Journal of The Korea Institute of Information Security & Cryptology(JKIISC) ISSN 1598-3986(Print)
VOL.23, NO.5, October 2013 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2013.23.5.859

# 페르마정리에 기반하는 오류 주입 공격에 안전한 classical RSA 암호시스템

서 개 원,[1†] 백 유 진,[2] 김 성 경,[3] 김 태 원,[1] 홍 석 희[1‡]
[1]고려대학교, [2]우석대학교, [3]삼성전자

# Secure classical RSA Cryptosystem against Fault Injection Attack based on Fermat's Theorem

GaeWon Seo,[1†] YooJin Baek,[2] SungKyoung Kim,[3] TaeWon Kim,[1] Seokhie Hong[1‡]
[1]Korea University, [2]WooSuk University, [3]Samsung Electronics

요 약

스마트카드, 전자여권 등과 같은 내장형 장치(embedded system) 환경이 늘어나고, 민감한 데이터의 보안에 대한 수요가 증가함에 따라 다양한 부채널 공격에 대한 암호시스템의 안전한 구현이 중요시 되고 있다. 특히, 오류 주입 공격은 암호 시스템 구현에 큰 위협 중 하나이며, 하나의 평문-암호문 쌍에 의해 전체 시스템의 안전성이 위협을 받을 수 있기 때문에 암호시스템 구현자에 의해 심각하게 고려되어야 한다. 오류 주입 공격을 방지하는 몇몇 기술은 다양한 암호시스템을 위해 도입되었지만 여전히 classical RSA 암호시스템에 적용되는 실질적인 오류 주입 공격 대응책으로는 부족하다. 본 논문은 classical RSA 암호시스템을 위한 효율적인 오류 주입 공격 대응법을 제안한다. 제안하는 대응방법은 페르마의 정리를 사용하며 추가 연산이 적다는 이점이 있다.

ABSTRACT

Esmbedded devices such as smart cards and electronic passports highly demand security of sensitive data. So, the secure implementation of the cryptographic system against various side-channel attacks are becoming more important. In particular, the fault injection attack is one of the threats to the cryptosystem and can destroy the whole system only with single pair of the plain and cipher texts. Therefore, the implementors must consider seriously the attack. Several techniques for preventing fault injection attacks were introduced to a variety of the cryptosystem, But the countermeasures are still inefficient to be applied to the classical RSA cryptosystem. This paper introduces an efficient countermeasure against the fault injection attack for the classical RSA cryptosystem, which is based on the famous Fermat's theorem. The proposed countermeasure has the advantage that it has less computational overhead, compared with the previous countermeasures.

**Keywords**: classical RSA, Fault Injection Attack, Fermat's Theorem

## I. 서 론

공개키 암호 시스템 중 하나인 RSA 암호는 Rivest, Shamir, Adleman에 의해서 1978년에 고안된 암호 시스템이며, 매우 큰 정수에 대한 소인수 분해가 어렵고 해를 찾아내는데 많은 시간을 요구한다는 점을 이용한 암호 시스템으로, 키 교환과 디지털서명을 위해 가장 광범위하게 사용된다[1].

기존의 암호 시스템 안전성 증명은 수학적인 방법을 통하여 안전성을 증명하였으나, 암호 시스템이 스

접수일(2013년 7월 2일), 수정일(1차: 2013년 7월 25일, 2차: 2013년 8월 8일), 게재확정일(2013년 8월 8일)
† 주저자, blueaiz@naver.com
‡ 교신저자, hhong@korea.ac.kr(Corresponding author)

마트카드와 전자여권 등 다양한 내장형 장치(embedded system)에 사용되면서 부채널 분석 공격(Side channel attack)이라는 물리적 공격방법으로 분석되기 시작하였다[2]. 부채널 공격이 소개된 후 암호 알고리즘에 대한 다양한 공격방법이 소개되었다. 수동적인 공격방법으로 분류되는 공격은 TA(시간 공격, timing attack), SPA(단순전력분석, simple power analysis), DPA(차분전력분석, differential power analysis), EMA(전자파 분석, electromagnetic analysis)이고, 능동적인 공격은 변형된 외부 클럭 주입, 전압 글리치 주입, 온도 변화등의 방법으로 장치에 오류를 주입하는 FA(오류 주입 공격, fault attack)이다.

오류 주입 공격은 1996년 Bellcore가 중국인의 나머지 정리를 이용한 RSA 암호시스템(CRT-RSA)에 오류 주입 공격방법[3]을 제안한 후, 1년 뒤인 1997년에 D.Boneh[4] 등이 공격방법에 대한 자세한 내용을 언급하였다. 소개한 공격방법은 암호 시스템이 동작하는 스마트카드와 같은 환경에 오류를 주입하여 RSA 암호 시스템의 경우 비밀 값인 소수 값을 공격자가 알아내는 방법이다. 오류 주입 공격에 대한 연구가 활발해지면서 스마트카드와 같은 장치에서 안전한 암호학적 알고리즘을 구현하는 문제가 매우 중요해졌다. 특히 RSA는 현실적으로 가장 널리 쓰는 알고리즘이고 CRT를 RSA에 적용할 경우 효율적으로 연산을 수행할 수 있기 때문에, CRT-RSA를 오류 주입 공격으로부터 안전하게 구현하려는 노력이 지속되어 왔다. 또한 최근 DES[5], RSA[1], ElGamal[6], ECC[7,12], AES[8]등 다양한 암호 알고리즘 역시 오류주입공격에 취약함이 알려졌다. 하지만 CRT-RSA에 대한 공격방법과 대응 방법 역시 많은 연구가 진행되고 있으나, 그 외의 암호 알고리즘에 대한 연구는 미흡하다. 특히 본 논문에서는 classical RSA에 대한 대응방법에 대한 연구를 소개한다. classical RSA(이후, RSA 표기)에 대한 대응 방법에 대한 연구는 아직 미흡한 상황이다.

본 논문은 RSA에 적용되는 오류 주입 공격에 대한 대응 방법을 제안한다. 제안하는 방법은 오류 주입 공격에 취약하다고 알려진 비교 연산을 사용하지 않고, 페르마 정리를 이용하여 적은 추가연산으로도 연산중에 오류가 주입 되었는지를 확인하는 방법을 사용한다. 본 논문에서는 또한 새로운 방어기법의 이론적인 안전성과 추가 연산량에 대한 분석을 수행한다.

본 논문의 구성은 다음과 같다. 2장에서는 RSA 서명 알고리즘과 이에 대한 오류 주입 공격방법을 설명하고, 3장에서는 제안하는 대응 방법을 소개한다. 마지막으로 4장에서는 본 논문의 결론을 맺는다.

## II. RSA 서명 알고리즘 및 오류 주입 공격방법

본 장에서는 RSA 서명 알고리즘에 대한 소개와 RSA에 대한 오류 주입 공격방법을 설명한다.

### 2.1 RSA 암호/서명 알고리즘

공개키 암호 시스템인 RSA는 인수분해의 어려움에 안정성을 기반으로 설계되었다. 즉, 충분히 큰 서로 다른 두 개의 소수 $p, q$에 대하여 $p \times q = N$을 만족하는 $N$이 공개되어도, $N$을 인수분해 하여 $p, q$를 알아내는 것은 매우 어렵다는 사실에 기반을 둔다. 이를 바탕으로 사용자는 자신의 메시지가 노출되지 않도록 상대방과 비밀통신을 할 수 있으며, 전자서명을 통하여 자신임을 인증할 수 있다. RSA 암호시스템의 암·복호화 및 서명·검증 알고리즘은 다음과 같다.

a. 키 생성
- $p, q$ : 서로 다른 두 개의 큰 소수 $(p, q \geq 512 bits)$
- $N$ : $N = p \times q$ , 사용자의 공개키
- $\phi(N)$ : $\phi(N) = (p-1) \times (q-1)$
- $e$ : $\gcd(e, \phi(N)) = 1, 1 \leq e < \phi(N)$, 사용자의 공개키
- $d$ : $ed \equiv 1 (\mathrm{mod} \phi(N))$, 사용자의 개인키

키 생성의 결과로 공개키 $(N, e)$와 개인키 $(N, d)$을 얻는다.

b. 암호화 / 검증

메시지 $m$에 대한 암호화는 다음과 같다.

$$C = m^e \mod N$$

서명 값 $S$에 대한 검증은 다음과 같다.

$$m = S^e \mod N$$

c. 복호화 / 서명

암호문 $C$에 대한 복호화는 다음과 같다.

$$m = C^d \mod N$$

메시지 $m$에 대한 서명 값 $S$는 다음과 같다.

$$S = m^d \mod N$$

### 2.2 오류 주입 공격

본 소절에서는 현재까지 제안된 일반적인 RSA 암호 시스템에 대한 오류 주입 공격에 대해 살펴본다. 현재까지 알려진 오류 주입 공격방법은 3가지로 나눠볼 수 있다. Bellcore에 의해서 제안된 공격방법은 지수승 연산 중 레지스터에 오류가 주입되는 방법[4]과 Bao가 제안한 방법인 지수승 연산에 사용하는 비밀 지수 값 $d$에 오류를 주입하는 방법[9], 마지막으로 지수승 연산 시에 사용하지 않는 메모리나 더미 연산을 이용하여 분석하는 safe error[10] 방법으로 나눠진다. 이 중에서 safe error 방법은 SPA를 막기 위해 multiply always 지수승 연산을 사용할 경우에 대한 공격방법이므로 본 논문에서는 다루지 않는다. Bellcore와 Bao 오류 주입 공격방법은 모두 비트 플립 오류(bit-flip error)모델 이지만, 오류가 주입되는 대상에 따라 Bellcore 오류 방법을 연산 오류, Bao 오류주입 방법을 메모리 오류로 구분할 수 있다.

#### 2.2.1 연산 오류 모델

Bellcore 등이 제안한 공격방법은 공격자가 지수승 알고리즘의 현재 값을 포함하는 메모리영역 안에 일시적인 오류를 주입하여 한 비트가 비트 플립 되었다고 가정한다.

사용된 지수승 알고리즘은 "right-to-left" binary 지수승 알고리즘이다. 따라서 연산 중 오류가 주입된 중간 값을 통해 남은 지수승 연산을 수행하여 최종적으로 오류 서명 값을 얻는다. 공격자는 이러한 서명 값을 수집하여 다음과 같은 분석을 통해 비밀정보를 복원한다. 분석 방법은 다음과 같다.

단계 1) 개인키 $d(=d_{n-1}d_{n-2}\cdots d_0)$를 이용하여 지수승 연산 수행 시, 공격자는 $j$번째 지수승 연산 후 중간 값이 저장되어 있는 레지스터에 오류를 주입한다.

$$\left(\prod_{j=0}^{t-1} m^{d_j 2^j}\right) \pm 2^b \ , \ 0 \le b < n$$

단계 2) 오류가 주입된 중간 값을 통해 나머지 지수승 연산 수행 후 공격자는 오류 서명 값 $\hat{S}$을 얻을 수 있다.

$$\hat{S} = \left(\left(\prod_{j=0}^{t-1} m^{d_j 2^j}\right) \pm 2^b\right) \cdot \prod_{j=t}^{n-1} m^{d_j 2^j}$$

단계 3) 오류 서명을 이용하여 공격자는 다음과 같이 비밀정보를 복원할 수 있다.

$$\begin{aligned}\hat{S} &= \prod_{j=0}^{t-1} m^{d_j 2^j} \cdot \prod_{j=t}^{n-1} m^{d_j 2^j} \pm 2^b \cdot \prod_{j=t}^{n-1} m^{d_j 2^j} \\ &= S \pm 2^b \cdot \prod_{j=t}^{n-1} m^{d_j 2^j} \\ S &= \hat{S} \pm 2^b \cdot \prod_{j=t}^{n-1} m^{d_j 2^j} \\ &= \hat{S} \pm 2^b \cdot m^{d_{[t]}}, d_{[t]} = \sum_{j=t}^{n-1} 2^j d_j\end{aligned}$$

메시지 $m$을 알고 있는 공격자는 $d_{[t]}$를 추측하여 다음 등식의 성립을 확인한다.

$$m = (\hat{S} \pm 2^b \cdot m^{d_{[t]}})^e \bmod N$$

추측한 $d_{[t]}$가 옳다면 공개키 $e$에 대한 지수승 연산 후 값이 메시지 $m$과 같을 것이다. 따라서 공격자는 최종적으로 개인 키 $d$의 $n-t$비트를 알아낼 수 있다. 이와 같은 과정을 서로 다른 오류 서명을 이용하여 반복적으로 수행하면 개인키 $d$의 모든 비트정보를 복원할 수 있다.

#### 2.2.2 메모리 오류

Bao 등이 제안한 공격방법은 공격자는 RSA의 복호화 알고리즘이 수행되는 동안 개인키 $d$가 저장되어 있는 레지스터에 오류를 주입하여 개인키의 임의의 한 비트가 비트 플립 되었다고 가정한다. 이를 통해 오류가 주입된 평문을 얻은 공격자는 다음과정을 통해 개인키를 추출한다.

단계 1) 개인키 $d$의 $i$번째 비트에 오류 발생, 오류가 주입된 개인키는 $\hat{d} = d \pm 2^i$라 표현.

단계 2) 공격자는 오류가 주입된 $\hat{d}$를 이용하여 다음과 같이 복호화 과정을 진행한다.

$$\begin{aligned}\hat{m} &= C^{\hat{d}} \bmod N \\ &= C^{d \pm 2^i} \bmod N\end{aligned}$$

단계 3) 공격자는 다음 두 식을 확인해 봄으로써 오류가 발생했던 개인키의 한 비트를 찾을 수 있다.

$$\hat{m} \cdot m^{-1} = C^{-2^i} \bmod N \Rightarrow d_i = 1$$

$$\hat{m} \cdot m^{-1} = C^{2^i} \bmod N \Rightarrow d_i = 0$$

비트 플립 오류는 랜덤 한 비트에 오류가 주입되므

로 공격자는 서로 다른 $i$에 대해 반복적으로 단계 3을 수행함으로써 개인키 $d$의 $i$번째 비트를 찾을 수 있다. 이러한 공격이 가능한 이유는 만약 개인키 $d$의 $i$번째 비트가 0에서 1로 바뀌었다면 $\hat{m} \cdot m$은 $C^{2^i} \bmod N$ 와 같아질 것이고 1에서 0으로 바뀌었다면 $\hat{m} \cdot m$은 $C^{-2^i} \bmod N$와 같아지기 때문이다.

## III. 오류 주입 공격에 안전한 RSA 알고리즘

본 장에서는 앞장에서 언급한 오류 주입 공격을 동시에 방어할 수 있는 새로운 RSA 알고리즘을 제안한다. 제안하는 알고리즘은 오류가 주입되면 그 오류가 서명문(복호문) 전체에 확산되도록 설계하였다. 이렇게 확산된 오류는 최종적으로 공격자가 예측할 수 없는 랜덤 값을 출력한다.

### 3.1 제안하는 RSA 암호알고리즘

본 소절에서는 제안하는 RSA 알고리즘을 소개한다. 제안하는 대응방법의 가장 큰 특징은 페르마의 소정리를 이용하여 오류 주입 여부를 확인한다. 정리 1은 페르마의 소정리이다.

**정리 1.** (페르마의 소정리, Fermat's little theorem) $p$가 소수이고 $a$를 $p$의 배수가 아닌 자연수라 할 때, 즉, $\gcd(a,p)=1$임이 성립할 때

$$a^{p-1} \equiv 1 \pmod{p}$$

이다.

Algorithm 1은 제안하는 오류 주입 공격에 안전한 RSA이다. 메시지 $m$과 비밀 지수 값 $d$, 그리고 모듈러스 $N$을 사용하여 $m^d \bmod N$을 연산한다.

Algorithm 1에 오류가 주입 되지 않을 경우에는 아래와 같이 연산하므로, 정확한 RSA 서명 연산이 계산된다.

단계5. $c_1 = m^{d-d_0+1} \bmod N'$

단계6. $c_2 = m^{d_0-1} \bmod N'$

단계7. $$\begin{aligned} c' &= (c_1 \cdot c_2 \bmod N) \oplus T_1 \\ &= (m^{d-d_0+1} \cdot m^{d_0-1} \bmod N) \oplus T_1 \\ &= m^d \bmod N \oplus T_1 \end{aligned}$$

,where $T_1 = \underbrace{((m \bmod r) \| ... \| (m \bmod r))}_{\lceil |N|/|r| \rceil}$

단계8.

$$\begin{aligned} c &= c' \oplus T_2 \\ &= ((m^d \bmod N) \oplus T_1) \oplus T_2 \\ &= m^d \bmod N \ (\because T_1 = T_2) \end{aligned}$$

,where $T_2 = \underbrace{((c_1 \bmod r) \| ... \| (c_1 \bmod r))}_{\lceil |N|/|r| \rceil}$

이 때 $\lceil |N|/|r| \rceil$은 모듈러스 $N$의 길이에 $r$의 길이를 나누는 것이다. 예를 들어 $N$이 5비트이고 $r$이 3비트인 경우, $\lceil |N|/|r| \rceil$는 2가 된다.

Algorithm 1. Proposed classical RSA against Fault attacks

input: modulus $N$, private key $d$, message $m$
output: $c = m^d \bmod N$

1. Generate a small prime number $r$
2. $N' \leftarrow N \cdot r$
3. $d_0 \leftarrow d \bmod (r-1)$
4. $d' \leftarrow d - d_0 + 1$
5. $c_1 \leftarrow m^{d'} \bmod N'$
6. $c_2 \leftarrow m^{d_0-1} \bmod N'$
7. $c' \leftarrow (c_1 \cdot c_2 \bmod N) \oplus T_1$
   ,where $T_1 = \underbrace{((m \bmod r) \| ... \| (m \bmod r))}_{\lceil |N|/|r| \rceil}$
8. $c \leftarrow c' \oplus T_2$
   ,where $T_2 = \underbrace{((c_1 \bmod r) \| ... \| (c_1 \bmod r))}_{\lceil |N|/|r| \rceil}$

### 3.2 안전성 증명

본 소절에서는 제안하는 알고리즘에 오류가 주입되었을 경우를 가정하여 안전성을 증명한다.

**Lemma 1.** Algorithm 1은 Bellcore가 제안한 공격방법을 적용하더라도 공격자는 키를 찾을 수 없다.

**Proof**: 제안하는 방법은 두 번의 지수승 연산을 수행하여 서명 값을 생성한다. 따라서 한 번의 지수승 연산의 오류 주입으로 인하여 오류 서명 값을 가지게 되더라도 다른 지수승 연산 값과 곱하면서 공격자가 원하지 않는 랜덤 값으로 오류를 출력하게 된다. 또한, 단계 5의 $c_1$ 지수승 연산 과정 중에 사용되는 레지스터나 연산에 오류가 주입되게 되면 $\hat{T}_2 \oplus T_1$이 0이 아닌 랜덤 한 값이 되므로, 공격자는 랜덤 한 출력 값

을 가진다.

만약 단계 6에 오류 주입을 통해 $d_0$의 값을 알 수 있고, 공격자가 랜덤 한 $r$의 값을 안다고 하면 비밀키 $d$의 특정 부분을 복원할 수 있다. 그렇지만 공격자가 찾은 값이 맞는지를 확인하기 위해서는 추가적인 오류 주입 값이 필요하고 $r$이 64비트 이상의 크기라고 가정한다면, $r$을 추측하고 공격하는 연산량은 실행가능하다고 볼 수 없다. 따라서 제안하는 알고리즘은 Bellcore 공격에 안전하다.

**Lemma 2.** Algorithm 1의 비밀 지수 값 $d$에 지속적인 오류가 주입 될 경우(즉, Bao공격)에 공격자는 키를 찾을 수 없는 랜덤 값을 얻는다.

**Proof**: 오류가 주입 된 $\hat{d}$를 이용하여 $\hat{d_0}$와 $\hat{d'}$가 연산되고, 이를 이용하여 $\hat{c_1}$이 계산되고, $\hat{T_2}$은 $T_1$과 동일한 값을 가진다. 하지만, 아래 식처럼 공격자는 key를 찾을 수 없다.

$d$에 오류가 발생하여 $d'$의 $j$번째 값에 오류가 주입이 되었다면, 단계 5에서

$\left(\prod_{j=0}^{t-1} m^{d_j'2^j}\right)\pm 2^b, \quad 0 \le b < n$가 연산되고,

공격자는 단계7에서 다음과 같은 결과값을 얻는다.

$$\hat{c'} = \left(m^{d_0-1} \cdot \prod_{j=t}^{n-1} m^{d_j'2^j}\left(\prod_{j=0}^{t-1} m^{d_j'2^j} \pm 2^b\right)\right) \oplus T_1$$
$$= \left(m^{d} \pm 2^b m^{d_0-1} \cdot \prod_{j=t}^{n-1} d_j'2^j\right) \oplus T_1$$

단계 8은 다음과 같다.

$$\hat{c} = \hat{c'} \oplus \hat{T_2}$$
$$= \left(m^{d} \pm 2^b m^{d_0-1} \prod_{j=t}^{n-1} d_j'2^j\right) \oplus T_1 \oplus \hat{T_2}$$

하지만 공격자는 $r$을 모르기 때문에 $d_0$를 유추할 수 없다. 따라서 $d$의 지속적인 오류에도 안전하게 동작할 수 있다.

**Lemma 3.** Algorithm 1은 safe-error 공격에 안전하다.

**Proof**: 제안하는 방법은 랜덤 한 값으로 $d$를 $d_0$와 $d'$으로 분해하여 두 번의 지수승 연산을 수행하므로 safe-error로 알아낸 비트 값이 비밀 지수 값인 $d$의 비트 값이라고 할 수 없다. 따라서 본 논문에서 제안하는 알고리즘은 safe-error 공격에도 안전하게 수행될 수 있다.

**Lemma 4.** Algorithm 1은 전력 분석 공격(SPA, DPA, SODPA[11])에 대하여 안전하게 수행될 수 있다.

**Proof**: 제안하는 방법은 모듈러스 랜덤화 마스킹 방법과 비밀 지수 값의 분해로 인하여 전력 분석 공격에도 안전하게 수행 할 수 있다.

### 3.3 알고리즘 복잡도

Algorithm 1의 연산 복잡도를 살펴보면 다음과 같다. 랜덤값 $r$의 비트 길이를 $b$, 모듈러 $N$의 비트 길이는 $l$이라고 한다면, 모듈러스 $(l+b)$비트인 $b$비트의 지수승 연산 1번, $l$비트의 지수승 연산 1번, $l+b$비트의 곱셈 1번, 그리고 $l$비트와 $b$비트의 곱셈 1번으로 계산될 수 있으며 그 연산량은 다음과 같다. 이 때, 지수승 연산은 multiply-always방법을 사용한다고 가정한다.

$$2b(l+b)^2 + 2l(l+b)^2 + (l+b)^2 + bl$$

따라서 원래 지수승 연산 $m^d \bmod N$ 과 비교하여 제안한 방법의 오버헤드는 다음과 같이 요약될 수 있다.

$$\frac{(2b(l+b)^2 + 2l(l+b)^2 + (l+b)^2 + bl) - 2l^3}{l^3} \approx \frac{6b+1}{2l}$$

만일 $b$ 가 $l$ 보다 충분히 작다면, 예를 들어 만일 $l$이 1024비트 모듈러이고 $b$이 64비트로 선택이 된다면, 오버헤드는 약 18.79%이다. 쉬운 참조를 위해 [표 1]은 $b$과 $l$의 대표적인 값에 대한 오버헤드를 요약하였다. 대응 방법이 없는 지수승 연산 $m^d \bmod N$의 연산량은 $l^3$이므로 제안하는 방법의 연산량 오버헤드는 [표 1]에서 확인 가능하다.

[표 1] 연산량 Overhead

| $l$ | $b$ | Overhead(%) |
|---|---|---|
| 1024 | 64 | 18.79 |
| | 128 | 37.5 |
| 2048 | 64 | 9.39 |
| | 128 | 18.77 |

## IV. 결 론

본 논문은 classical RSA 암호시스템에 대하여 새로운 오류 주입 공격 대응 방법을 제안한다. 제안하는 방법은 페르마 정리에 기반하여 안전성을 가지고, 적은 추가 연산량으로 안전하게 사용할 수 있다. 가령 1024비트 모듈러스와 64비트 랜덤값을 사용할 경우 18.79% 정도의 적은 추가 연산을 요구한다. 따라서 제안하는 방법은 계산 능력과 메모리가 제한된 모든 환경에서 안전하게 사용할 수 있다.

## 참고문헌


〔1〕 Rivest, R. Shamir, A. Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM 21(2), 120-126, 1978.

〔2〕 P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," CRYPTO-1999, vol. 1666, Lecture Notes in Computer Science, pp. 388-397, Springer-Verlag, 1999.

〔3〕 Bellcore Press Release, "New threat model breaks crypto codes," Sep. 1996.

〔4〕 D. Boneh, R. DeMillo and R. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults," EUROCRYPT '97, Lecture Notes in Computer Science, vol. 1233, pp. 37-51, Springer-Verlag, 1997.

〔5〕 "Data Encryption Standard," FIPS PUB 46-3: National Bureau of Standards, US Dept. of Commerce, Jan. 1977.

〔6〕 Taher El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," In G. R. Blakley and David Chaum, editors, CRYPTO-1984, volume 196 of Lecture Notes in Computer Science, pp. 10-18, Aug. 1984.

〔7〕 Koblitz, N., "Elliptic curve cryptosystems," Mathematics of computation 48, pp. 203-209, Jan. 1987.

〔8〕 "National Institute of Standards and Technology," FIPS-197: Advanced Encryption Standard〔S〕, Sep. 2001.

〔9〕 F. Bao, R. H. Deng, Y. Han, A. Jeng, A. D. Narasimhalu, andT. Ngair, "Breaking public key cryptosystems on tamper resistantdevices in the presence of transient faults," Security Protocols, LNCS, vol. 1361, pp. 115-124, April. 1997.

〔10〕 Yen, S.M., Kim, S., Lim, S., Moon, S., "A countermeasure against one physical cryptanalysis may benefit another attack," Information Security and Cryptology-ICISC 2001, Lecture Notes in Computer Science, vol. 2288, pp. 414-427, Dec. 2002.

〔11〕 Emmanuel Prouff, Matthieu Rivain, and Régis Bevan, "Statistical analysis of second order differential power analysis," IEEE Trans. on Computers, 58(6), pp. 799-811, 2009.

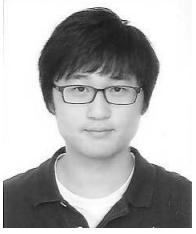
〔12〕 A. Jurisic and A. Menezes, "Elliptic Curves and Cryptography," pp. 1-13, April. 1997.

## 〈저자소개〉


서 개 원 (Gae-Won Seo) 정회원
2006년 2월: 홍익대학교 전자공학과 졸업
2005년 12월~현재: 삼성전자 선임연구원
2011년 9월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 오류주입 공격, 스마트 카드 보안

백 유 진 (TaeWon Kim) 정회원
1997년 2월: 서울대학교 수학과 졸업
1999년 2월: 서울대학교 수학과 이학석사
2003년 2월: 서울대학교 수리과학부 이학박사
2003년 3월~2003년 6월: KAIST 박사후 연구원
2003년 7월~2013년 3월: 삼성전자 책임 연구원
2013년 3월~현재: 우석대학교 정보보안학과 조교수
〈관심분야〉 부채널 공격, 정보 보안

김 성 경(Sung Kyoung Kim) 학생회원
2005년 2월: 부산 동의대학교 수학과/소프트웨어공학과 학사
2007년 8월: 고려대학교 정보경영공학 전문대학교 공학석사
2012년 2월: 고려대학교 정보보호대학원 공학박사
2011년 4월~현재: 삼성전자 S.LSI 사업부 Smart Card개발팀 책임 엔지니어
〈관심분야〉 부채널 공격, 공개키 암호 알고리즘, 암호칩 설계 기술



김 태 원 (TaeWon Kim) 학생회원
2010년 2월: 광운대학교 수학과 학사
2012년 8월: 고려대학교 정보보호대학원 석사
2012년 8월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 부채널 공격, 스마트 카드 보안, 암호시스템 안전성 분석 및 고속구현

홍 석 희 (Seokhie Hong) 정회원
1995년 2월: 고려대학교 수학과 학사
1997년 2월: 고려대학교 수학과 석사
2001년 8월: 고려대학교 수학과 박사
1999년 8월~2004년 2월: (주) 시큐리티 테크놀로지스 선임연구원
2003년 8월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U.Leuven, ESAT/SCD-COSIC 박사후연구원
2005년 3월~현재: 고려대학교 정보보호대학원 부교수
〈관심분야〉 대칭키·공개키 암호 분석 및 설계, 컴퓨터 포레식