SPA와 FA에 안전한 CRT를 사용하는 RSA 알고리즘*

김 성 경[1], 김 태 현[1], 한 동 국[2], 홍 석 희[1]
[1]고려대학교 정보경영공학전문대학원, [2]한국전자통신연구원
likesk@cist.korea.ac.kr

Secure CRT-RSA against SPA and FA

Sung-Kyoung Kim[1], HeeSeok Kim[1], Tae Hyun Kim[1], Dong-Guk Han[2],
Jeong Choon Ryoo[1], Jongin Lim[1]
[1]Graduate School of Information Management and Security, Korea University,
[2]Electronics and Telecommunications Research Institute

요약

본 논문에서는 단순전력 분석(SPA)과 오류주입공격(FA)에 안전한 중국인의 나머지 정리를 이용한 RSA 암호 시스템(CRT-RSA)에 대하여 논한다. CRT-RSA를 이용한 서명 알고리즘은 스마트카드와 같은 내장형 장치(embedded device)에서 널리 사용된다. 하지만 이러한 장치들은 전력분석 공격과 오류주입 공격에 취약하다. 2005년 Giraud가 처음으로 단순전력분석과 오류주입공격에 모두 안전한 대응 방법을 제안하였다. 본 논문에서는 Giraud의 대응 방법에 대한 다른 공격방법을 소개하고, 제시한 공격 방법에도 안전한 대응 방법을 제안한다. 본 논문에서 제안하는 대응 방법은 세 개의 메모리와 덧셈과 뺄셈연산을 추가적으로 요구한다. 추가적으로 요구되는 연산량은 모듈러 지수승 연산에 필요한 연산량에 비교하면 크게 고려하지 않아도 될 연산량이다. 그러므로 본 논문에서 제안하는 대응 방법은 내장형 장치와 같은 환경에서 안전하고 효율적으로 이용될 수 있다.

1. 서론

부채널 공격[19]이 소개 된 후 내장형 장치(embedded device)의 암호 알고리즘에 대한 다양한 공격 방법이 소개되었다. 수동적인 공격 방법으로 분류되는 공격은 TA(시간 공격, timing attack), SPA(단순 전력분석, simple power analysis), DPA(차분 전력 분석, differential power analysis), EMA(전자파 분석, electromagnetic analysis)이라고 한다[1,10,11,13].

능동적인 공격은 변형된 외부 클럭을 주입하거나, 온도를 변화시키거나 장치에 X-ray와 같은 레이저를 이용하여 공격하며[4], 이와 같이 능동적인 공격 방법을 오류주입 공격(fault attack, FA)이라고 한다. 1996년 Bellcore가 중국인의 나머지 정리를 이용한 RSA 암호 시스템(CRT-RSA)[9]에서의 오류주입 공격 방법을 제시한 후, 최근 DES[6], RSA[9], ElGamal[3], ECC[5], AES[8] 등 다양한 암호 알고리즘에서 공격되었다. CRT-RSA 암호시스템에 대한 오류주입 공격 방법에 대한 대응책은 처음 Shamir가 1997년에 CRT 연산 마지막 단계에 비교 연산을 추가함으로 오류 주입 여부를 확인하였고[20,21], 그 방법을 응용하여 다양한 대응 방법이 제안되었다[2,14,15,22].

최근 2005년 Giraud가 처음으로 전력분석 공격 방법 중 하나인 단순전력 분석공격과 오류주입 공격에 안전한 방법을 제안하였다[14,15]. Giraud의 방법에서의 지수승 알고리즘은 단순전력 분석와 오류주입공격에 안전한 Montgomery Ladder 방법을 이용하여 계산되고, 개인키만을 이용하여 오류주입을 확인하게 된다.

본 논문에서는 단순전력 분석와 오류주입 공격에 안전하다고 소개된 Giraud에 대한 다른 공격 방법을 제시하고 소개한 공격 방법에도 안전한 대응 방법을 제안한다. 제시한 공격 방법은 모듈러 곱셈 과정에서의 입력 값에 대한 공격이며, 이 공격에 안전하게 하기 위해 오류를 확산하는 방법을 이용한다. 제안하는 대응 방법은 기존의 Giraud의 장점을 모두 가지면서 덧셈과 뺄셈 연산과 같은 적은 추가 연산을 요구한다.

2. CRT 기반의 RSA 암호 알고리즘과 오류주입 공격

RSA 암호 시스템에서는 개인키를 이용하여 연산하는 서명 생성 과정의 속도 향상을 위해 중국인의 나머지 정리(Chinese Remainder Theorem, CRT)를 많이 이용한다. CRT 기반의 RSA 암호 시스템에서 사용되는 파라미터들은 RSA 모듈러 값인 $n = p \cdot q$와 공개키 e 그리고 개인키 d이다. 그리고 $z = CRT(x, y) \bmod n$는 $z \equiv x \bmod p$, $z \equiv x \bmod q$을 만족하는 유일한 값($z (\in Z/nZ)$)이다. CRT 기반의 RSA 암호 시스템에서 $m^d \bmod n$의 계산은

$$s_p \equiv m^{d \bmod (p-1)} \bmod p,\ s_q \equiv m^{d \bmod (q-1)} \bmod q$$

을 계산한 후 $s = CRT(s_p, s_q)$연산을 시행한다.

그러나 [9] 논문에 의해서 오류주입 공격을 사용하면 CRT 기반의 RSA 암호시스템은 안전하지 않다는 것이 소개되었다. 만약 s_p(또는 s_q) 계산에서 오류가 발생하고, s_q(또는 s_p)는 오류가 발생하지 않는 값이라고 할 경우 다음과 같은 연산을 통하여 비밀 정보인 q(또는 p)

* "본 연구의 일부는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (IITA-2006-(C1090-0603-0025))

의 값을 공격자가 알 수 있다.

$$q = \gcd(s'^{e} - m, n) \quad \text{if } s_p \text{ error}$$
$$p = \gcd(s'^{e} - m, n) \quad \text{if } s_q \text{ error}$$

Shamir가 Eurocrypt'97[20]와 1999년 등록 된 특허[21]에서 CRT 기반의 RSA 암호 시스템에 대한 오류주입 공격에 대한 대응 방법을 제안하였다. 제안하는 방법은 32 비트 정도의 랜덤 한 값 r을 선택한 후

$$s_p \equiv m^{d \bmod (p-1)(r-1)} \bmod pr$$
$$s_q \equiv m^{d \bmod (q-1)(r-1)} \bmod qr$$

을 계산하고, CRT 연산 전에 $s_p \equiv s_q \bmod r$를 확인한다. 만약 s_p(또는 s_q)에서 오류가 발생하였을 경우 $s_p \equiv s_q \bmod r$의 연산 단계를 통과하지 못하기 때문에 오류 주입 여부를 확인할 수 있다. 그러나 Shamir의 방법에는 여러 가지 문제점이 제안되었고 그 후로 많은 대응 방법이 소개되었다. 최근 2005년에 Ciet와 Joye, 그리고 Giraud가 새로운 대응 방법을 제시하였다[14,17]. Ciet와 Joye[17]은 Shamir의 대응 방법을 일반화하였고, Giraud[14,15]은 Montgomery Ladder[18]을 이용하여 오류주입 공격에 안전한 방법을 제안하였다. Giraud의 방법은 전력 분석 공격 중 하나인 단순전력분석공격에도 안전한 방법이다. 본 논문에서는 Giraud의 대응 방법에 대한 다른 공격 방법을 소개하고, 본 논문에서 소개하는 공격 방법에도 안전한 대응 방법을 제안한다.

3. Giraud의 대응 방법

Giraud [14,15]의 방법은 Joye와 Yen이 제안하였던 단순전력 분석 대응 방법의 Montgomery Ladder[18]모듈러 지수승 알고리즘[18]의 결과 값인 (m^{a-1}, m^{a})을 이용하였다. Algorithm 1이 [14,15] 에서 제안한 알고리즘이다. Algorithm 1을 이용하여 계산 되는 Giraud의 CRT-RSA 방법은 Algorithm 2와 같다. Giraud가 제안한 CRT-RSA 방법은 공개키 e를 요구하지 않으므로 Java Card의 공개키 환경에서처럼 오직 개인키 d만 요구하는 경우에도 사용가능하다. 또한 공개키 e를 이용하여 CRT-RSA 연산을 사용 할 경우 공개키의 길이가 커질수록 속도도 늦어지게 되는 단점을 가지고 있다.

Algorithm1. Giraud의 모듈러 지수승 알고리즘[14,15]

INPUT $m, d, oddN, k$

OUTPUT $(m^{d-1} \bmod kN, m^{d} \bmod kN)$

1. $a_0 \leftarrow m$
2. $a_1 \leftarrow a_0^2 \bmod kN$
3. for i from $n-2$ to 1 do
   3.1. $a_{\overline{d_i}} \leftarrow a_{\overline{d_i}} \cdot a_{d_i} \bmod kN$
   3.2. $a_{d_i} \leftarrow a_{d_i}^2 \bmod kN$
4. $a_1 \leftarrow a_1 \cdot a_0 \bmod kN$
5. $a_1 \leftarrow a_0^2 \bmod kN$
6. if (Loop Counter i not disturbed) & (Exponent d not disturbed) then return (a_0, a_1) else return(error)

Giraud가 제안하는 CRT-RSA 방법은 두 번의 CRT연산을 수행

Algorithm2. Giraud의 대응방법[14,15]

INPUT $m, p, q, d_p, d_q, a(=p^{-1} \bmod q)$;

OUTPUT $m^{d} \bmod n$;

1. random integer k
2. $(s_p', s_p) \leftarrow Algorithm1(m, d_p, p, k)$
3. $(s_q', s_q) \leftarrow Algorithm1(m, d_q, q, k)$
4. $s' \leftarrow CRT_{blinded}(s_p', s_q')$
5. $s \leftarrow CRT_{blinded}(s_p, s_q)$
6. $s' \leftarrow m \cdot s' \bmod n$
7. if $(s' = s)$ and (parameters p, q, a not modified) then return s else return (error)

한다. 입력 값 m, p, q, d_p, d_q, a가 주어졌을 때, 먼저 랜덤 한 k를 선택한다. 단계 2에서 (m, d_p, p, k)를 사용하여 (s_p', s_p)를 계산하고 $s_p' = m^{d_p - 1} \bmod k \cdot p$, $s_p = m^{d_p} \bmod k \cdot p$ 값을 가진다. 단계 3의 (s_q', s_q)의 값도 동일한 방법으로 계산된다. 단계 4, 5에서는 두 번의 CRT 연산을 통하여 $s' = m^{d-1} \bmod n$, $s = m^{d} \bmod n$값을 계산하고, 단계 6에서 s' 값에 m을 곱하여 s와 동일한 값을 만들어 단계 7에서 s값과 비교한다. 이 때,

$$CRT_{blinded}(s_p, s_q) = (((s_q - s_p) \bmod k \cdot q)) \cdot a \bmod k \cdot q)) \cdot p + s_p \bmod p \cdot q$$

이다. 만약 그 전 단계에서 오류가 주입되었을 경우 단계 7에서의 $s' = s$를 통과하지 못하게 된다. 또한, 계산에 사용되는 각 변수들에 대한 변경 역시 단계 7에서 확인하게 된다. [14,15] 논문에서 제시 된 안전성은 다음과 같다. 본 논문에서는 오류가 주입되어 변경된 값을 $\hat{}$ (hat)을 사용하여 표현한다. 즉, x에 오류가 주입된 경우 $\hat{x}$로 표시한다.

1. (s_q', s_q) 계산중에 m에 오류가 주입되었을 경우. $(s_p', \hat{s}_q')$를 이용하여 s'를 계산하고, $(s_p, \hat{s}_q)$를 이용하여 s를 계산한다. $\hat{s}_q'$, $\hat{s}_q$의 값은 다음과 같다.

$$\hat{s}_q' = \hat{m}^{d_p - 1} \bmod k \cdot q, \quad \hat{s}_q = \hat{m}^{d_p} \bmod k \cdot q$$

따라서 단계 7에서 $s' = s$ 비교단계는

$$\begin{aligned} \hat{s} - m \cdot \hat{s}' &= ((\hat{s}_p - s_p) \cdot a \bmod q) \cdot p + s_p \\ &- m \cdot (((\hat{s}_q - s'_p) \cdot a \bmod q) \cdot p + s'_p) \bmod n \\ &= ((\hat{m}^{d_q} - m \cdot \mu \cdot q - m^{d_p} + m \cdot \lambda \cdot p) \cdot a + a_0 \cdot q) \cdot p \\ &+ m^{d_q} - m \cdot \lambda \cdot p \\ &- ((m \cdot \hat{m}^{d_p - 1} - m \cdot \mu \cdot q - m^{d_p} + m \cdot \lambda \cdot p) \cdot a \\ &+ m \cdot a_1 \cdot q) \cdot p \\ &= a \cdot p \cdot \hat{m}^{d_q - 1} \cdot (\hat{m} - m) \bmod n \end{aligned}$$

이고, λ와 μ는 유일한 정수 값이다. 이 때, $\hat{m} - m$은 q로 나눠지지 않고 '0'의 값도 아니기 때문에 단계 7을 통과하지 못한다. 그러므로 공격자는 비밀 정보를 알아낼 수 없다.

2. s' (또는 s) 계산 과정에서 오류가 주입되었을 경우. s (또는 s')의 계산을 위해 CRT 연산 시 오류가 주입 되었을 경우에는 $s = m \cdot s' \bmod n$의 관계가 성립되지 않으므로 단계 7을 통과하지

못하게 된다.

3. (s'_p, s_p) (또는 (s'_q, s_q)) 계산 과정에 오류가 주입되었을 경우. 만약 $(\hat{s}'_p, \hat{s}_p)$의 경우 단계 7은

$$\begin{aligned}\hat{s} - m\cdot\hat{s}' &= ((s_q - \hat{s}_p)\cdot a \bmod q)\cdot p + s_p \\ &- m\cdot(((s'_q - \hat{s}'_p)\cdot a \bmod q)\cdot p + \hat{s}'_p) \bmod n \\ &= ((m^{d_q} - m\cdot\mu\cdot q - \hat{s}_p)\cdot a + \zeta_0\cdot q)\cdot p + \hat{s}_p \\ &- (((m^{d_p} - m\cdot\mu\cdot q - m\cdot\hat{s}'_p)\cdot a + m\cdot\zeta_1\cdot q)\cdot p \\ &- m\cdot\hat{s}'_p) \bmod (p\cdot q), \quad (\zeta_0, \zeta_1 \in Z) \\ &= (m\cdot\hat{s}'_p - \hat{s}_p)\cdot(a\cdot p - 1) \bmod n\end{aligned}$$

이다. $a = p^{-1} \bmod q$ 이므로 $a\cdot p = 1 \bmod p$이고, $a\cdot p = 1 + \zeta\cdot p$이므로

$\hat{s} - m\cdot\hat{s}' = (m\cdot\hat{s}'_p - \hat{s}_p)\cdot\zeta\cdot q \bmod n$ 을 만족한다. $m\cdot\hat{s}'_p \neq \hat{s}_p \bmod p$이기 때문에 '0'의 값도 아니고 p로 나눠지지 않는다.

4. 제안하는 공격 방법

본 장에서는 3장에서 소개하였던 Giraud의 대응 방법에 대한 새로운 오류주입 공격 방법을 제안한다. [15] 논문의 [TABLE 1.]에서는 d_p와 d_q에 오류가 주입되었을 경우에도 안전하다고 제시하였다. 하지만 단계 2 (또는 단계 3)에서 d_p (또는 d_q)에 오류가 주입된 경우에는 안전하지 않다. d_p에 오류가 일시적으로 주입되었을 경우를 예로 들면 다음과 같다.

1. 단계 2에서 (s_p', s_p)를 구하는 과정에서 d_p에 오류가 주입되었을 경우를 고려한다면, 단계 2의 값은 $(\hat{s}_p', \hat{s}_p) = (m^{\hat{d}_p - 1} \bmod k\cdot p, m^{\hat{d}_p} \bmod k\cdot p)$이다.

2. 단계 3은 $(s_q', s_q) = (m^{d_q - 1} \bmod k\cdot q, m^{d_q} \bmod k\cdot q)$ 이다.

3. 단계 4에서 $\hat{s}' = CRT(\hat{s}_p', s_q')$,
   단계 5에서 $\hat{s} = CRT(\hat{s}_p, s_q)$로 계산된다.

4. 알고리즘의 확인 과정은 $m\hat{s}' = \hat{s} \bmod n$이다.

$$m\hat{s}' = m_p\hat{s}_p' \bmod p, \quad m\hat{s}' = m_q s_q' \bmod q$$

이고, 다음을 만족한다.

$$\hat{s} = \hat{s}_p \bmod p, \quad \hat{s} = s_q \bmod q.$$

$m_p\hat{s}_p' \equiv \hat{s}_p \bmod p$이고 $m_q s_q' \equiv s_q \bmod q$이기 때문에 $m\hat{s}' \equiv \hat{s} \bmod n$을 만족하게 된다. 그리고 p, q, a 역시 변경되지 않았기 때문에 단계 7을 통과하게 된다. 동일한 방법으로 d_q에서 오류가 발생하더라도 단계 7을 통과하게 된다. 즉, 두 번의 CRT 연산 과정에서 같은 오류의 값으로 계산되기 때문에 비교 과정에서 a, p, q에서의 오류주입만 발생하지 않았다면 공격이 가능하게 된다. 따라서 Giraud가 제안한 대응 방법은 d_p, d_q에서의 오류 주입에 대해서 공격이 가능하게 된다.

5. 제안하는 알고리즘

Algorithm3. 제안하는 대응 방법

INPUT

$m, p, q, T(= d_p + d_q), d_p, d_q, a(= p^{-1} \bmod q)$;

OUTPUT $m^d \bmod n$;

1. random integer k
2. $(s_p', s_p) \leftarrow Algorithm1(m, d_p', p, k)$, $d_p' = T - d_q$
3. $(s_q', s_q) \leftarrow Algorithm1(m, d_q', q, k)$, $d_q' = T - d_p'$
4. $s' \leftarrow CRT_{blinded}(s_p', s_q')$
5. $s \leftarrow CRT_{blinded}(s_p, s_q)$
6. $s' \leftarrow (m\cdot s' \bmod n) \wedge (\sum_{i=\alpha}^{\rho-1} 2^i + (\overline{T} \oplus (d_p + d_q')))$
7. if ($s' = s$) and (parameters p, q, a not modified) then return s else return (error)

본 장에서는 4장에서 제안하였던 공격방법에도 안전하게 구현될 수 있는 방법을 제안한다. 공격이 가능했던 이유가 단계 2 또는 3에서의 계산 시 오류가 발생하였더라도 (s_p', s_q') (또는 (s_p, s_q))의 두 값 사이에 연결성이 유지되어 있기 때문이다. 따라서 본 논문에서 제안하는 대응 방법은 (s_p', s_q') (또는 (s_p, s_q))을 계산할 때의 오류를 확인하는 과정을 추가하여 단계 7의 $s' = s$를 통과하지 못하게 한다. 제안하는 대응 방법은 Algorithm 3과 같다. 기존의 방법과 달리 입력 값이 $T(= d_p + d_q), d_p, d_q$이다. 단계 2에서의 입력 값은 $m, d_p'(= T - d_p), p, k$가 되고 단계 3의 입력 값은 $m, d_q'(= T - d_p'), q, k$이다. 단계 6에서 $m\cdot s' \bmod n$의 값과 $(\sum_{i=\alpha}^{\rho-1} 2^i + (\overline{T} \oplus (d_p + d_q')))$ 값을 AND 연산하게 된다. 이때 $\overline{T}$의 값은 T의 보수 값이고, α는 T의 비트 길이, ρ은 모듈러 n의 비트길이이다. 오류가 주입되지 않았을 경우는 T 값과 $d_p + d_q'$의 값이 동일하기 때문에 ⊕(XOR) 연산의 결과 $2^\alpha - 1$이고 $\sum_{i=\alpha}^{\rho-1} 2^i + 2^\alpha - 1 = 2^\rho - 2^\alpha + 2^\alpha - 1 = 2^\rho - 1$이다. (다시 말해, $\sum_{i=\alpha}^{\rho-1} 2^i$는 AND 연산을 위한 비트 길이를 맞추는 패딩 연산이고, 결과 값은 $\underbrace{111\cdots1}_{\rho}$이다.) 따라서 $m\cdot s' \bmod n$ 값과 AND 연산을 하게 되면 기존의 $m\cdot s' \bmod n$값이 된다. 그러므로 단계 7을 통과하게 되어서 올바른 서명 값을 출력하게 된다. 오류가 발생 될 수 있는 경우는 입력 값과 중간 계산 결과 값이다. 즉, $m, p, q, T, d_p, d_q, a, s_p', s_p, s_q', s_q, s', s, k$이다. 오류가 발생하였을 경우를 살펴보면 아래와 같다.

1. 단계 2의 계산 시 d_q에 오류가 발생하였을 경우. $\hat{d}_q$의 경우 d_p'의 값도 오류가 발생하게 되고, 단계 3에서의 입력 값 d_p'에 오류가 주입되었기 때문에 s_p, s_p', s_q, s_q' 모두 오류가 주입된 값이므로 공격자가 올바르지 않는 값을 이용하여 p또는 q를 얻지 못한다.

2. 단계 2, 단계 3에서의 d'_p 메모리에 오류가 발생하였을 경우. 이와 같은 경우는 1번과 동일하게 s_p, s_p', s_q, s_q' 모두 오류가 주입된 값이므로 공격자가 올바르지 않는 값을 이용하여 p또는 q를 얻지 못한다.

3. 단계 3의 계산 시 d'_p에 오류가 발생하였을 경우. 첫 번째 오

류와는 달리 d'_p에 오류가 발생한 경우에는 $\hat{d}_q'$이 되며, 첫 번째 경우와는 달리 오류 확산이 일어나지 않는다. 따라서 별도로 확인하는 과정이 필요하다. 그 과정이 단계 6에서 $(\sum_{i=a}^{b-1} 2^i + (\overline{T} \oplus (d_p + d'_q)))$을 계산하는 것이다. $(\overline{T} \oplus (d_p + \hat{d}_q'))$의 값의 모든 비트가 '1'이 아니기 때문에 단계 6의 결과 값인 s'이 s값과 달라지기 때문에 단계 7을 통과하지 못한다.

4. 일시적인 T의 오류. 단계 2에서의 일시적으로 T에 오류가 발생하였을 경우에는 첫 번째 분석과 동일하게 오류 확산이 일어나기 때문에 공격이 불가능하다. 또한 단계 3에서의 일시적 오류인 경우도 두 번째 분석과 동일하게 단계 7을 통과하지 못한다.

5. 지속적인 T의 오류. 지속적인 T에 오류가 발생하였을 경우 d'_p는 오류가 주입된 값이 되지만 d'_q의 값은 올바른 값이 된다. 따라서 T의 오류를 확인하는 과정이 없다면 공격자는 공격을 성공하게 된다. T의 오류 확인 과정도 단계 6에서 $(\overline{T} \oplus (d_p + d'_q))$을 이용한다. $(d_p + d'_q)$는 기존의 오류가 주입되지 않은 T값과 동일하지만, 오류가 주입 된 $\overline{T}$와 달라지므로 단계 7을 통과하지 못하게 된다.

다른 오류가 주입되었을 경우는 기존의 Giraud의 방법의 분석과 동일하게 된다. 따라서 본 논문에서 제안하는 대응 방법은 단계 2, 단계 3에서의 입력 값에 대한 일시적인 오류까지 안전한 방법이다. 본 대응방법은 기존의 방법보다 세 개의 레지스터 d_p', d_q', T를 추가 사용하고, 단계2, 3에서의 두 번의 뺄셈연산, 단계 6에서 두 번의 덧셈연산, AND연산과 XOR연산만을 추가하여 공격에 대응하게 된다.

6. 결론

본 논문에서는 오류주입 공격과 단순전력 분석 공격에 안전하다고 제안된 Giraud의 대응 방법에 대한 새로운 공격 방법과 제안하는 공격 방법에 안전한 대응 방법을 제안하였다. 제안한 방법은 Giraud의 대응 방법에 대한 장점을 모두 취하며, 세 개의 레지스터를 더 요구하며 추가적인 연산으로 덧셈, 뺄셈 연산, 그리고 XOR, AND 연산을 사용한다. 모듈러 지수승 연산에 필요한 연산량에 비교하면 크게 고려하지 않아도 되는 추가 연산량이다. 따라서 제안하는 방법은 계산 능력과 메모리가 제안된 모든 환경에서 안전하게 이용될 수 있다.

References

[1]. D. Agrawal, B. Archambeault, J. Rao, and P. Rohatgi. The EM side-channel(s). In Cryptographic Hardware and Embedded Systems - CHES'02, volume 2523 of Lecture Notes in Computer Science, pages 29 - 45. Springer, 2002.

[2]. C. Aumüller, P. Bier, W. Fischer, P. Hofreiter, and J.-P. Seifert, "Fault attacks on RSA with CRT: Concrete results and practical countermeasures," Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2002, LNCS 2523, pp. 260 - 75, Springer-Verlag, 2003.

[3]. F. Bao, R. Deng, Y. Han, A. Jeng, A. Narasimhalu, and T. Ngair. Breaking public key cryptosystems on tamper resistant devices in the presence of transient faults. In Proceedings of the 5th Workshop on Secure Protocols, volume 1361 of Lecture Notes in Computer Science, pages 115 - 124. Springer, 1997.

[4]. H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan. The sorcerer's apprentice guide to fault attacks. In Fault Diagnosis and Tolerance in Cryptography in association with DSN 2004 - The International Conference on Dependable Systems and Networks, pages 330 - 342, 2004.

[5]. I. Biehl, B. Meyer, and V. Muller. Differential fault attacks on elliptic curve cryptosystems. In Advances in Cryptology - CRYPTO 2000, volume 1880 of Lecture Notes in Computer Science, pages 131 - 146. Springer, 2000.

[6] E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. In Proceedings of the 17th annual international cryptology conference on advances in cryptology, volume 1294 of Lecture Notes in Computer Science, pages 513 - 525. Springer, 1997.

[7]. J. BlÄomer and J.-P. Seifert. Fault based cryptanalysis of the advanced encryption standard (AES). In Financial Cryptography - FC'03, volume 2742 of Lecture Notes in Computer Science, pages 162 - 181. Springer, 2003.

[8] B. Boer, K. Lemke, and G. Wicke. A DPA attack against the modular reduction within a CRT implementation of RSA. In Cryptographic Hardware and Embedded Systems - CHES'02, volume 2523 of Lecture Notes in Computer Science, pages 228 - 243. Springer, 2003.

[9]. D. Boneh, R. DeMillo, and R. Lipton. On the importance of checking cryptographic protocols for faluts. EUROCRYPT'97, volume 1223 of LNCS, Springer, 1997.

[10]. E. Brier, C. Clavier, and F. Olivier. Correlation power analysis with a leakage model. In Cryptographic Hardware and Embedded Systems - CHES'04, volume 3156 of Lecture Notes in Computer Science, pages 16 - 29. Springer, 2004.

[11] D. Brumley and D. Boneh. Remote timing attacks are practical. In Proceedings of the 12th Usenix Security Symposium, pages 1 - 14, 2003.

[12]. J.-F. Dhem, F. Koeune, P.-A. Leroux, P. Mestr, J.-J. Quisquater, and J.-L. Willems. A practical implementation of the timing attack. In Smart Card Research and Advanced Applications - CARDIS'98, volume 1820 of Lecture Notes in Computer Science, pages 167 - 182. Springer, 1998.

[13]. K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis: Concrete results. volume 2162 of Lecture Notes in Computer Science, pages 251 - 261. Springer, 2001.

[14]. C. Giraud. Fault resistant RSA implementation. In Fault Diagnosis and Tolerance in Cryptography - FDTC'05, volume 2779 of Lecture Notes in Computer Science, pages 142 - 151. Springer, 2005.

[15]. C. Giraud. An RSA Implementation Resistant to Fault attacks and to Simple Power Analysis. IEEE Transactions on Computers, Vol.55, No 9, 2006.

[16]. M.Joye ,A.K.Lenstra and J.-J.Quisquater, "Chinese

reamindering based cryptosystem in the presence of faults", Journal of cryptology, 12(4), pp.241-245, 1999

[17]. M. Joye, P. Pailler, and S.-M. Yen. Secure evaluation of modular functions. In International Workshop on Cryptology and Network Security 2001, pages 227 - 229, 2001.

[18]. M. Joye and S.-M. Yen. The montgomery powering ladder. In Cryptographic Hardware and Embedded Systems - CHES'02, volume 2523 of Lecture Notes in Computer Science, pages 291 - 302. Springer, 2002.

[19]. P. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Advances in Cryptology - CRYPTO 96, volume 1109 of Lecture Notes in Computer Science, pages 104 - 113. Springer, 1996.

[20]. A. Shamir, "How to Check Modular Exponentiation", presented at the rump session of EUROCRYPT'97, May 1997.

[21]. A. Shamir, "Method and Apparatus for Protecting Public Key Schemes from Timing and Fault Attacks" US Patent 5991415, 23 Nov. 1999.

[22] Sung-Ming Yen, Dongryeol Kim, and SangJae Moon, "Cryptanalysis of Two Protocols for RSA with CRT Based on Fault Infection," FDTC 2006, LNCS 4236, pp. 53 - 1, Springer-Verlag, 2006.