Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.25, NO.4, Aug. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.4.727

수평적 상관관계 분석에 안전한 블라인딩 대응기법에 대한 전력 분석 공격

이 상 엽,[1†] 김 태 원,[1] 김 희 석,[2] 홍 석 희[1‡]
[1]고려대학교 정보보호연구원, [2]한국과학기술정보연구원

Power Analysis Attacks on Blinding Countermeasure against Horizontal CPA

Sangyub Lee,[1†] Taewon Kim,[1] HeeSeok Kim,[2] Seokhie Hong[1‡]
[1]Center for Information Security Technologies(CIST), Korea University,
[2]Korea Institute of Science and Technology Information

요 약

현재 전력 분석은 여러 가지 부채널 분석 중 가장 활발하게 연구되고 있다. 1999년 Kocher 등에 의해 차분 전력 분석이 제안된 이후로 소프트웨어/하드웨어 기반 암호 디바이스를 대상으로 하는 다양하고 현실적인 전력 분석 공격이 제안되었다. 본 논문은 공개키 암호 알고리즘에 대하여 단 하나의 파형을 이용하는 전력분석에 안전한 대응기법의 취약성을 분석한다. 2010년 ICICS에서 Clavier 등은 단 하나의 지수승 파형으로 비밀 정보를 찾아낼 수 있는 수평적 상관관계 분석과 그에 대한 대응기법을 제안하였다. 그 중 하나인 "Blind operands in LIM" 대응기법은 큰 정수 곱셈의 두 입력에 대한 덧셈 블라인딩을 이용하여 비밀정보와 관련된 중간 값 노출을 막는다. 그럼에도 불구하고 이 대응기법은 공격자가 알고 있는 평문에 대한 전력 누설을 일으킬 수 있는 취약점을 가지고 있다. 본 논문에서는 세 가지 공격시나리오를 통해 취약점을 분석했고 실제적인 실험을 통해 이를 증명하였다.

ABSTRACT

Until recently, power analysis is one of the most popular research issues among various side channel analyses. Since Differential Power Analysis had been first proposed by Kocher et al., various practical power analyses correspond with software/hardware cryptographic devices have been proposed. In this paper, we analyze vulnerability of countermeasure against power analysis exploiting single power trace of public cryptographic algorithm. In ICICS 2010, Clavier et al. proposed Horizontal Correlation Analysis which can recover secret information from a single exponentiation trace and corresponding countermeasures. "Blind operands in LIM", one of their countermeasures, exploits additive blinding in order to prevent leakage of intermediate value related to secret information. However, this countermeasure has vulnerability of having power leakage that is dependant with the message known by an adversary. In this paper, we analyzed vulnerabilities by three attack scenarios and proved them by practical correlation power analysis experiments.

Keywords: Side-channel power analysis, horizontal correlation analysis, long integer multiplication, additive operand blinding countermeasure

접수일(2015년 4월 2일), 수정일(2015년 5월 7일),
게재확정일(2015년 7월 14일)

† 주저자, pssycho@korea.ac.kr
‡ 교신저자, shhong@korea.ac.kr(Corresponding author)

I. 서 론

부채널 분석(Side channel analysis)은 물리적 디바이스가 작동할 때 발생되는 전자기파, 전력 소모량, 소리, 동작시간 정보 등을 이용하여 비밀 정보를 알아내는 공격이다.

전력 소모량을 이용한 부채널 분석은 1999년 Kocher 등[1]이 차분 전력 분석(Differential Power Analysis, DPA)을 처음 소개한 이래 활발하게 연구되고 있다. 이는 공격자가 가정한 알고리즘상의 비밀 중간 값에 따라 분류한 전력 파형의 차분을 구하여 비밀 정보를 알아내는 공격이다.

2004년 Brier 등은 차분 전력 분석보다 향상된 상관 전력 분석(Correlation Power Analysis, CPA)을 소개하였다[2]. 공격의 주된 아이디어는 공격자가 가정한 비밀 중간 값의 전력 누설 모델과 실제 전력소모량 간의 상관 계수(Pearson's correlation coefficient)를 구하여 비밀 정보를 복원하는 것이다.

부채널 분석에 의한 RSA 알고리즘 공격은 Kocher 등에 의해 제안되었고[1] 이에 대한 구체적인 분석 방법은 1999년 Messerges 등에 의해 소개되었다[3]. 소개된 세 가지 방법 모두 알고리즘상 비밀 중간 값에 따라 측정한 전력 파형을 분류한 후 이들 간의 차분을 구하여 비밀정보에 대한 옳은 가정을 구별해내는 공통점을 가지고 있다.

전력 분석 공격이 제안됨과 동시에 이를 방어하기 위한 대응기법 연구도 진행되었다. Coron은 공개키 알고리즘에서 전력이 비밀 지수에 의존해서 소모되는 취약점을 방지하기 위해 랜덤 수와 모듈러 값을 이용하여 비밀 지수를 매번 랜덤하게 변형시키는 지수 블라인딩(Exponent blinding)을 제안하였다[4].

Walter는 하나의 지수승 파형을 이용하여 공격이 가능한 빅 맥 공격(Big Mac attack)을 제안하였다[5]. 이 공격 방법을 이용하면 d 대신 $d+r\times\phi(n)$을 알아낼 수 있지만 $d\equiv d+r\times\phi(n)\bmod\phi(n)$ 이므로 비밀 정보를 알고 있는 것이나 다름없게 된다. 빅 맥 공격은 일종의 충돌 공격(collision attack)으로 비밀 지수 값(0 또는 1)에 따라 달라지는 큰 정수 곱셈의 평균 파형을 구분하여 비밀 지수를 찾아낸다. 이 공격은 큰 정수 곱셈의 중간 값을 알지 못해도 공격이 가능하지만 정규화 되지 않은 square and multiply 알고리즘을 대상으로 하는 공격이다[6].

2010년 Clavier 등에 의해 수평적 상관관계 분석(horizontal correlation analysis)이 처음으로 제안되었다. 이 방법은 빅 맥 공격과 같이 단 하나의 파형으로 분석이 가능하다는 공통점을 가지고 있지만 상관 전력 분석을 기반으로 하기 때문에 공격자가 중간 값을 알 수 있어야 공격이 가능하다. 그러나 빅 맥 공격과는 다르게 square and multiply always 알고리즘이나 Montgomery ladder 알고리즘과 같은 정규화 된 지수승 알고리즘에 대해서도 공격이 가능하다[7].

이후, Bauer 등[6]은 [7]에서 소개된 "수평적" 부채널 분석 개념과 기존의 "수직적" 부채널 분석 개념을 아우르며 안전성을 분석할 수 있는 일반적인 틀을 제시하였다. 또한 [7]에서 제시된 대응기법의 취약성을 분석하고 새로운 대응기법을 제시하였다.

2013년, Heyszl 등[17]은 하나의 지수승 파형을 이용한 실질적인 공격방법을 제시하였다. 프로파일링, 수동 튜닝(manual tuning), 누수 모델(leakage model) 등이 제공되지 않는 환경에서 하나의 지수승 파형에서 공격에 필요한 파형 묶음을 얻어낼 수 있는 일반적인 알고리즘을 제안하고 FPGA를 이용한 실험 결과를 제시하였다.

본 논문에서는 Clavier 등이 제안한 수평적 상관관계 분석에 대비하는 일종의 덧셈 블라인딩 대응기법인 "Blind operands in LIM"[7] 의 취약점을 분석하고 KLA-Scarf 시스템[20]의 MSP430 부채널 분석 보드를 이용하여 수평적 상관관계 분석 실험을 수행하였다. 취약점 분석에서 드러난 전력 소모량 누설을 수직적 상관관계 분석 실험을 통해 확인하였고 이를 이용하여 수평적 상관관계 분석 실험을 수행하여 비밀 정보 의 한 비트 정보를 복원하였다.

본 논문의 구성은 다음과 같다. 2장에서 수평적 상관관계 분석 방법과 대응기법에 대해 소개한다. 그리고 3장은 2장에서 소개한 대응기법을 분석하고 취약점을 살펴본다. 4장은 제안한 공격방법에 대한 실제적인 실험 결과에 대해 보인다. 5장에서는 이러한 공격에 대한 대응기법에 대해 토의하고 결론을 맺는다.

II. 지수승 알고리즘 구현 방법과 수평적 상관관계 분석

본 장에서는 분석대상에 대한 배경지식을 소개한다. 수평적 상관관계 분석은 지수승 알고리즘을 대상으로 하므로 우선 지수승 알고리즘과 기반이 되는 큰

정수 곱셈에 대해 설명한다. 이 후 수평적 상관관계 분석과 이에 대한 대응기법을 살펴본다.

2.1 모듈러 지수승 연산과 Long Integer Multiplication

RSA, ECC[8] 등의 공개키 기반 암호 알고리즘에서 가장 핵심이 되는 연산은 모듈러 지수승 연산(스칼라 곱셈, 이하 모듈러 지수승에 관해서만 언급)이다. 암호문을 얻기 위해 수행되는 연산 중 차지하는 비중이 가장 클 뿐만 아니라 비밀 지수승에 의존하여 연산이 수행되어 민감한 정보를 가장 많이 담고 있기 때문이다. 모듈러 지수승 연산은 큰 정수 곱셈(Long Integer Multiplication, LIM) 연산을 기반으로 한다. 기본적으로 두 개의 큰 수를 입력으로 받아 두 입력 값의 워드 개수의 곱만큼 단위 워드 곱셈(single precision multiplication)이 이루어진다. 이를 알고리즘으로 표현한 것이 Algorithm 1이다. l개의 워드로 이루어진 두 개의 큰 수 x 와 y를 입력으로 받아 $2l$ 워드의 곱셈 결과를 출력한다. x의 최하위 워드부터 시작하여 큰 수 y와의 곱셈이 l번 발생한다. 큰 수 x의 각 워드와 큰 수 y의 곱셈에는 단위 워드 곱셈이 l번 발생한다. 따라서 한 번의 큰 정수 곱셈에는 l^2개의 단위 워드 곱셈이 이루어진다.

이러한 큰 정수 곱셈을 효율적으로 수행하기 위한 방법으로 Comba의 방법[10], Operand-caching 방법[18], Karatsuba의 방법[11, 19] 등을 사용할 수 있다. (수평적 상관관계 분석은 큰 정수 곱셈에서의 단위 워드 곱셈들의 중간 값을 이용하여 수행되므로 Algorithm 1과 같은 형태가 아니라도 공격이 가능하다.)

Algorithm 2는 square and multiply 지수승이다[12]. 비밀지수 d를 최상위 비트부터 한 비트씩 확인하여 값이 1일 때는 제곱과 곱셈을 수행하고 0일 때는 제곱 연산을 수행한다.

제곱이나 곱셈 결과마다 모듈로 n을 수행하게 되는데 이를 효율적으로 구현하기 위하여 몽고메리 곱셈[9]을 이용한 지수승 연산을 이용할 수 있다.

지수승 연산에서 큰 정수 곱셈의 개수는 비밀 지수 d의 길이에 의존한다. Algorithm 2에서는 v번의 제곱연산이 일어나고 d의 2진수 표현에서 1의 개수만큼의 곱셈 연산이 일어난다. 2진수 표현에서 한 비트의 값이 1일 확률은 $\frac{1}{2}$이므로 Algorithm 2에서 큰 정수 곱셈의 평균 개수는 $v+\frac{v}{2}$ 이다.

Algorithm 1. Long Integer Multiplication

Input : $x=(x_{l-1}x_{l-2}\cdots x_0)_b$, $y=(y_{l-1}y_{l-2}\cdots y_0)_b$

Output : $LIM(x, y)=x\times y$ with maximum length $2l$

1. **for** i from 0 to $2l-1$ **do** $w_i=0$
2. **for** i from 0 to $l-1$ **do**
3. $\quad c\leftarrow 0$
4. $\quad$ **for** j from 0 to $l-1$ **do**
5. $\quad\quad (uv)_b\leftarrow (w_{i+j}+x_i\times y_j)+c$
6. $\quad\quad w_{i+j}\leftarrow v$ and $c\leftarrow u$
7. $\quad w_{i+l}\leftarrow c$
8. **Return**(w)

Algorithm 2. Square and Multiply Exponentiation

Input : integer m, n $(m<n)$
$d=(d_{v-1}d_{v-2}\cdots d_0)_2$

Output : $\mathrm{Exp}(m, d, n)=m^d \bmod n$

1. $a\leftarrow 1$
2. **for** i from $v-1$ to 0 **do**
3. $\quad a\leftarrow LIM(a, a) \bmod n$
4. $\quad$ **if** $d_i=1$ **then**
5. $\quad\quad a\leftarrow LIM(a, m) \bmod n$
6. **Return**(a)

2.2 수평적 상관관계 분석 및 대응기법[7]

2.1.1 수평적 상관관계 분석 방법

모듈러 지수승 연산을 사용하는 공개키 암호 알고리즘에 대한 기존의 전력 분석과 마찬가지로 수평적 상관관계 분석도 비밀 지수 d를 한 비트씩 복원한다.

하지만 여러 개의 파형을 사용하여 통계적 기법으로 비밀정보를 복원하는 기존의 전력분석과는 대조적으로 수평적 상관관계 분석은 단 한 개의 파형을 사용한다. 하나의 모듈러 지수승 파형에는 큰 정수 곱셈이 반복적으로 존재한다. 또한 각각의 큰 정수 곱셈마다 단위 워드 곱셈이 반복된다. 수평적 상관관계 분석은 하나의 모듈러 지수승 파형에서 반복되는 단

위 워드 곱셈에 해당하는 부분을 잘라서 이용한다.

비밀 지수 d의 한 비트 정보에 따라서 큰 정수 곱셈 연산의 입력 값이 달라지고 그에 따라 단위 워드 곱셈의 입출력 값도 달라진다. 잘라낸 단위 워드 곱셈에 대한 부분 파형들과 비밀 지수 d의 비트 변화(0 또는 1)에 따라서 수립할 수 있는 두 가지 전력 모델의 상관관계를 각각 구하여 비교하여 비밀 지수 d 한 비트 정보를 알아낼 수 있다.

이 때, 잘라낸 부분 파형들을 기존의 전력분석과 같이 같은 시간 축에 대하여 수직적으로 정렬하고 각 워드 곱셈에 대한 입출력 값을 이용하여 전력 모델을 수립하고 기존의 상관관계 분석을 수행한다.

공격자가 개인키 $d=(d_{v-1}d_{v-2}...d_0)_2$의 최상위 비트부터 s개의 비트 정보를 알아냈다고 가정하자. 그러면 공격자는 복원해낸 정보를 이용하여 Algorithm 2의 Step 3 또는 Step 5의 중간 값 a를 계산 할 수 있다. 이를 a_s로 나타내기로 한다.

또한 최상위 비트부터 s번째 비트까지 지수승 연산에 따라 수행된 큰 정수 곱셈의 개수는 제곱연산과 지수에 따른 곱셈연산에 의존한다. 따라서 이 때 까지 수행된 큰 정수 곱셈의 개수는 식 (1)과 같이 나타낼 수 있다.

$$t=s+HW(d_{v-1}d_{v-2}...d_{v-s}) \quad (1)$$

(HW는 Hamming Weight를 나타내며 $HW(x)$는 x를 2진수로 나타냈을 때 1의 개수를 의미한다.)

공격자가 한 번의 큰 정수 곱셈 연산 전력 파형에서 구분하여 잘라낸 각각의 단위 워드 곱셈 파형을 파형 조각(curve segment) $C_{i,j}^k$로 정의한다. (k는 몇 번째 큰 정수 곱셈인지 나타낸다. i는 $x=(x_{l-1}x_{l-2}...x_0)_b$의 워드 인덱스, j는 $y=(y_{l-1}y_{l-2}...y_0)_b$의 워드 인덱스이다.

C^k는 k번째 큰 정수 곱셈 $LIM(x, y)$에서 큰 정수 입력 값 x 와 y의 단위 워드 표현 x_i 와 y_j ($i, j=0,1,...l-1$)로 이루어진 모든 단위 워드 곱셈의 파형 조각 $C_{i,j}^k$ ($i, j=0, 1, ...l-1$)을 간략하게 나타낸 것이다.

x_i에 의존하는 단위 워드 곱셈의 파형 조각은 $C_i^k=\sum_{j=0}^{l-1}C_{i,j}^k$으로 나타낼 수 있으며 y_j에 대한 단위 워드 곱셈의 파형 조각은 $C_j^k=\sum_{i=0}^{l-1}C_{i,j}^k$으로 나타낼 수 있다.)

square and multiply 알고리즘에서 $(t+1)$번째의 큰 정수 곱셈 연산은 비밀 지수 d의 $(s+1)$번째 비트 정보에 관계없이 $LIM(a_s, a_s)$이 될 것이다. $(t+2)$번째 큰 정수 곱셈 연산은 $d_{v-s-1}=1$ 이면 $LIM(a_s^2, m)$이 되고 $d_{v-s-1}=0$이면 $LIM(a_s^2, a_s^2)$가 된다. 공격자는 C^{t+2}에서 m을 입력으로 하는 큰 정수 곱셈이 이루어지는지 수평적 상관관계 분석법을 통하여 알아낼 수 있고 따라서 d_{v-s-1}의 값을 알아낼 수 있다.

위와 같은 방법의 분석법을 Clavier 등은 세 가지로 소개하였다. 이에 대한 내용은 다음과 같다. (큰 정수 곱셈은 Algorithm 1과 같이 수행되며 두 입력 값의 길이는 각각 l이라고 가정한다.)

1. $C_{i,j}^{t+2}$와 m의 워드 값 m_j의 해밍 웨이트(Hamming Weight, HW)와의 상관 계수를 구하는 것이다. 이는 l개의 C_j^{t+2}를 측정한 파형으로 하고 각각의 $HW(m_j)$를 추측 모델로 하는 Classical CPA와 비슷한 과정이라 할 수 있다.
2. 큰 정수 곱셈 내부의 단위 워드 곱셈 $x_i \times y_j$의 중간 값과의 상관관계를 구하는 것이다. $x=a_s$, $y=m$으로 하여 $HW(a_i \times m_j)$와 $C_{i,j}^{t+2}$와의 피어슨 상관계수를 구한다. 이때에는 첫 번째 방법과 달리 l^2개의 파형 조각과의 상관계수를 구할 수 있다.
3. $C_{i,j}^{t+3}$의 파형 조각과 $a_s^2 \times m$의 결과 값의 해밍 웨이트의 상관관계를 이용하는 것이다. $(t+2)$번째 연산에서 m과의 곱셈 연산이 일어났다면 $(t+3)$번째의 제곱 연산의 입력 값은 $a_s^2 \times m$이 될 것이고, 상관관계가 높게 나타나게 되어 d의 $s+1$번째 비트 값이 1임을 추론할 수 있다.

2.1.2 수평적 상관관계 분석에 대한 대응기법

Clavier 등은 [7]에서 수평적 상관관계 분석에 대한 세 가지 대응기법을 제안하였다. 이러한 대응기법은 다음과 같다.

Blind Operands in LIM

큰 정수 곱셈에서의 두 입력 값에 대한 블라인딩을 하는 것으로써 단위 워드 곱셈 단계에서 워드를 랜덤 값으로 가리는 것이다. 수평적 상관관계 분석법은 단위 워드 곱셈을 이용하여 공격을 수행한다. 다시 말해 공격자는 워드 곱에 대한 중간 값을 추측할 수 있어야 한다. 따라서 이러한 조건을 충족시키지 못하게 곱셈을 랜덤한 두 워드로 진행하도록 설계하는 것이다.

Algorithm 1의 Step 5에서 블라인딩 기법이 적용된 연산 방법은 다음과 같다.

$$(w_{i+j}+(x_i-r_1)\times(y_j-r_2)) \\ +r_1\times y_j+r_2\times x_i-r_1\times r_2+c \quad (2)$$

r_1, r_2는 x_i, y_j와 같은 비트의 랜덤 값이다. 수평적 상관관계 분석이 큰 정수 곱셈 연산의 중간 값 노출을 이용하기 때문에 두 입력 값의 중간 값을 공격자가 알 수 없도록 랜덤한 수를 선택하여 덧셈을 시킴으로써 덧셈 블라인딩을 하는 것이다. 효율성을 위해 $r_1\times y_j$, $r_2\times x_i$, $r_1\times r_2$는 선행계산 될 수 있으며 이 때 선행계산은 상관계수분석에 안전하도록 해야 한다.

Randomize One Loop in LIM and Blind

큰 정수 곱셈에서 단위 워드 곱셈을 할 때 두 입력 값 중 한 쪽은 랜덤한 순서로 입력되도록 하고 다른 쪽은 워드를 랜덤 값으로 가리는 것이다.(여기에서는 Algorithm 1의 큰 정수 입력 x의 워드 순서를 랜덤하게 하고 큰 정수 입력 y의 워드에 대하여 블라인딩하는 것으로 설명한다.)

Algorithm 1의 Step 5의 y_j는 같은 비트길이만큼의 랜덤 값으로 블라인딩이 이루어진다. 따라서 공격자는 y_j에 의존하는 단위 워드 곱셈의 중간 값을 추측할 수 없다.

또한 Step 2의 i가 0부터 $(l-1)$까지 순차적으로 증가하는 것이 아니라 $[0, (l-1)]$에서의 랜덤 순열이 되도록 한다. 공격자가 x_i에 의존하는 중간 값을 이용하기 위해 파형조각 $C_{i'}$를 수집해도 $i\neq i'$이기 때문에 파형조각과 공격자가 예측한 중간 값과의 상관도가 낮아진다.

l개의 워드를 이용하기 위한 랜덤 순열을 추측하려면 $l!$의 경우의 수를 고려해야하기 때문에 추측이 거의 불가능하다.

Randomize the Two Loops in LIM

앞서 서술한 방식에서 워드 단위 블라인딩을 하지 않고 Algorithm 1의 Step 2와 Step 4의 i와 j를 모두 랜덤 순열을 따르도록 하는 방법이다. 공격자는 단위 워드 곱셈의 중간 값은 추측할 수 있지만 파형 조각에 해당하는 중간 값이 어떤 것인지는 알 수 없다. 따라서 파형 조각과 상관도가 높은 중간 값을 알 수 없어 공격이 불가능하다. 또한 공격자가 랜덤 순열을 추측하려면 $(l!)^2$ 경우의 수를 고려해야하기 때문에 추측이 불가능하다.

III. 수평적 상관관계 분석법의 대응기법에 대한 취약성 분석

본 장에서는 2장에서 소개한 수평적 상관관계 분석법의 대응기법에 대한 취약점을 소개한다. 세 가지의 대응기법 중 첫 번째 대응기법을 대상으로 공격방법을 제안하였고 실제적인 실험을 통하여 이를 증명하였다.

3.1 취약점 분석

식 (2)에서 단위 워드 곱셈의 두 입력 x_i와 y_j는 r_1, r_2에 의해 덧셈 블라인딩(additive blinding)이 된다. 이 때 Algorithm 3과 같이 블라인딩 상태에서 곱셈 연산 수행 시 8개의 중간 값을 관찰할 수 있다. (여기서 사전 연산 되는 $r_1\times y_j$, $r_2\times x_i$, $r_1\times r_2$는 [7]에서의 가정에 의해 안전하다고 가정한다. 따라서 위의 세 가지 연산에 대해서 상관계수 분석을 통하여 r_1과 r_2를 복원하는 방법은 배제하기로 한다. 그러므로 우리는 사전 연산되는 위 3개의 중간 값을 제외한 8개의 중간 값을 분석 대상으로 설정하였다.)

8개의 중간 값에 대하여 Step 7, Step 8의 중간 값은 랜덤요소가 사라진 공격자가 예측할 수 있는 값이다. 이것은 2개의 랜덤 수로 워드 곱셈을 진행하지만 옳은 결과 값을 출력하기 위해 값을 보정해 주는 과정이 수행되기 때문이다. Step 7을 수행한 후

의 결과 값 $(w_{i+j}+x_i\times y_j)$이 레지스터 R_0에 쓰여질 때 소모되는 전력소모량은 공격자가 알고 있는 x_i와 y_j에 의하여 모델링이 가능하다. 이와 마찬가지로 Step 8을 수행한 후의 결과 값 $(w_{i+j}+x_i\times y_j)+c$에 의한 전력 소모량과 공격자가 알고 있는 정보 w_{i+j}, x_i, y_j를 이용하여 공격을 수행할 수 있다.

Step 1, Step 2의 결과 값은 공격자가 예측할 수 없는 랜덤 수지만 연산이 수행될 때 x_i와 y_j에 대한 전력 노출이 일어난다. Step 1, Step 2의 뺄셈에서 오른쪽 연산자는 공격자가 알 수 없는 랜덤 수지만 왼쪽 연산자는 x_i(또는 y_j)가 그대로 입력된다. 이 때 디바이스가 뺄셈 연산을 위해 입력 값을 임시 레지스터에 로드할 때 소모되는 전력 소모량은 여러 개의 파형 조각 상 같은 시점에 노출된다. 따라서 이는 공격자가 알고 있는 x_i(또는 y_j) 값으로 모링이 가능하기 때문에 공격을 수행할 수 있다.

Algorithm 3. Additively Blinded Multiplication
Input : x_i, y_j, w_{i+j}, c, r_1, r_2
Output : $(w_{i+j}+x_i\times y_j)+c$
$R_0\leftarrow 0$, $R_1\leftarrow 0$
Securely pre-computed
$R_2=r_1\times y_j$, $R_3=r_2\times x_i$, $R_4=r_1\times r_2$
1. $R_0\leftarrow x_i-r_1$
2. $R_1\leftarrow y_j-r_2$
3. $R_0\leftarrow R_0\times R_1$
4. $R_0\leftarrow w_{i+j}+R_0$
5. $R_0\leftarrow R_0+R_2$
6. $R_0\leftarrow R_0+R_3$
7. $R_0\leftarrow R_0-R_4$
8. $R_0\leftarrow R_0+c$
9. Return(R_0)

3.2 분석 시나리오

이번 절에서는 앞 절에서 소개한 취약점을 이용하여 구체적으로 비밀정보를 복원하는 분석방법을 소개한다. 총 세 가지 시나리오로 나누어 공격방법을 제안하였으며 각각의 공격방법 모두 높은 성공률로 비밀 정보를 복원하였다.

분석 시나리오에서 공격자는 개인키 $d=(d_{v-1}d_{v-2}...d_0)_2$의 최상위 비트부터 s개의 비트 정보를 알고 있다고 가정한다. 그러면 공격자는 s번째 비트에서의 지수승 연산에 대한 결과 값을 계산할 수 있다. (앞 장에서 설명한 바와 같이 이 때 까지 수행된 큰 정수 곱셈의 개수는 식 (1) 이며 이때의 결과 값을 a_s라 한다.)

위의 정보를 이용하여 공격자는 d_{v-s-1} 값을 다음과 같은 과정을 통해 확인할 수 있다.

Algorithm 2의 Step 3에 의해 $(t+1)$번째 연산은 $LIM(a_s,a_s)$의 제곱 연산이다. 만약 $d_{v-s-1}=1$이면 $(t+2)$번째 연산은 Step 5에 해당하는 $LIM(a_s^2,m)$이고, $(t+3)$번째 연산은 Step3에 해당하는 $LIM(a_s^2\times m,a_s^2\times m)$이다.

$d_{v-s-1}=0$이면 $(t+2)$번째 연산은 Step3에 해당하는 $LIM(a_s^2,a_s^2)$이다. $(t+3)$번째 연산은 d_{v-s-2} 값에 따라 달라지지만 공통적으로 큰 정수 곱셈의 왼쪽 연산자의 입력 값이 a_s^4가 된다.

공격자는 $(t+2)$번째 연산에서 d_{v-s-1} 값에 따라 큰 정수 곱셈의 오른쪽 입력 값이 다른 점을 이용할 수 있다. 또한 $(t+3)$번째 연산에서는 d_{v-s-1}값에 따라 큰 정수 곱셈 왼쪽 입력 값이 달라진다. 큰 정수 입력 값에 따라 달라지는 단위 워드 곱셈의 중간 값 또는 결과 값을 이용하여 수집한 파형 조각과의 상관도를 구하면 d_{v-s-1}의 값을 구분할 수 있다.

시나리오 1

큰 정수 곱셈 연산의 왼쪽 연산자 입력 x의 값을 이용하는 분석 방법이다. Algorithm 3은 단위 워드 곱셈 연산에서 공격자가 알고 있는 입력 값에 의한 전력소모량 노출을 막기 위해 x_i, y_j 각각에 랜덤 수와의 뺄셈 연산을 한다. 하지만 입력 값 x_i, y_j에 의한 결과 값은 블라인딩 처리 되지 않는다. 따라서 공격자는 $(t+3)$번째 큰 정수 곱셈의 입력 값을 이미 알고 있는 a_s을 이용하여 구할 수 있다. 또한 앞에서 분석한 바와 같이 Algorithm 3의 Step 1에서 x_i-r_1 연산이 일어날 때 x_i 값은 노출된다. 따라서 공격자가 계산할 수 있는 $(t+3)$번째 큰 정수 곱셈의 입력 값 $a_s^2\times m$(또는 a_s^4)을 이용하여 수평적 상관계수 분석을 수행할 수 있다.

공격자는 $(t+3)$번째 큰 정수 곱셈 연산에 해당하는 파형에서 단위 워드 곱셈에 해당하는 파형조각 C_i^{t+3} 을 구한다. (큰 정수가 l비트 길이이면 x_i $(i=0,1,...l-1)$에 해당하는 l개의 파형조각을 수집한다.)

큰 정수 $x=a_s^2\times m$일 때의 x_i 값을 구하고 $HW(x_i)$와 수집 파형 간의 상관계수 분석을 한다. 파형조각과 예측한 중간 값의 상관도가 높게 나타나면 $d_{v-s-1}=1$이고 반대의 경우 $d_{v-s-1}=0$임을 알 수 있다.

시나리오 2

시나리오 1과 같은 맥락으로 큰 정수 곱셈 연산의 오른쪽 연산자 입력 y의 값을 이용하는 분석 방법이다. 앞에서와 마찬가지로 공격자는 $(t+2)$번째 큰 정수 곱셈의 입력 값을 이미 알고 있는 a_s을 이용하여 구할 수 있다. 그리고 단위 워드 곱셈의 입력 값 y_j는 Algorithm 3의 Step 2에서 노출된다. 따라서 공격자가 계산할 수 있는 $(t+2)$번째 큰 정수 곱셈의 입력 값 m(또는 a_s^2)을 이용하여 수평적 상관계수 분석을 수행할 수 있다.

공격자는 $(t+2)$번째 큰 정수 곱셈 연산에 해당하는 파형에서 단위 워드 곱셈에 해당하는 파형조각 C_j^{t+2} 을 구한다. (큰 정수가 l비트 길이이면 y_j $(j=0,1,...l-1)$에 해당하는 l개의 파형조각을 수집한다.)

큰 정수 $y=m$일 때의 y_j 값을 구하고 $HW(y_j)$와 수집 파형 간의 상관계수 분석을 한다. 파형조각과 예측한 중간 값의 상관도가 높게 나타나면 $d_{v-s-1}=1$이고 반대의 경우 $d_{v-s-1}=0$임을 알 수 있다.

시나리오 3

앞의 두 가지 시나리오는 공격자가 이전의 큰 정수 곱셈의 결과 값을 계산하여 다음 연산에 필요한 입력 값을 알고 있을 때, 단위 워드 곱셈 중간에 입력 값에 대한 블라인딩에도 불구하고 입력 값에 대한 전력소모량 노출이 일어남을 이용한 것이다. 그러나 이 시나리오는 단위 워드 곱셈 중간 값이 가려진다 하더라도 공격자가 알고 있는 입력 값에 대한 연산 결과 값이 Algorithm 3에 의해서도 여전히 예측 가능한 것을 이용한다. (랜덤 수에 의해 입력 값에 대한 중간 값만 블라인딩 처리 될 뿐 연산의 결과 값은 블라인딩 처리 되지 않기 때문이다.) 따라서 공격자는 $(w_{i+j}+x_i\times y_j)+c$의 결과 값을 이용하여 분석할 수 있다. (이 값은 Algorithm 3의 Step 8에서 저장되는 R_0 값과 같다.)

공격자는 $(t+2)$번째 큰 정수 곱셈 연산에 해당하는 파형에서 단위 워드 곱셈에 해당하는 파형조각 $C_{i,j}^{t+2}$ 을 구한다. (큰 정수 x, y가 모두 l비트 길이이면 $x_i\times y_j$ $(i,j=0,1,...l-1)$에 해당하는 l^2개의 파형조각을 수집한다.)

$x=a_s^2\times m$, $y=m$일 때 l^2개 단위 워드 곱셈의 중간 값 $(w_{i+j}+x_i\times y_j)+c$ 을 구하고 $HW((w_{i+j}+x_i\times y_j)+c)$ 과 수집 파형 간의 상관계수 분석을 한다. 파형조각과 예측한 중간 값의 상관도가 높게 나타나면 $d_{v-s-1}=1$이고 반대의 경우 $d_{v-s-1}=0$임을 알 수 있다.

위의 세 가지 시나리오를 통한 상관 계수 분석을 하여 피크가 발생하면 블라인딩 대응기법에도 불구하고 공격자가 알 수 있는 중간 값으로 모델링 할 수 있는 전력 누수가 여전히 존재한다는 것이다.

IV. 실험 결과

본 절에서는 앞 절에서 소개한 취약점과 이를 이용한 공격 시나리오를 실제파형에 적용하여 실험적으로 검증한다. 중간 값 노출을 방지하기 위하여 큰 정수 곱셈연산에서 블라인딩 기법을 사용했지만 우리는 수집한 파형에 대하여 공격자가 추측할 수 있는 값에 대해 전력 누출이 발생함을 확인할 수 있었다. 또한 이런 정보를 이용하여 수직적 상관관계 분석과 수평적 상관관계 분석으로 비밀 지수의 값을 복원해 낼 수 있다.

4.1 실험 환경

우리는 KLA-Scarf 시스템[20]의 MSP430 부채널 분석 보드를 이용하여 실험을 진행하였다. 실험의 효율성을 위해 지수승 전체를 구현하는 대신 Clavier 등의 블라인딩 대응기법을 적용하여 1024 (64×16)비트 큰 정수 곱셈만을 구현하여 공격을

수행했다. 실험환경에 대한 구체적인 사항은 다음과 같다.

- Scarf MSP430(8MHz) 검증 보드
- Lecroy Waverunner 오실로스코프
- 50MS/s 샘플링 레이트 (6.25 PPC(Point Per Clock)
- 10000개의 파형을 수집
- 파형 6포인트씩 압축 (절대 값의 평균)

4.2 파형 분석

제안하는 수평적 상관 분석은 단위 워드 곱셈의 파형을 이용한다. 큰 정수 곱셈 파형에서 각각의 단위 워드 곱셈을 구분하여 파형 조각을 얻어내기 위하여 공격을 진행하기 전에 수집한 파형에 대하여 시각적인 분석을 수행하였다.

실험에 사용된 MSP430 마이크로프로세서의 워드 단위는 16비트이기 때문에 1024비트 입력 x와 y의 워드 길이 l은 64이다. 따라서 총 4096개의 단위 워드 곱셈이 수행된다.

Fig.1. (a)에서 보이는 65개의 피크로 단위 워드 곱셈 연산을 구분할 수 있다. 하나의 피크와 인접한 피크 사이의 구간동안 64개의 단위 워드 곱셈이 수행된다. Fig.1. (b)는 Fig.1. (a)에서 처음 4개의 피크까지의 파형을 확대한 것이다. Algorithm 1과 같이 y를 고정하고 x를 워드 단위로 스캔하면서 곱셈이 수행됨을 관찰할 수 있다. 또한 이를 확대하면 64개의 단위 워드 곱셈을 관찰할 수 있다.

따라서 관찰된 정보를 이용하여 단위 워드 곱셈을 모두 구분하고 파형을 재구성하여 수평적 상관관계분석에 이용할 수 있다.

4.3 수직적 상관관계 분석

수평적 상관관계 분석을 수행하기 전에 위에서 확인한 정보를 바탕으로 수직적 상관관계 분석을 수행하였다. 분석 시나리오 1과 2에 따르면 x_i와 y_i에 대한 전력소모량이 노출될 수 있다. 이를 이용하여 수직적 상관관계 분석을 하면 각각의 단위 워드 곱셈에서 공격자가 이용하고자하는 중간 값과 상관관계가 큰 파형상의 위치에서 피크가 발생할 것이다. 또한 이러한 위치는 수평적 상관관계 분석에 이용할 수 있다.

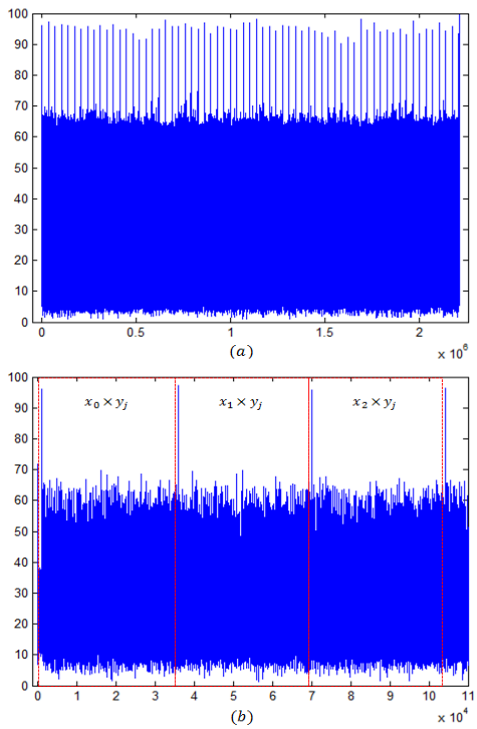

Fig. 1. (a) Power trace of $LIM(x, y)$ operation (b) Position of operations $x_0 \times y_j$, $x_1 \times y_j$, $x_2 \times y_j$ ($j = 0,1,...63$)

Fig. 2. (a)에서 관찰되는 64개의 피크는 $HW(x_i)$ ($i = 0, 1, ..., 63$)와 전력 파형과의 상관도를 나타낸다. $x_0, x_1, ..., x_{63}$값에 대하여 순차적으로 피크가 발생하는 것을 볼 수 있으며 각각의 피크가 Fig.1. (b)에서 확인한 단위 워드 곱셈 64개가 발생하는 구간만큼 존재함을 확인할 수 있다.

Fig.2. (b)는 Fig.2. (a)에서 나타나는 피크 중 하나를 확대한 것이다. 한 가지 색에 대하여 64개의 피크를 확인할 수 있으며 64개의 단위 워드 곱셈에서 x_1에 대한 상관도가 모두 높게 나타나는 것을 확인할 수 있다.

Fig. 3.은 시나리오 2에 따라 $HW(y_j)$ ($j = 0, 1, ..., 63$)를 이용한 수직적 상관계수 분석결과이다. 각각의 색상마다 색상이 다른 것을 확인할 수 있다. (각각의 y_j ($j = 0, 1, ..., 63$)에 대한 피크의 위치를 확인하기 위해 $HW(x_1)$에 대한 상관계수 피크를 회색으로 겹쳐서 나타내었다.) 하나의 피크당 단위 워드 연산 하나가 일어나는 것을 알 수 있

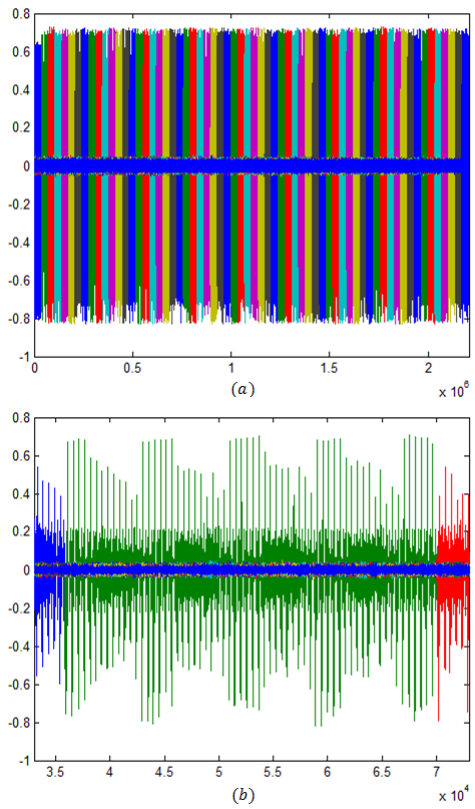

Fig. 2. (a) Result of CPA correlated with $HW(x_i)$ (b) CPA peak trace of x_1

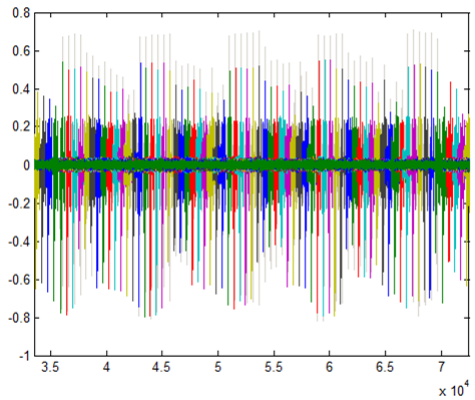

Fig. 3. Result of CPA correlated with $HW(y_j)$, gray trace is result of CPA correlated with $HW(x_1)$

다. 또한 단위 워드 연산에서 y_j에 대한 전력소모량이 노출되는 위치를 확인할 수 있다.

Fig. 4.는 분석 시나리오 3에 따라 Algorithm 3의 Step 8의 R_0 결과 값을 이용한 수직적 상관관계 분석 결과이다. 각각의 피크는 색깔이 다르며 하나의 피크마다 단위 워드 결과 곱셈의 결과 값과의 상관도가 높음을 나타낸다. Fig.4.에서 $i=0$ 일 때와 $i=1$일 때 각각 64개의 서로 다른 피크가 나타나는 것을 확인할 수 있다.

결과적으로 수직적 상관관계 분석을 통하여 분석하였던 전력 노출이 일어나는 것을 모두 확인할 수 있었다. 또한 하나의 단위 워드 곱셈에서 입력 x와 y, 그에 따른 출력에 대해서 전력소모량이 노출되는 위치를 확인할 수 있었다.

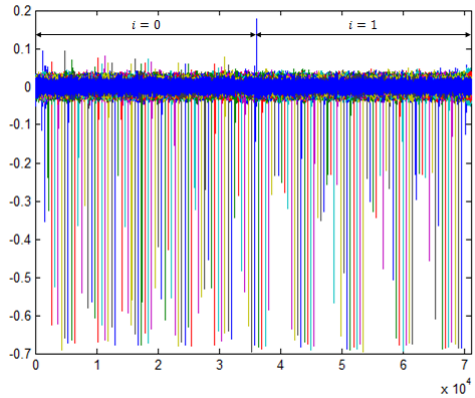

Fig. 4. Result of CPA correlated with $HW((w_{i+j}+x_i\times y_j)+c)$

4.4 수평적 상관관계 분석

이번에는 앞의 실험에서 분석한 파형상의 전력소모량 노출 위치를 이용하여 수평적 상관관계 분석을 수행하였다. 2절의 가.에서 설명한 바와 같이 비밀지수 $d=(d_{v-1}d_{v-2}...d_0)_2$의 최상위 비트부터 s 비트까지 정보를 알아냈다고 가정하면 $(t+2)$번째 큰 정수 곱셈 파형이 $LIM(a_s^2, a_s^2)$에 해당하는지 또는 $LIM(a_s^2, m)$에 해당하는지에 대해 파악하면 비밀정보 d의 한 비트를 알아낼 수 있다.

구체적으로 큰 정수 곱셈의 오른쪽 연산자 입력이 a_s^2인 경우와 m인 경우에 대하여 단위 워드 곱셈의 중간 값을 계산한다. 각각의 경우에 대하여 수평적 상관관계 분석을 수행한다. 수집한 파형조각과 큰 정수 입력이 a_s^2일 때의 중간 값과의 상관도가 높으면 $d_{v-s-1}=0$, 그 반대의 경우는 $d_{v-s-1}=1$임을 추론

할 수 있다.

우리는 위와 같은 공격을 수행하기 위하여 두 가지 시나리오를 이용하였다. 큰 정수 곱셈에서의 오른쪽 연산자에 대하여 추측이 맞는지 확인하여야 하므로 y_j에 의존하는 중간 값을 이용해야 한다. 따라서 $C_j(j=0, 1, ..., 63)$와 $HW(y_j)$의 수평적 상관관계 분석을 수행하고(시나리오 2), $C_{i,j}$ $(i, j=0, 1, ..., 63)$와 $HW((w_{i+j}+x_i\times y_j)+c)$의 수평적 상관관계 분석을 수행하였다(시나리오 3).

Fig.5.는 공격 시나리오 2에 따라 64개의 파형 조각 $C_j(j=0, 1, ..., 63)$를 수집하고 $HW(y_j)$를 이용하여 수평적 상관관계 분석을 수행한 결과이다. 앞의 수직적 상관분석에서 확인한 y_j와 상관도가 높게 나타나는 위치를 중심으로 양쪽으로 500포인트를 수집하여 64개의 파형조각을 얻었다. 이를 이용하여 수평적 상관관계 분석을 수행한 결과 큰 정수 곱셈의 입력이 $y=m$인 경우 피크가 발생함을 볼 수 있다. 회색선은 $y=a^2$인 경우의 수평적 상관관계 분석 결과이다.

Fig.6.는 공격 시나리오 3에 따라 파형조각 $C_{i,j}$ $(i, j=0, 1, ..., 63)$와 $HW((w_{i+j}+x_i\times y_j)+c)$의 수평적 상관관계 분석을 수행한 결과이다. 앞의 수직적 상관분석에서 확인한 Algorithm 3, Step 8의 R_0 결과 값과 상관도가 높게 나타나는 위치를 중심으로 양쪽으로 500포인트를 수집하였다. 이 경우 x_i와 y_j 모두 중간 값에 영향을 미치므로 4096개의 파형조각을 얻었다. 이를 이용하여 수평적 상관관계 분석을 수행한 결과 큰 정수 곱셈의 입력이 $x=a_s^2$, $y=m$인 경우 피크가 발생함을 확인할 수 있다. 회색선은 $x=a_s^2$, $y=a_s^2$ 인 경우의 결과이다.

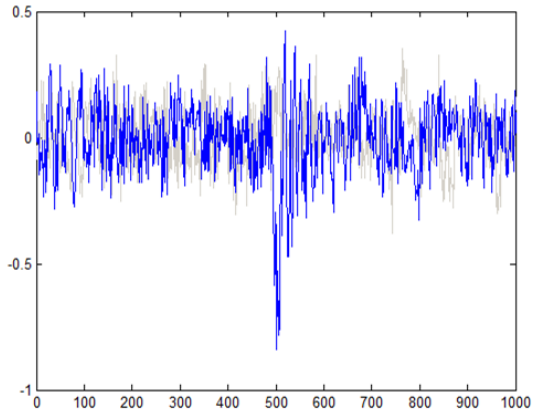

Fig. 5. Result of Horizontal CPA correlated with $HW(y_j=m_j)$

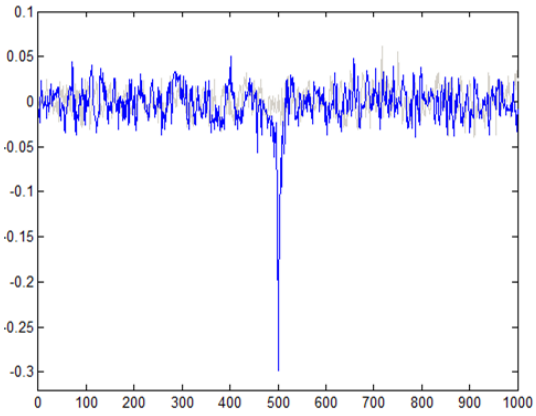

Fig. 6. Result of Horizontal CPA correlated with $HW((w_{i+j}+x_i\times y_j)+c)$ assuming $x=a_s^2$, $y=m$

두 가지의 공격에 모두 $(t+2)$번째 큰 정수 곱셈의 오른쪽 연산자가 m일 경우에 단위 워드 곱셈에서 수집한 파형 조각과 상관관계가 높게 나타남을 확인하였다. 따라서 $d_{v-s-1}=1$임을 확인할 수 있었다.

V. 대응기법

"Blind operands in LIM" 대응기법의 경우에는 위의 실험에서 알 수 있듯 큰 정수 곱셈 알고리즘의 입력 값이 블라인드 되지 않은 상태이다. 알고리즘 안에서 랜덤 값을 이용하여 덧셈 블라인딩(additive blinding)을 하여도 공격자가 알고 있는 입력 값을 그대로 이용한 상관계수 분석이 가능하다는 것을 실험을 통해 알 수 있다.

또한 식 (1)의 $((x_i-r_1)\times(y_j-r_2))+r_1\times y_j+r_2\times x_i-r_1\times r_2$ 연산의 결과는 $x_i\times y_j$ 값과 같으므로 이 경우에도 공격자가 일고 있는 입력 값을 이용하여 $x_i\times y_j$의 결과 값을 구하고 그를 이용하여 상관계수 분석이 가능함을 보였다. 이러한 분석은 수평적 상관관계 분석으로도 가능하다.

따라서 대응기법은 큰 정수 곱셈의 입력 자체가 블라인드된 형태가 되어야 할 것이다. 이는 곱셈 블라인딩(multiplicative blinding)[13]으로 가능하

며, RSA에 대하여 수정된 BRIP 알고리즘[14, 15, 16] 등으로 가능하다. Clavier 등이 분석한 바와 같이 곱셈 블라인딩은 랜덤 값의 비트길이에 의존하여 공격에 대한 취약성이 결정된다[7].

VI. 결 론

본 논문에서는 수평적 상관관계 분석에 대한 Clavier 등의 블라인딩 대응기법[7]에 대한 공격을 실험적으로 보였다. 큰 정수 곱셈 알고리즘에서 각각의 입력 값에 대해 단위 워드 곱셈 단계에서 덧셈 블라인딩을 하여도 전력 누설이 생긴다는 것을 확인하였다. 이는 기존의 대응기법인 곱셈 블라인딩이나 수정된 BRIP 알고리즘 등으로 방지할 수 있다.

그러나 Clavier 등의 덧셈 블라인딩 대응기법은 단위 워드 곱셈 단계에서 이용되는 데이터에 대한 전력 누설을 방지하려는 의도로 제안되었으며 본 논문의 공격 방법도 단위 워드 곱셈을 대상으로 한다. 따라서 단위 워드 곱셈 단계에서 이용되는 데이터에 대한 전력 누설을 원천적으로 막을 수 있는 대응기법에 대한 연구가 필요하다.

References

[1] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis," CRYPTO 1999, LNCS 1666, pp. 388 - 397.

[2] E. Brier, C. Clavier and F. Olivier, "Correlation Power Analysis with a Leakage Model," CHES 2004, LNCS 3156, pp. 16 - 29.

[3] T. Messerges, E. Dabbish, and R. Sloan, "Power Analysis Attacks of Modular Exponentiation in Smartcards," CHES 1999, LNCS 1717, pp. 144-157.

[4] J.S. Coron, "Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems," CHES 1999, LNCS 1717, pp. 292-302.

[5] C.D. Walter, "Sliding Windows Succumbs to Big Mac Attack," CHES 2001. LNCS 2162, pp. 286-299.

[6] A. Bauer, E. Jaulmes, E. Prouff and J. Wild, "Horizontal and Vertical Side-Channel Attacks against Secure RSA Implementations," RSA 2013, LNCS 7779, pp. 1-17.

[7] C. Clavier, B. Feix, G. Gagnerot, M. Roussellet and V. Verneuil, "Horizontal Correlation Analysis on Exponentiation," ICICS 2010, LNCS 6476, pp. 46-61.

[8] P. Gallagher and C. Furlani, "Digital Signature Standard," FIPS PUB 186-3, Oct. 2009.

[9] P.L. Montgomery, "Modular Multiplication without Trial Division," Mathematics of Computation, vol. 44, no. 170, pp. 519 - 521, Apr. 1985.

[10] P.G. Comba, "Exponentiation Cryptosystems on the IBM PC," IBM Systems Journal, vol. 29, no. 4, pp. 526 - 538, 1990.

[11] A.A. Karatsuba and Y.P. Ofman, "Multiplication of Multidigit Numbers on Automata," Soviet Physics Doklady, vol. 7, pp. 595, Jan. 1963.

[12] A. Menezes, P.C. van Oorschot and S.A. Vanstone Handbook of Applied Cryptography, CRC Press, 1996.

[13] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Systems," CRYPTO 1996, LNCS 1109, pp. 104 - 113.

[14] H. Mamiya, A. Miyaji and H. Morimoto, "Efficient Countermeasures against RPA, DPA, and SPA," CHES 2004, LNCS 3156, pp. 343-356.

[15] K. Itoh, T. Izu and M. Takenaka, "Improving the Randomized Initial Point Countermeasure against DPA," ACNS 2006, LNCS 3989, pp. 459-469.

[16] F. Amiel and B. Feix, "On the BRIP Algorithms Security for RSA," WISTP 2008, LNCS 5019, pp. 136-149.

[17] J. Heyszl, A. Ibing, S. Mangard, F. D. Santis and G. Sigl, "Clustering Algorithms for Non-profiled Single-Execution Attacks on Exponentiations," CARDIS 2013, LNCS

8419, pp. 79-93.
〔18〕 M. Hutter and E. Wenger, "Fast Multi-precision Multiplication for Public-Key Cryptography on Embedded Microprocessors," CHES 2011, LNCS 6917, pp. 459-474.
〔19〕 M. Hutter and P. Schwabe, "Multiprecision Multiplication on AVR Revisited," IACR ePrint 2014-592, Jul. 2014.
〔20〕 Y. J. Choi, D. H. Cho, J. C. Ryou, "Implementing Side Channel Analysis Evaluation Boards of KLA-SCARF system,"Journal of The Korea Institude of Information Security & Cryptology, Vol. 24, no. 1 pp. 229-240. 2014.

〈저자소개〉

이 상 엽 (Sangyub Lee) 학생회원
2010년 8월: 고려대학교 전파통신공학과 졸업
2013년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 부채널 분석 및 대응법

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

김 희 석 (HeeSeok Kim) 정회원
2006년: 연세대학교 수학과 학사
2008년: 고려대학교 정보보호대학원 공학석사
2011년: 고려대학교 정보보호대학원 공학박사
2011년 9월~2012년 12월: Bristol University 박사후 연구원
2013년~현재: 한국과학기술정보연구원 (KISTI) 과학기술정보보호실 선임연구원
〈관심분야〉 부채널 공격, 암호시스템 안전성 분석 및 고속구현, 암호칩 설계 기술, 보안관제, 네트워크 보안

홍 석 희 (Seok-hie Hong) 종신회원
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~2013년 8월: 고려대학교 정보보호대학원 부교수
2013년 9월~현재: 고려대학교 정보보호대학원 정교수
〈관심분야〉 대칭키·공개키 암호 분석 및 설계, 컴퓨터 포렌식