Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.6, Dec. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.6.1091

모듈라 곱셈의 충돌 입력에 기반한 부채널 공격 및 대응책*

최 용 제,[1†] 최 두 호,[1] 하 재 철[2‡]
[1]ETRI, [2]호서대학교

Side-Channel Analysis Based on Input Collisions in Modular Multiplications and its Countermeasure*

Yongje Choi,[1†] Dooho Choi,[1] Jaecheol Ha[2‡]
[1]ETRI, [2]Hoseo University

요 약

전력 분석 공격은 물리적 장치에 내장된 암호 알고리듬을 수행할 때 발생하는 부채널 전력 정보를 이용하여 사용자의 비밀 키를 찾아내는 공격 기법이다. 특히, RSA와 같은 공개 키 암호 시스템에 사용되는 멱승은 수백 번의 모듈라 곱셈으로 이루어져 있는데 이 연산이 전력 분석 공격의 목표가 되어 왔다. 최근에는 동일한 입력을 가지는 두 개의 모듈라 곱셈에서 발생한 전력의 상관 분석을 통해 비밀 키를 추출하는 공격이 제안되었다. 본 논문에서는 모듈라 곱셈의 입력 충돌에 기반한 부채널 공격의 원리를 살펴보고 정규화 특성을 갖는 멱승 알고리듬에 대한 취약성을 분석하였다. 또한, 충돌 입력쌍을 이용한 상관 전력 분석 공격을 포함한 기존 부채널 공격에 대응할 수 있는 효율적인 멱승 방법을 제안하고 안전성을 비교 분석하였다.

ABSTRACT

The power analysis attack is a cryptanalytic technique to retrieve an user's secret key using the side-channel power leakage occurred during the execution of cryptographic algorithm embedded on a physical device. Especially, many power analysis attacks have targeted on an exponentiation algorithm which is composed of hundreds of squarings and multiplications and adopted in public key cryptosystem such as RSA. Recently, a new correlation power attack, which is tried when two modular multiplications have a same input, is proposed in order to recover secret key. In this paper, after reviewing the principle of side-channel attack based on input collisions in modular multiplications, we analyze the vulnerability of some exponentiation algorithms having regularity property. Furthermore, we present an improved exponentiation countermeasure to resist against the input collision-based CPA(Correlation Power Analysis) attack and existing side channel attacks and compare its security with other countermeasures.

Keywords: Power Analysis Attack, Exponentiation Algorithm, Modular Multiplication, Input Collision-based CPA

I. 서 론

최근에는 데이터 암호화나 디지털 서명과 같은 정보보호 서비스를 제공하기 위해 필요한 암호 알고리듬을 스마트 카드와 같은 임베디드 장치에 직접 구현하여 사용하고 있다. 이 경우 사용자는 암호용 디바이스 내부에 자신의 비밀 키를 저장하게 되며

접수일(2014년 10월 13일), 수정일(2014년 11월 17일), 게재확정일(2014년 11월 26일)
* 본 연구는 ETRI의 연구개발과제인 KLF-SCARF 프로젝트로 수행하였음(암호키 누출 검증 및 방지 원천 기술 연구)
† 주저자, choiyj@etri.re.kr
‡ 교신저자, jcha@hoseo.edu (Corresponding author)

이 키를 이용하여 정보보호에 필요한 연산을 수행하게 된다. 그러나 이러한 암호 알고리듬을 전용 칩에 구현하는 과정에서 개발자가 미처 고려하지 못했던 여러 가지 오류나 문제점으로 인해 부채널 공격(Side Channel Attack, SCA)이 가능함이 밝혀졌다[1].

부채널 공격은 크게 수동적 공격과 능동적 공격으로 나누어 볼 수 있는데, 수동적 공격이란 정보보호 디바이스에 구현된 알고리듬이 동작하는 과정에서 발생하는 전력 소비량이나 방출되는 전자기파 등을 측정하고 이를 분석함으로써 비밀 키를 찾아내는 공격 방법이다. 수동적 공격 방법 중에서 소비 전력량에 기반한 공격을 전력 분석(Power Analysis, PA) 공격이라 하며 단순 전력 분석(Simple Power Analysis, SPA) 공격과 차분 전력 분석(Differential Power Analysis, DPA) 공격 등으로 나누어진다[2]. 능동적인 공격 방법으로는 Boneh 등에 의해 제안된 오류 주입 공격(Fault Injection Attack, FA)이 있는데 이 공격은 암호 알고리듬이 수행되는 도중 암호용 칩에 고의적으로 오류를 주입하고 그 출력을 분석하여 비밀 키를 찾아내는 물리적 공격 기법이다[3].

지금까지 대부분의 부채널 공격들은 AES (Advanced Encryption Standard)[4]와 같은 블록 암호 시스템을 비롯하여 RSA(Rivest, Shamir, and Adelman)[5]와 같은 공개 키 암호 시스템을 대상으로 활발하게 연구되어 왔다. 특히, RSA나 D-H(Diffie-Hellman) 암호 시스템[6]을 구현할 때 사용하는 멱승(exponentiation) 알고리듬[7]은 부채널 공격의 주된 목표가 되고 있다.

지금까지 제시되었던 멱승 알고리듬에 대한 수동적 전력 분석 공격으로는 SPA, DPA, Doubling 공격[8], 상관 전력 분석(Correlation Power Analysis, CPA) 공격[9] 등이 있었으며 오류 주입 공격 방법으로는 C-safe Error 공격[10]이 대표적이다. 또한, 오류 주입 공격과 전력 분석 공격을 결합한 새로운 형태의 조합 공격(Passive and Active Combined Attack, PACA)이 제안되기도 하였다[11, 12]. 최근에는 RSA 멱승 알고리듬에 사용되는 두 개의 곱셈 연산에서 입력 충돌쌍에 대한 전력 분석을 실시하여 비밀 키를 찾아내는 입력 충돌에 기반한(Input Collision-based) CPA 공격이 제안되었다[13, 14]. 이 공격은 인접 곱셈과의 전력 상관 관계를 분석하는 공격으로서 단 하나의 멱승 소비 전력 파형을 통해 전체 비밀 키를 찾아낼 수 있는 매우 위협적인 공격 방법이다.

본 논문에서는 RSA 멱승 알고리듬에 대한 입력 충돌 기반 CPA 공격의 성공 요소 및 알고리듬 설계 메커니즘의 취약성을 분석한다. 그리고 Multiply-Only 알고리듬 등 여러 정규화된 이진 멱승 알고리듬들이 이 공격에 취약함을 밝히고자 한다. 또한, 논문에서는 입력 충돌에 기반의 CPA를 비롯하여 기존의 여러 부채널 공격을 방어할 수 있는 대응 알고리듬을 제안하고 다른 멱승 방법들과 안전성면에서 비교 분석한다.

II. 멱승 알고리듬에 대한 부채널 공격

2.1 멱승 알고리듬 및 모듈라 곱셈

공개 키 암호 시스템인 RSA를 이용하여 서명을 하거나 데이터를 복호할 경우에는 다음과 같이 일정한 메시지 M에 대해 비밀 키 d를 지수(exponent)로 하는 멱승 연산을 수행한다. 여기서 N은 두 소수 p와 q의 합성수로서 일반적으로 1024비트 이상의 큰 정수를 사용한다.

$$S = M^d \bmod N \tag{1}$$

멱승을 수행 방법에는 지수로 사용되는 비밀 키 d를 어떤 단위로 나누어 처리하는가에 따라 이진 방식(binary method), m진 방식(m-ary method) 그리고 윈도우 방식(window method) 등이 있다. 그러나 가용 자원이 제한된 구현 환경에서는 효율성을 고려하여 이진 방식을 많이 사용하고 있다[7, 15]. 여기서 비밀 키 d를 l비트라고 가정하면 아래와 같이 표현할 수 있다.

$$d = (d_{l-1} d_{l-2} \ldots\ldots d_1 d_0)_2 \tag{2}$$

또한, 이진 멱승 알고리듬은 비밀 키를 한 비트씩 탐색하는 방향에 따라 Right-to-Left 방식과 Left-to-Right 방식으로 나누어지게 되는데 이러한 키 탐색 구조에 따라 부채널 공격 기술이나 그 대응 기법이 크게 달라진다. 일반적으로 Right-to-Left

방식이 Left-to-Right 방식보다 부채널 공격에 강인한 특성을 가지고 있는 것으로 알려져 있다[8,16,17].

멱승 알고리듬은 지수 크기에 따라 수백 번의 모듈라 곱셈($X = A \cdot B \bmod N$)으로 이루어지는데 모듈라 곱셈은 단순 곱셈($C = A \cdot B$)과 모듈라 감소($X = C \bmod N$)과정으로 나누어진다. 두 정수에 대한 곱셈을 위해서는 다음 Fig. 1과 같이 l비트의 A와 B를 곱하여 $2l$비트 크기의 결과 $C = A \cdot B$를 출력하는 큰 정수 곱셈(Long Integer Multiplication, LIM) 기법을 사용한다. 곱셈 $A \cdot B$에 사용되는 두 정수 중 A를 피승수(multiplicand) B를 승수(multiplier)라 하며 다음과 같이 t비트로 구성된 워드 단위로 표현하여 처리할 수 있다. 여기서 $b = 2^t$이고 $k = \lceil \log_b(A) \rceil$ 이다.

$$A = (A_{k-1}A_{k-2} \ldots\ldots A_1A_0)_b \quad (3)$$

두 정수의 곱셈 결과인 C에 대한 모듈라 감소 $C \bmod N$은 몫 추정 기법에 의해 나눗셈을 이용하거나[15]이나 Montgomery 모듈라 감소 방법[18] 등을 사용하여 처리한다.

Long Integer Multiplication ($C = A \cdot B$)

1. for $i = 0$ up to $k-1$ {
2. $Carry = 0$
3. for $j = 0$ up to $k-1$ {
4. $(UV)_b = C_{i+j} + A_j \cdot B_i + Carry$
5. $C_{i+j} = V$
6. $Carry = U$ }
7. $C_{i+k} = Carry$ }
8. Return(C)

Fig. 1. Long integer multiplication

2.2 전력 분석 및 오류 공격

RSA 멱승 알고리듬에 대한 전력 분석 공격은 암호용 칩에서 연산을 수행할 때 누설되는 전력 소비량을 측정하여 이를 분석함으로써 비밀 키 찾아내는 공격이다. 가장 간단한 공격 방법인 SPA 공격에서는 한 개의 소비 전력 파형만으로 비밀 키 비트 d_i와 관련한 곱셈 연산이 있는 지 여부를 관측하여 전체 비밀 키 d를 추출한다.

지금까지 SPA 공격에 대응하기 위해 Square-Multiply Always 멱승 방식[19]이나 Montgomery

Square-Multiply Always(L-to-R)

1. $S = 1$
2. for $i = l-1$ down to 0 {
3. $S = S \cdot S \bmod N$
4. if($d_i = 1$) $S = S \cdot M \bmod N$
5. else $T = S \cdot M \bmod N$ }
6. Return(S)

Fig. 2. Square-Multiply Always algorithm

Ladder 방식[20] 등이 제안되기도 하였는데 Fig. 2는 Square-Multiply Always 이진 방식을 나타낸 것이다. 이 SPA 대응 알고리듬에서는 비밀 키 비트와 관련한 루프(loop) 연산을 할 경우 자승과 곱셈을 한 번씩만 수행하는데 이와 같은 성질을 알고리듬의 정규성(regularity)이라 한다. 그러나 이 멱승 알고리듬은 단계 5에서 멱승 결과 값과 관련이 없는 더미(dummy) 연산을 사용함으로 인해 C-safe Error 공격[10]에 취약한 특성을 가지고 있다.

다음 Fig. 3은 SPA를 방어하기 위해 제안된 Montgomery Ladder 멱승 알고리듬을 나타낸 것으로서 한 비트의 비밀 키를 처리하는 매 루프마다 한 번씩의 곱셈 연산과 자승 연산을 수행하게 된다. 하지만 Square-Multiply Always 이진 방식이나 Montgomery Ladder 방식은 Left-to-Right 형태의 멱승 구조를 이용하므로 Doubling 공격[8]이나 Relative Doubling 공격[16]에 취약한 특성을 가지고 있다.

DPA 공격은 수십~수백 개의 메시지에 대한 멱승 전력 파형을 수집한 후 통계학적인 특성을 이용하여 비밀 키를 추출하는 공격 기법이다. DPA 공격에 대응하기 위해서는 지수 랜덤화, 메시지 랜덤화, 모듈러스 랜덤화 등과 같은 블라인딩(blinding) 기법을 사용한다[19].

오류 공격은 멱승 연산을 수행하는 도중 공격자가 의도적으로 오류를 주입한 후 그 출력 결과를 분석하여 비밀 키를 찾아내는 공격으로서 C-safe Error 공격이 대

Montgomery Ladder(L-to-R)

1. $S[0] = 1$
2. $S[1] = M$
3. for $i = l-1$ down to 0 {
4. $S[\overline{d_i}] = S[0] \cdot S[1] \bmod N$
5. $S[d_i] = S[d_i]^2 \bmod N$ }
6. Return($S[0]$)

Fig. 3. Montgomery Ladder algorithm

표적이다. C-safe Error 공격은 상기한 Square-Multiply Always 알고리듬과 같이 더미 연산이 있는 경우에 쉽게 적용할 수 있다.

이외에도 오류 주입 공격과 SPA 공격을 결합한 조합 공격 공격이 제시되기도 하였다. 문헌 [11]에서는 명령어를 건너뛰는 오류 주입 실험을 통해 레지스터를 초기화시킬 수 있으며 실제로 비밀 키 추출이 가능함을 검증하였다. SPA나 오류 주입 공격에 강한 알고리듬으로 알려진 BNP(Boscher, Naciri, and Prouff) 멱승 알고리듬[21]도 이 조합 공격에는 취약한 것으로 밝혀졌다[12].

III. 곱셈 충돌 입력쌍을 이용한 CPA 공격

본 장에서는 최근 제안된 부채널 공격 중 모듈라 곱셈의 입력이 충돌하는 멱승 알고리듬에 대한 상관 전력 분석 공격에 대해 살펴본다. 이 공격은 SPA에 대응하기 위한 정규성을 가지고 있고 DPA에 대응하는 블라인딩 기법을 적용한 멱승 알고리듬도 공격할 수 있는 매우 위협적인 공격 방법이다.

3.1 모듈라 곱셈에서의 충돌 입력쌍 추출

Witteman 등은 Fig. 2에 도시한 Square-Multiply Always 알고리듬에 대해 모듈라 곱셈에서의 입력 충돌에 기반한 CPA 공격을 제안하였다[13]. 저자들은 논문에서 곱셈과 자승을 하나의 연산 단위인 세그먼트(segment)로 보고 이 세그먼트간의 소비 전력량을 측정한 후 상호 상관도(cross-correlation)를 분석하면 비밀 키를 찾아낼 수 있음을 실험적으로 증명하였다.

이 공격 실험에서는 한 번의 곱셈이나 자승 시 발생하는 소비 전력을 하나의 분석 샘플로 압축(compression)한 후 인접 연산과의 전력 상관도를 계산하였다. 즉, 두 랜덤 변수 X와 Y에 대한 Pearson의 상관 계수를 다음과 같이 두고 상관도를 분석하였다.

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X) \cdot Var(Y)}} \tag{4}$$

구체적으로 Fig. 2의 단계 4와 5에서 보면 비밀 키 비트가 1일 때는 S값이 갱신되지만 0일 때는 S값이 갱신되지 않음을 볼 수 있다. 따라서 비밀 키 d_i가 0일 때는 i번째 루프의 단계 5와 다음 루프의 단계 3을 연속해서 수행하게 된다.

$$\begin{cases} T = S \cdot M \mod N, & i-th \text{ loop} \\ S = S \cdot S \mod N, & (i+1)-th \text{loop} \end{cases} \tag{5}$$

이 경우 곱셈과 자승 연산에 사용된 첫 번째 피승수 S는 서로 동일한 값임을 알 수 있다. 따라서 이 두 연산의 전력 소비량은 서로 높은 상관도를 가지고 있다.

그러나 비밀 키 d_i가 1일 때는 i번째 루프의 단계 4를 수행한 후 다음 루프의 단계 3을 수행하게 된다.

$$\begin{cases} S = S \cdot M \mod N, & i-th \text{ loop} \\ S = S \cdot S \mod N, & (i+1)-th \text{loop} \end{cases} \tag{6}$$

그런데 첫 번째 곱셈 과정에서 S값이 이미 갱신된 상태이므로 두 연산에서 사용된 피승수는 같은 값이 아니다. 따라서 이 경우에는 곱셈과 자승 연산은 낮은 전력 상관도를 가지게 된다.

결국, 공격자는 i번째 루프의 곱셈 연산과 $(i+1)$번째 루프의 자승 연산이 높은 상관도를 가지면 비밀 키 비트를 0으로 판단하고, 반대로 낮은 상관도를 가지면 1로 판단하여 모든 비밀 키 비트를 찾아낸다. 결국, 충돌 입력쌍을 이용한 CPA 공격에서는 두 개의 연산 단위에서 동일한 피승수나 승수를 사용하는지 여부를 판단할 수 있는 전력 파형간의 상관도 분석을 통해 사용자의 비밀 키를 추출해 낸다.

이와 같은 입력 충돌에 기반한 공격 기법은 Joye가 제안한 Square-Multiply Ladder 알고리듬에도 유사하게 적용될 수 있음이 최근 밝혀졌다[14]. 저자들의 논문에서는 공격 알고리듬을 타원곡선 암호 시스템에 사용되는 스칼라 곱셈에 대한 공격으로 설명하고 있으나 공격 메커니즘은 RSA 멱승 연산에서도 그대로 적용될 수 있다. 여기서 주목할 것은 Witteman 등의 방법에서는 상관도 분석을 통해 비밀 키 비트가 0인지 1인지 직접 판단한 것에 비해 이 공격에서는 인접한 비밀 키 비트와의 일치 여부를 판단하여 비밀 키 전체를 추출한다는 차이점이다

3.2 Multiply-Only 멱승 알고리듬 공격

최근 Kim 등은 오류 주입 공격과 SPA에 대응하는 알고리듬으로 Multiply-Only 멱승 알고리듬[22]을 제안하였는데 본 논문에서는 이 알고리듬도

Multiply-Only Ladder(R-to-L)

1. $S[0]=M$
2. $S[1]=1$
3. $T=M$
4. for $i=0$ up to $l-1$ {
5. $\quad S[d_i]=S[d_i]\cdot T \bmod N$
6. $\quad T=S[\overline{d_i}]\cdot S[d_i] \bmod N$ }
7. return($S[1]$)

Fig. 4. Multiply-Only Ladder algorithm

입력 충돌에 기반한 CPA 공격에 취약함을 밝히고자 한다. Multiply-Only 알고리듬을 나타낸 것이 Fig. 4이다. 이 알고리듬에서는 각 비트 처리를 위해 하나의 루프 연산에서 정규적으로 두 번의 곱셈 연산을 처리하며 자승 연산이 사용되지 않음을 알 수 있다[1].

이 멱승 알고리듬을 분석해 보면 단계 5에서 비밀 키 비트 d_i에 따라 레지스터 $S[d_i]$가 갱신되며 이 값은 단계 6에서 승수 값으로 사용된다. 만약 d_i가 다음 루프의 d_{i+1}과 같은 값이라면 계속해서 $S[d_i]$ 값은 갱신되지만 $S[\overline{d_i}]$ 값은 갱신되지 않는다. 따라서 단계 6의 곱셈은 아래와 같이 공통의 피승수 값을 갖게 된다. 이 과정을 일반화하면 아래와 같다.

1) $d_i=d_{i+1}$ 인 경우

$$\begin{cases} P=A\cdot B \bmod N, & i-th \text{ loop} \\ Q=A\cdot C \bmod N, & (i+1)-th \text{ loop} \end{cases}$$

2) $d_i\neq d_{i+1}$ 인 경우

$$\begin{cases} P=A\cdot B \bmod N, & i-th \text{ loop} \\ Q=B\cdot C \bmod N, & (i+1)-th \text{ loop} \end{cases}$$

따라서 Multiply-Only 알고리듬에서도 두 곱셈 연산에 대해 충돌 입력쌍을 이용한 CPA 공격을 수행하면 비밀 키를 찾을 수 있다.

다음 Fig. 5는 d가 $91=1011011_2$일 때 Multiply-Only 알고리듬의 연산 과정과 두 곱셈 사이의 상관도 분석을 통한 키 추출 과정을 설명한 것이다. 예시한 그림에서 보는 바와 같이 최하위 비트를 1로 가정하면 T를 계산하는 과정에서 i번째 루프의 곱셈과 다음 루프의 곱셈이 같은 피승수를 사용하는 경우를 찾아볼 수 있다.

그림에서는 피승수에 대한 충돌쌍은 짙은 색으로 표시하였다. 그리고 모듈라 곱셈의 충돌이 발생하는 경우에는 인접한 두 비밀 키 비트가 동일함을 알 수 있다. 결국 두 루프의 곱셈에 대한 전력 파형상관도 분석만으로도 인접 비트간의 연관성을 구할 수 있고 순차적으로 모든 비밀 키를 추출할 수 있다.

IV. 대응 방법 제안 및 비교 분석

모듈라 곱셈에서의 충돌 입력쌍 분석을 통한 CPA 공격을 방어하기 위해서는 공통의 피승수를 가지는 연산이 발생하지 않도록 알고리듬을 설계하여야 한다. 그러나 대부분의 멱승 알고리듬들은 레지스터에 저장된 이전 값을 순차적으로 갱신하는 과정으로 이루어져 있기 때문에 공통 피승수 곱셈 없이 알고리듬을 구현하는 것은 쉽지 않다. 또한, 공통 피승수 입력을 갖는 연산이 있더라도 비밀 키와 연관된 정보를 누출하지 않도록 하는 것이 중요한 설계 요소가 된다.

본 논문에서는 입력 충돌에 기반한 CPA를 방어하기 위해 피승수와 승수의 위치를 비밀 키에 따라 변경함으로써 인접 루프에서 충돌 입력쌍이 발생하지 않도록 알고리듬을 설계하였다. 이를 위해 다음과 같은 공격 모델과 관련한 가정이 필요하다.

곱셈 $A\cdot B \bmod N$와 $A\cdot C \bmod N$를 연산할 경우 모두 입력 A를 공통 피승수로 가지고 있어 소비되는 전력 파형은 $C\cdot D \bmod N$와 같은 공통의 오퍼랜드가 없는 연산에 비해 서로 높은 상관도를 가진다는 가정이다. 또한, 곱셈 $A\cdot B \bmod N$와 $C\cdot A \bmod N$는 모두 입력 A를 가지고 있지만 하나는 피승수로 다른 하나는 승수로 사용되어 전력 분석 측면에서는 서로 상관도를 가질 수 없다고 가정이다. 즉, 두 개의 곱셈에서 입력 값 A가 모두 피승수에 위치하지 않는다면 충돌 입력쌍으로 보지 않는다는 점이다. 실제로 Fig. 1의 LIM 연산에서 보듯이 피승수와 승수의 역할과 연산 순서는 완전히 다르므로 입력 A가 하나는 피승수로 다른 하나는 승수로 사용될 경우 두 곱셈 연산은 낮은 전력 상관도를 가지게 된다. 이와 같은 가정은 Witteman 등의 실험에서도 검증된 바 있으며[13] 본 논문에서도 전력 분석 실험을 확인하였다.

1) 실제로 그림에서 $i=0$일 때의 루프에서는 한번 자승이 필요하지만 이는 곱셈 연산으로 처리할 수 있다. 그러나 이 멱승 방식을 타원 곡선 암호시스템에서의 스칼라 곱셈에 활용할 경우, 두 배 연산(doubling)을 덧셈(addition) 연산으로 처리할 수 없어 이 알고리듬을 그대로 적용할 수는 없다.

i	d_i	$S[0]$	$S[1]$	T	d_i
		M	1	M	
0	1		$S[1]=S[1]\cdot T$ $=1\cdot M=M$	$T=S[0]\cdot S[1]$ $=M\cdot M=M^2$	$d_0=1$
1	1		$S[1]=S[1]\cdot T$ $=M\cdot M^2=M^3$	$T=S[0]\cdot S[1]$ $=M\cdot M^3=M^4$	$d_0=d_1(1)$
2	0	$S[0]=S[0]\cdot T$ $=M\cdot M^4=M^5$		$T=S[1]\cdot S[0]$ $=M^3\cdot M^5=M^8$	$d_1\neq d_2(0)$
3	1		$S[1]=S[1]\cdot T$ $=M^3\cdot M^8=M^{11}$	$T=S[0]\cdot S[1]$ $=M^5\cdot M^{11}=M^{16}$	$d_2\neq d_3(1)$
4	1		$S[1]=S[1]\cdot T$ $=M^{11}\cdot M^{16}=M^{27}$	$T=S[0]\cdot S[1]$ $=M^5\cdot M^{27}=M^{32}$	$d_3=d_4(1)$
5	0	$S[0]=S[0]\cdot T$ $=M^5\cdot M^{32}=M^{37}$		$T=S[1]\cdot S[0]$ $=M^{27}\cdot M^{37}=M^{64}$	$d_4\neq d_5(0)$
6	1		$S[1]=S[1]\cdot T$ $=M^{27}\cdot M^{64}=M^{91}$	$T=S[0]\cdot S[1]$ $=M^{37}\cdot M^{91}=M^{128}$	$d_5\neq d_6(1)$

Fig. 5. Input collision-based CPA attack on Multiply-Only algorithm

4.1 충돌 입력쌍 회피 멱승 알고리듬

본 논문에서 제안하는 충돌 입력쌍을 이용한 CPA 공격 대응 알고리듬은 Fig. 4에 기술한 Multiply-Only 알고리듬을 기반으로 설계하였다. Multiply-Only 알고리듬은 기본적으로 한 루프마다 두 번의 곱셈을 수행하게 되는데 이때 곱셈의 입력으로 사용되는 두 오퍼랜드 값(피승수와 승수)을 인접 루프와 입력 충돌이 발생하지 않도록 구조화 하였다. 전체 알고리듬을 도시한 것이 Fig. 6이다.

Adjacent Collision Resistant(R-to-L)

1. $S[0]=M$
2. $S[1]=1$
3. $T=M$
4. $t[0]=t[1]=0$
5. for $i=0$ up to $l-1$ {
6. $\quad R[0]=S[d_i]$
7. $\quad R[1]=T$
8. $\quad S[d_i]=R[t[d_i]]\cdot R[\overline{t[d_i]}]$
9. $\quad R[0]=S[d_i]$
10. $\quad R[1]=S[\overline{d_i}]$
11. $\quad T=R[t[d_i]]\cdot R[\overline{t[d_i]}]$
12. $\quad t[d_i]=\overline{t[d_i]}$
13. $\quad t[\overline{d_i}]=\overline{t[\overline{d_i}]}$
14. return($S[1]$)

Fig. 6. Countermeasure against input collision-based CPA

제안 알고리듬에서는 두 개의 임시 플래그(flag) 값 $t[d_i]$를 이용하여 현재 루프에서 사용된 피승수와 승수의 상태를 기억하도록 하였다. 그리고 이 플래그의 상태에 따라 다음 루프 곱셈에서 충돌쌍이 발생하지 않는 피승수와 승수를 결정하도록 규칙성을 부여하여 설계하였다.

제안한 멱승 알고리듬 Fig. 6에서 단계 6부터 11까지의 연산과정을 플래그 값에 따라 구별하여 상술하면 아래와 같다.

- $t[d_i]$가 0일 때
  단계 8 : $S[d_i]=S[d_i]\cdot T$
  단계 11 : $T=S[d_i]\cdot S[\overline{d_i}]$
- $t[d_i]$가 1일 때
  단계 8 : $S[d_i]=T\cdot S[d_i]$
  단계 11 : $T=S[\overline{d_i}]\cdot S[d_i]$

제안 알고리듬에서는 이와 같은 곱셈 과정을 if문과 같은 조건문으로 처리하지 않고 간단한 레지스터 설정 기법을 통해 처리하였다. 그 이유는 if문이 명령어를 건너뛰는 오류 주입 공격에 취약하므로 대응 기법을 무력화될 가능성이 있기 때문이다.

제안하는 멱승 알고리듬이 수행되는 동안 변하는 곱셈의 입력 오퍼랜드 사용 상태를 나타낸 것이 Fig. 7이다. 여기서 $t[d_i]$의 값은 현재 상태를 의미하며 ①, ②, ③, ④는 상태 번호를 나타낸다. 각 상태에서는 알

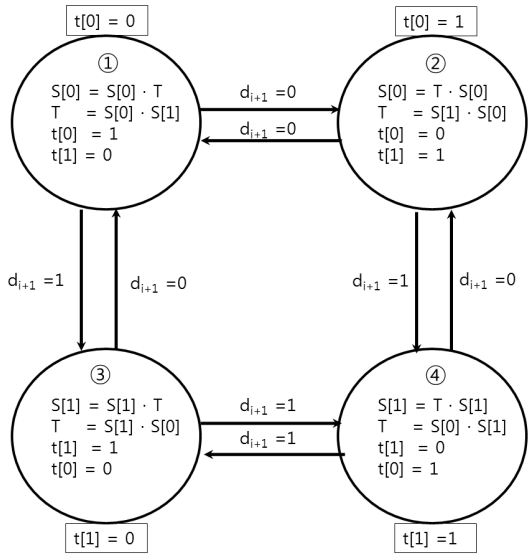


Fig. 7. State diagram of countermeasure algorithm

고리듬에 따라 해당 상태의 곱셈 연산을 수행한다.

그림에서 보는 바와 같이 현재 상태에서 사용한 두 곱셈의 입력 상태를 기억한 후 다음 루프에서 들어오는 비밀 키 d_{i+1}값에 따라 다음 곱셈의 오퍼랜드를 결정하게 된다. 이렇게 함으로써 인접 루프간에는 동일한 입력을 피승수로 갖는 연산이 발생하지 않도록 알고리듬을 구성하였다.

다음 Fig. 8는 위에서 사용한 예제의 d가 $91 = 1011011_2$일 때 제안 멱승 알고리듬의 연산 과정을 설명한 것이다. 예시한 그림에서 보는 바와 같이 인접한 루프에서는 충돌 입력쌍을 가지지 않음을 알 수 있다. 따라서 전력 상관도 분석을 통해서 인접 비밀 키 비트간의 관계도 알 수 없게 된다. 하지만 제안 알고리듬에서는 i번째 루프와 $(i+2)$번째 루프에서는 충돌 입력쌍이 발생할 수 있다. 그 이유는 i번째 루프의 두 번째 곱셈에서 사용된 피승수가 다음 루프에서는 사용되지 않았지만 $(i+2)$번째 루프에서는 사용될 수 있기 때문이다. Fig. 8의 예제에서도 $(i+2)$번째 루프에서 두 개의 충돌 입력쌍이 발생하였음을 볼 수 있다. 그러나 여기서 발생한 충돌 쌍은 공격자가 비밀 키를 추출하는데 유용한 정보가 아니다.

다음 Fig. 9은 i번째 루프와 $(i+2)$번째 루프에서 발생하는 충돌 입력쌍을 정리한 것이다. 한 예로서 Fig. 7의 상태 천이도에서 보면 $t[0]=0$일 때(상태 ①) i번째 루프의 두 번째 곱셈 $T = S[0] \cdot S[1] \bmod N$에서 이용된 $S[0]$는 $(i+1)$번 째 비밀 키가 1이고(상태 ③) $(i+2)$번째 키가 1일 경우(상태 ④)에는 $T = S[0] \cdot S[1]$를 계산할 때 공통의 피승수로 사용됨을 알 수 있다. 따라서 두 곱셈의 입력 충돌이 발생한다. 그리고 $(i+2)$번째 키가 0일 경우(상태 ①)에는 $S[0] = S[0] \cdot T$에서 충돌 입력쌍이 발생하게 된다.

따라서 Fig. 9에 정리한 바와 같이 제안한 대응 알고리듬에는 i번째와 $(i+2)$번째 루프에서 모두 8가지 경우의 입력 충돌이 발생하게 된다. 그러나 곱셈의 입력 충돌이 발생하는 확률은 1/8로 모두 동일함을

i	d_i	State	$S[0]$	$S[1]$	T	$t[0]$	$t[1]$
			M	1	M	0	0
0	1	③		$S[1] = S[1] \cdot T$ $= 1 \cdot M = M$	$T = S[1] \cdot S[0]$ $= M \cdot M = M^2$	$0 \rightarrow 0$	$0 \rightarrow 1$
1	1	④		$S[1] = T \cdot S[1]$ $= M^2 \cdot M = M^3$	$T = S[0] \cdot S[1]$ $= M \cdot M^3 = M^4$	$0 \rightarrow 1$	$1 \rightarrow 0$
2	0	②	$S[0] = T \cdot S[0]$ $= M^4 \cdot M = M^5$		$T = S[1] \cdot S[0]$ $= M^3 \cdot M^5 = M^8$	$1 \rightarrow 0$	$0 \rightarrow 1$
3	1	④		$S[1] = T \cdot S[1]$ $= M^8 \cdot M^3 = M^{11}$	$T = S[0] \cdot S[1]$ $= M^5 \cdot M^{11} = M^{16}$	$0 \rightarrow 1$	$1 \rightarrow 0$
4	1	③		$S[1] = S[1] \cdot T$ $= M^{11} \cdot M^{16} = M^{27}$	$T = S[1] \cdot S[0]$ $= M^{27} \cdot M^5 = M^{32}$	$1 \rightarrow 0$	$0 \rightarrow 1$
5	0	①	$S[0] = S[0] \cdot T$ $= M^5 \cdot M^{32} = M^{37}$		$T = S[0] \cdot S[1]$ $= M^{37} \cdot M^{27} = M^{64}$	$0 \rightarrow 1$	$1 \rightarrow 0$
6	1	③		$S[1] = S[1] \cdot T$ $= M^{27} \cdot M^{64} = M^{91}$	$T = S[1] \cdot S[0]$ $= M^{91} \cdot M^{37} = M^{128}$	$1 \rightarrow 0$	$0 \rightarrow 1$

Fig. 8. Operations of improved Multiply-Only algorithm

알 수 있다. 따라서 곱셈에 대한 충돌 입력쌍을 발견했다고 하더라도 i번째 루프에서 $(i+2)$번째 루프 사이의 비밀 키 값을 결정하거나 인접 비트간의 관계성을 찾을 수 없다. 결국 전력 파형을 통해 $(i+2)$번째 충돌 입력쌍을 찾았다 하더라도 비밀 키 상태가 어떤 경우에 발생한 것인지 알 수 없기 때문에 입력충돌 기반 CPA 공격을 방어할 수 있다.

4.2 이진 멱승 알고리듬의 안전성 비교 분석

본 절에서는 충돌 입력쌍을 이용한 CPA 공격을 비롯한 여러 부채널 공격에 대해 각 멱승 알고리듬을 안전성 측면에서 비교하였다. 이를 요약한 것이 Table 1이다. 먼저 전력 분석 공격 중 SPA, Doubling 공격 그리고 Relative Doubling 공격에 대해 각 알고리듬의 견고성을 분석하였다. 그리고 오류 주입 공격 중 C-safe Error 공격과 조합 공격에 대해서도 안전성을 비교 분석하였다.

지금까지 기술한 이진 멱승 방법들은 모두 SPA 공격에 강인한 정규적 특성을 갖도록 설계하였으며 비밀 키 한 비트를 처리하는데 두 번의 곱셈(혹은 한 번씩의 곱셈과 자승)을 수행한다. 따라서 Table 1에 나타낸 모든 알고리듬은 거의 동일한 수행 시간을 갖는다고 할 수 있다.

먼저, Square-Multiply Always 알고리듬은 Right-to-Left 형태로 구현된 것이라 Doubling 공격이 가능하며 더미 연산이 포함되어 있어 C-safe Error 공격에도 취약하다. Montgomery Ladder 알고리듬 역시 Right-to-Left 형태로 설계되어 있어 Relative Doubling 공격에 취약한 특성을 보이고 있다.

BNP 멱승 알고리듬은 SPA와 C-safe Error 공격에 대응하면서 CRT-RSA 연산에서의 Bellcore 오류 주입 공격을 효과적으로 방어하기 위해 제안되었다. 그러나 이 알고리듬은 특정 레지스터를 강제로 0으로 초기화시킨 후 SPA를 수행하는 조합 공격에 취약하다는 것이 밝혀졌다[12].

Square-Multiply Ladder 알고리듬과 Multiply-Only 알고리듬은 Left-to-Right 형태의 멱승 기법으로서 대부분의 부채널 공격을 방어 할 수 있지만 본 논문에서 살펴본 바와 같이 동일한 입력을 갖는 두 곱셈 연산의 상관도 분석을 통해 비밀 키가 노출될 수 있었다. 물론 입력 충돌에 기반한 CPA를 처음 적용했던 Square-Multiply Always 알고리듬이나 Montgomery Ladder 알고리듬도 동일한 취약성을 가지고 있다.

위에서 언급한 바와 같이 많은 멱승 알고리듬들은 이전 곱셈에 사용되었던 레지스터 값을 이용하여 다른 곱셈 연산을 처리하는 경우가 많다. 따라서 이러한 약점을 이용한 충돌 입력쌍 CPA 공격은 매우 강력한 공격 기법으로 사용될 수 있다. 더구나 이 공격은 한 번의 멱승 연산을 수행한 전력 파형만 필요하기 때문에 메시지에 대한 블라인딩 기법을 적용했다 하더라도 비밀 키가 쉽게 누출될 수 있다[13].

그러나 본 논문에서 제시한 멱승 알고리듬은 기본적으로 Multiply-Only 알고리듬에 기초하여 설계되어 있어 자승 연산이 없고 곱셈만으로 멱승을 수행한다. 따라서 Multiply-Only 알고리듬의 연산 속성을 대부분 그대로 유지하고 있어 SPA,

$t[d_i]$	Collision register	i-th multiplication	$(i+2)$-th multiplication	d_i(state)	d_{i+1}(state)	d_{i+2}(state)
$t[0]=0$	$S[0]$	$T=S[0]\cdot S[1]$	$T=S[0]\cdot S[1]$	0 (①)	1 (③)	1 (④)
			$S[0]=S[0]\cdot T$	0 (①)	1 (③)	0 (①)
$t[0]=1$	$S[1]$	$T=S[1]\cdot S[0]$	$T=S[1]\cdot S[0]$	0 (②)	0 (①)	0 (②)
			$S[1]=S[1]\cdot T$	0 (②)	0 (①)	1 (③)
$t[1]=0$	$S[1]$	$T=S[1]\cdot S[0]$	$T=S[1]\cdot S[0]$	1 (③)	0 (①)	0 (②)
			$S[1]=S[1]\cdot T$	1 (③)	0 (①)	1 (③)
$t[1]=1$	$S[0]$	$T=S[0]\cdot S[1]$	$T=S[1]\cdot S[0]$	1 (④)	1 (③)	1 (④)
			$S[0]=S[0]\cdot T$	1 (④)	1 (③)	0 (①)

Fig. 9. Input collision conditions between i-th and $(i+2)$-th loop

Table 1. Security comparison of countermeasure exponentiation algorithms

Algorithm & Attack	SPA [2]	Doubling [8]	Relative Doubling [16]	C-safe Error [10]	PACA[11]	Input Collision CPA[13]
Square-Multiply Always	O	X	O	X	O	X
Montgomery Ladder	O	O	X	O	O	X
BNP method	O	O	O	O	X	O
Square-Multiply Ladder	O	O	O	O	O	X
Multiply-Only	O	O	O	O	O	X
Proposed	O	O	O	O	O	O

O : Secure, X : Not secure

Doubling 공격, C-safe Error 공격 그리고 조합 공격 등에 대응할 수 있다. 또한 모듈라 곱셈의 피승수와 승수의 위치를 기억하면서 비밀 키 정보에 따라 곱셈의 입력 위치를 변경하는 기법을 사용하여 충돌 입력쌍 CPA 공격을 근본적으로 방어할 수 있다.

4.3 공통 입력을 갖는 곱셈 상관도 분석 실험

본 논문에서는 입력 충돌에 의한 CPA 공격이 가능함을 실제 하드웨어 구현 및 전력 분석 실험을 통해 검증하고자 한다. 실험에는 부채널 검증 보드인 SCARF-AVR 보드를 사용하였다[2]. 실험용 보드에 장착된 메인 CPU는 8비트 프로세서인 ATmega128이며 7.3728MHz의 외부 클럭을 사용하였다. Fig. 10은 전력 분석 실험을 위한 장치 및 테스트 환경을 나타낸 것이다.

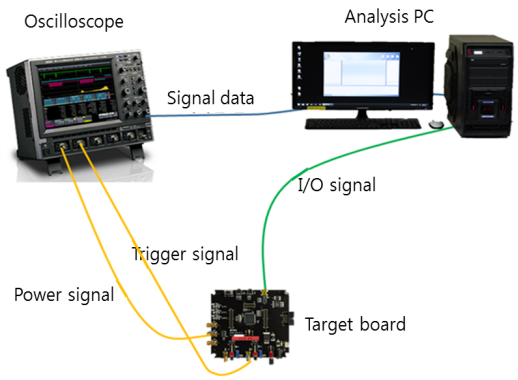


Fig. 10. Experimental board setup

2) SCARF(Side channel Analysis Resistant Framework) 평가 보드에 관한 URL : http://www.k-scarf.or.kr

실험에서는 두 개의 곱셈 연산에서의 전력 분석을 통해 입력 충돌을 찾아낼 수 있는가를 검증하였다. 즉, 하나의 곱셈 $A \cdot B$와 높은 상관도를 갖는 곱셈의 입력 조건을 찾아보았다. 곱셈 알고리듬은 Fig. 1에 제시한 알고리듬을 사용하였으며 식 (4)의 Pearson의 상관 계수를 이용하여 두 전력 파형의 상관도를 분석하였다.

다음 Fig. 11은 여러 곱셈 입력에 대해 입력 충돌 쌍을 찾기 위한 상관도 분석을 실시한 결과이다. 실험에서는 각 곱셈마다 10개씩의 전력 파형을 측정하여 상관도를 측정한 후 이를 평균하였다. 기준이 되는 곱셈은 $A \cdot B$이며 그림에서 보는 바와 같이 동일한 입력을 가지는 다른 곱셈과의 상관 계수는 0.72에 가까운 것을 확인할 수 있다. 동일한 연산임에도 상관 계수가 1에 가깝지 않은 이유는 실험 보드의 잡음이나 신호 처리 과정에서의 신호 손실이 있었기 때문으로 분석된다.

그럼에도 공통의 피승수를 가지는 곱셈 $A \cdot C$는

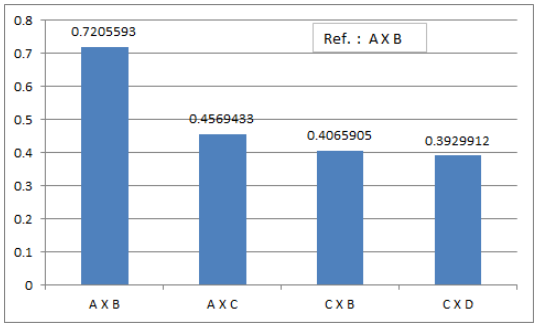


Fig. 11. Experimental power correlation of two multiplications

$A \cdot B$와 비교적 높은 상관 계수인 0.46을 나타냈다. 그리고 공통의 승수를 가지는 곱셈 $C \cdot B$는 $A \cdot B$와 비교적 낮은 상관도 0.41를 보였다. 하지만 공통의 입력을 갖지 않는 $C \cdot D$ 연산은 $A \cdot B$와의 상관도가 적어 가장 낮은 상관 계수 값 0.39를 나타내었다.

이와 같은 실험 결과를 보면 두 개의 곱셈 연산에서는 공통의 피승수를 가지는 연산이 서로 높은 상관성을 유지하며 이 값은 전력 분석을 통해 입력 충돌쌍이 존재하는지 구별할 수 있는 충분히 편차를 가지는 것을 확인하였다. 결국, Multiply-Only Ladder 알고리듬도 구현상의 허점을 이용한 입력 충돌 CPA 공격에 의해 공격될 가능성이 있으며 제안 알고리듬과 같이 입력 충돌을 회피하기 위한 대응책이 강구되어야 함을 실험을 통해 확인할 수 있었다.

V. 결 론

본 논문에서는 RSA 암호 시스템 등에 사용되는 멱승 알고리듬을 암호용 디바이스에 임베디드 형태로 구현할 경우 발생할 수 있는 부채널 공격들을 분석하였다. 특히, 멱승 과정에서 두 개의 모듈라 곱셈 시 충돌 입력쌍이 발생하는 경우, 이 연산에 대한 소비 전력 상관도 분석을 통해 멱승 알고리듬의 비밀 키가 노출될 수 있음을 확인하였다.

따라서 본 논문에서는 충돌 입력쌍을 이용한 CPA 공격 특성을 무력화시킬 수 있는 새로운 이진 멱승 알고리듬을 제안하였다. 제안한 알고리듬은 멱승을 곱셈 연산만으로 처리할 수 있도록 구성하였으며 인접 비밀 키 비트와의 상관성을 제거할 수 있도록 곱셈의 피승수와 승수 위치를 변경하는 기법을 사용하였다. 제안하는 멱승 알고리듬은 기존에 제시되었던 부채널 공격에 대응하면서도 구현을 위해 추가적인 파라미터나 부가적인 연산이 거의 없어 개발 환경이 제한된 암호 디바이스 구현에 매우 효과적으로 사용될 수 있다.

References

[1] P. Kocher, "Timing Attacks on Implementation of Diffie-Hellman, RSA, DSS, and Other Systems,"CRYPTO'96, LNCS 1109, pp. 104-113, Aug. 1996.

[2] P. Kocher, J. Jae, and B. Jun, "Differential power analysis," CRYPTO'99, LNCS 1666, pp. 388-397, Aug. 1999.

[3] D. Boneh, R. DeMillo, and R. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults," EUROCRYPTO'97, LNCS 1233, pp. 37-51, May. 1997.

[4] National Institute of Standards and Technology, "Advanced Encryption Standards," NIST FIPS PUB 197, Nov. 2001.

[5] R. Rivest, A Shamir, and L, Adelman, "A method for obtaining digital signature and public-key cryptosystems," Comm. of the ACM 21, pp. 120-126, Feb. 1978.

[6] W. Diffie and M. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, vol. IT-22, no. 6, pp. 644-654, Nov. 1976.

[7] D. Gordon, "A survey of fast exponentiation methods," Journal of Algorithms, vol. 27, pp. 129-146, May. 1998.

[8] P. Fouque and F. Valette, "The doubling attack- why upwards is better than downwards," CHES'03, LNCS 2779, pp. 269-280, Aug. 2003.

[9] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," CHES'04, LNCS 3156, pp. 135-152, Aug. 2004.

[10] S. Yen, S. Kim, S. Lim, and S. Moon, "A countermeasure against one physical cryptanalysis may benefit another attack," ICISC'01, LNCS 2288, pp. 414-427, Dec. 2001.

[11] F. Amiel, K. Villegas, B. Feix, and L. Mercel, "Passive and Active Combined Attacks: Combining fault attacks and side channel analysis," FDTC'07, IEEE-CS, pp. 92-102, Sep. 2007.

[12] H. Kim and J. Ha, "A physical combined attack and its countermeasure on BNP exponentiation algorithm," Journal of The Korea Institute of Information Security & Cryptology(JKIISC), vol. 23, no. 4, pp. 585-591, Aug. 2013.

[13] M. Witteman, J. Woudenberg, and F. Menarini, "Defeating RSA Multiply-Always and Message Blinding Countermeasures," CT-RSA'11, LNCS 6558, pp. 77-88, Aug. 2011.

[14] B. Feix, M. Roussellet, and A. Venelli, "Side-channel analysis on blinded regular scalar multiplications," Cryptology ePrint Archive, Report 2014/191. 2014. Available at http:eprint.iacr.org/2014/191

[15] D. Knuth, The Art of Programming, Vol 2: Seminumerical Algorithms, 2nd Ed. Addison-Wesley, 1981.

[16] S. Yen, L. Ko, S. Moon, and J. Ha, "Relative doubling attack against Montgomery ladder," ICISC'05, LNCS 3935, pp. 117-128, Dec. 2005.

[17] C. Clavier, B. Feix, G. Gagnerot, and M. Roussellet, "Square Always exponentiation," INDOCRYPT'11, LNCS 7107, pp. 40-57, Dec. 2011.

[18] P. Montgomery, "Modular multipli cation without trial division," Math. of Comp., Vol. 44, No. 170, pp. 519-521, Apr. 1985.

[19] J. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," CHES'99, LNCS 1717, pp. 292-302, Aug. 1999.

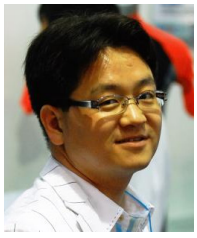
[20] M. Joye and S. M. Yen, "The Montgomery Powering Ladder," CHES'02, LNCS 2523, pp. 291-302, Aug. 2002.

[21] A. Boscher, R. Naciri, and E. Prouff, "CRT-RSA Algorithm Protected Against Fault Attacks," WISTP'07, LNCS 4462, pp. 237-252, May. 2007.

[22] H. Kim, Y. Choi, D. Choi, and J. Ha "A New Exponentiation Algorithm Resistant to Combined Side Channel Attack," Journal of Internet Services and Information Security(JISIS), Vol 3, No. 3/4, pp. 17-27, Nov. 2013.

〈저자소개〉

최 용 제 (Yongje Choi) 정회원
1996년 8월: 전남대학교 전자공학과 졸업
1999년 2월: 전남대학교 전자공학과 석사
1999년 2월~1999년 8월: 전남대학교 전자통신연구소 인턴연구원
1999년 8월~현재: 한국전자통신연구원 선임연구원
〈관심분야〉 보안프로세서 설계, 부채널 분석 시스템, RFID/USN 보안



최 두 호 (Dooho Choi) 정회원
1994년 2월: 성균관대학교 수학과 졸업
1996년 2월: KAIST 수학과 석사
2002년 2월: KAIST 수학과 박사
2002년 1월~현재: 한국전자통신연구원 책임연구원
〈관심분야〉 암호 엔지니어링, 부채널 분석, IoT 보안

하 재 철 (Jaecheol Ha) 종신회원
1989년 2월: 경북대학교 전자공학과 졸업
1993년 2월: 경북대학교 전자공학과 석사
1998년 2월: 경북대학교 전자공학과 박사
1998년 3월~2007년 2월: 나사렛대학교 정보통신학과 부교수
2007년 3월~현재: 호서대학교 정보보호학과 교수
〈관심분야〉 암호 알고리듬, 네트워크 보안, 부채널 공격