18] J.C. Ha, J.H. Park, and S.J. Moon, "A Countermeasure Resistant to Fault Attacks on CRT-RSA using Fault Injective Method," Journal of Korea Institute of Information Security & Cryptology, vol. 18, no. 2, pp. 75-83, Apr. 2008.

[19] A. Boscher, R. Naciri, and E. Prouff, "CRT-RSA Algorithm Protected Against Fault Attacks," Workshop in Information Security Theory and practices WISTP'07, LNCS 4462, pp. 237-252, 2007.

[20] E.J. Kwon, J.H. Shin, and P.J. Lee, "Fault Attack on Secure Exponentiation algorithm Against SPA-FA," Conference on Information Security and Cryptology in Summer - CISC-S 2007, pp. 237-252, June 2007.

[21] S.K. Kim, T.H. Kim, D.H. Han, Y.H. Park, and S.H. Hong, "Secure RSA with CRT Protected Against Fault Attacks without using Checking Procedure," Journal of Korea Institute of Information Security & Cryptology, vol. 18, no. 4, pp. 17-22, Aug. 2008.