정 보 보 호 학 회 지
제 31 권 제2호, 2021. 4

민감 정보 종속 공격

심 보 연*, 한 동 국**

요 약

본 논문에서는 민감 정보, 즉, 비밀 값에 따른 전력 소비 파형을 군집화하여 비밀 값을 찾아내는 민감 정보 종속 공격 동향을 소개한다. 암호 알고리즘의 입·출력 값에 대한 정보 없이 부채널 정보만을 이용한 공격으로 공개키, 대칭키 암호 알고리즘뿐만 아니라 양자 내성(후양자) 암호 알고리즘도 민감 정보 종속 공격에 취약함을 보인다.

Ⅰ. 서 론

과거 안전한 암호 체계 구축을 위한 암호 해독 연구는 이론적인 암호 해독 공격, 즉, 정보보호 기기에 탑재된 암호 알고리즘의 입·출력을 기반으로 하는 수학적 분석을 일컬었다. 1996년 Paul Kocher에 의해 처음 제시된 부채널 공격(SCA, side-channel attack)은 암호 알고리즘이 장비 위에서 동작하는 동안 발생하는 물리적인 정보를 활용하여 비밀 정보를 찾는 것이다. 암호 알고리즘 수행 시간, 수행되는 동안 소비하는 전력량, 방출하는 전자파 및 광자 등을 부채널 정보라고 하며, 이를 계측하여 비밀 정보를 해독할 수 있다. 따라서 수학 이론을 기반으로 하는 암호 해독 공격에 대한 안전성이 증명된 암호 알고리즘이라도 실제 장비 위에서 동작하는 동안 누출하는 부채널 정보 기반 공격에 대한 안전성을 담보할 수 없다. 이에 암호화 장비의 안전성을 평가하기 위해 부채널 공격 분야 연구가 활발히 이루어져 왔으며, 안전한 암호 알고리즘의 사용을 위해 잔존 하는 물리적 취약점이 없는지에 대한 연구는 지속적으로 수행되어야 한다.

본 논문은 현재 널리 사용되는 공개키, 대칭키 암호 알고리즘뿐만 아니라 양자 컴퓨팅 시대를 대비하여 활발히 연구되고 있는 양자 내성(후양자) 암호 알고리즘(PQC, post quantum cryptography)에 대한 신규 부채널 공격 결과를 제시한다. 특히, 제안하는 단일 파형 공격의 경우 부채널 정보만을 이용하기 때문에 암호 알고리즘의 입·출력 값에 대한 정보가 없어도 된다. 더불어 민감 정보 값에 대한 정보가 없어도 되며, 민감 정보 값에 따라 두 개 이상의 분화된 연산이 수행되고, 이에 대한 부채널 정보를 각 연산 집합으로 군집화 할 수 있다는 가정을 기반으로 한다.

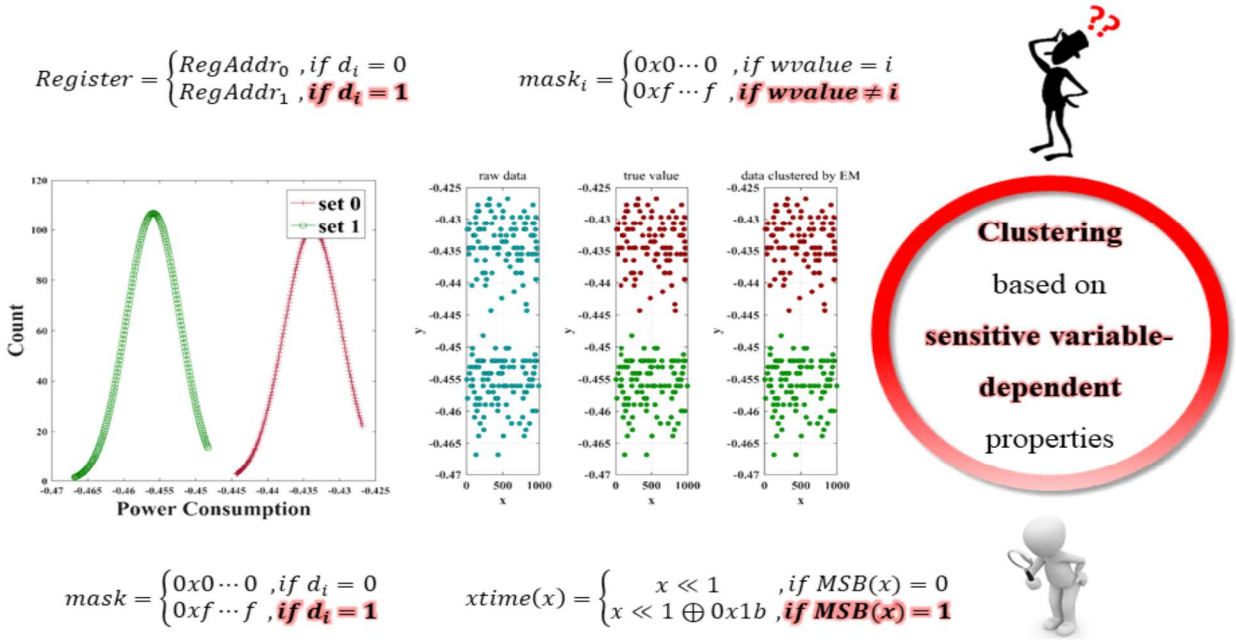
Ⅱ. 민감 정보 종속 공격 [6]

[6]에서 제안하는 민감 정보 종속 공격은 민감한 값, 즉, 비밀 값에 의해 결정되는 중간값에 따른 전력 소비 파형을 군집화하여 비밀 값을 찾아내는 공격이다. 본 논문에서는 이 민감한 중간값을 “결정자”라고 정의하며, 전력 소비 모델이 해밍 웨이트(HW, hamming weight)정보에 의존한다고 가정한다. 이때, HW는 레지스터에 저장된 중간값의 ‘1’ 비트 수이다.

〔**정의 1**〕 결정자는 민감한 값에 따라 정의되는 중간값으로 2가지 경우의 수가 존재하며, 두 경우의 HW 차이는 2 이상이다.

[그림 1]은 민감 정보 종속 공격의 개념을 표현한 그림이다. $Register$, $mask$, $mask_i$, $xtime(x)$값이 비밀 값 d_i, $wvalue$, $MSB(x)$에 의해 그 값이 결정되는 결정자이다. 결정자는 비밀 값에 따라 두 가지 중 한 가지 값을 가지며, 발생 가능한 두 중간값의 HW 차이가 클수록 [그림 1]과 같이 각 경우에 대한 분포가 중첩되는 부분이 작다. 민감 정보 종속 공격은 군집 알고리즘을 통해 비밀 값에 따른 결정자를 두 집합으로 분류하여 비밀 값을 찾는 공격으로, 두 중간값의 HW 차이가 클수록 공격 성공률이 높다. 본 논문에서는 관

* 한국전자통신연구원 (연구원, sboyeon37@etri.re.kr)
** 국민대학교 정보보안암호수학과 (교수, christa@kookmin.ac.kr)



〔그림 1〕 민감 정보 종속 공격

심 영역(PoI, point of interest)를 식별하기 위해 SOST(sum of squared pairwise t-differences) 값을 계산한다. $E(\bullet), \sigma(\bullet), n(\bullet)$는 각 집합의 평균, 표준편차, 원소의 개수를 나타낸다.

$$SOST = \left(\frac{E(G_1) - E(G_2)}{\sqrt{\frac{\sigma(G_1)^2}{n(G_1)} + \frac{\sigma(G_2)^2}{n(G_2)}}} \right)^2$$

Ⅲ. 공개키 암호 알고리즘 RSA/ECC에 대한 민감 정보 종속 공격 [6]

대표적인 공개 라이브러리 OpenSSL과 LibreSSL은 시간 공격 및 단순 전력 분석에 안전한 RSA/ECC 암호 알고리즘을 제공하기 위해 윈도우 기법(window method)을 적용했다. 특히 RSA의 경우 사전 연산 값을 메모리에서 불러올 때 상수 시간 메모리 접근 알고리즘을 사용한다. 이때, 정확한 결과를 얻기 위해 모든 비트가 0이거나 1인 $mask_i$값을 사용하며, 이는 비밀 윈도우 값 $wvalue$을 기준으로 결정된다. 따라서 $mask_i$값을 계산하거나 참조할 때의 전력 소비 파형을 군집화하여 $wvalue$값을 찾을 수 있으므로 비밀키를 찾을 수 있다. 윈도우 크기 w에 관계없이 단일 파형을 이용하여 100%의 성공률로 비밀키를 찾을 수 있다.

본 논문에서는 $w=2$일 때와 $w=6$일 때의 실험 결과를 보인다. LibreSSL 라이브러리는 OpenSSL에서 파생된 라이브러리로 상수 시간 메모리 접근 알고리즘이 같으므로 OpenSSL에 대한 실험 결과를 보인다. 소비 전력 파형은 8비트 AVR 프로세서를 탑재한 ChipWhisperer-Lite XMEGA 보드를 이용하여 29.54MS/s으로 수집하였다. gcc-arm-none-eabi-6-2017-q2-update 컴파일러를 사용하였으며, 본 논문에서는 최적화를 적용하지 않은 -O0 옵션에 대한 실험 결과를 보인다. [6]에서 기본 옵션인 크기 최적화 -Os에 대해서도 100%의 성공률을 가짐을 확인할 수 있다. 부채널 신호 계측 장비의 특성에 따라 HW 값이 클수록 전력 소비 파형 값이 작게 계측된다고 가정한다[5].

3.1. 윈도우 크기 $w=2$

윈도우 크기 $w=2$일 때 OpenSSL 상수 시간 메모리 접근 알고리즘은 8비트 프로세서에서 아래 수식을 기반으로 동작한다.

$$out| = (A[i] \wedge mask_i)$$

$$mask_i = \begin{cases} 0xff, \text{if } wvalue = i \\ 0x00, \text{if } wvalue \neq i \end{cases}$$

($\wedge$=&, bitwise AND, $0 \le i < 2^w = 4$)

$w=2$이므로 총 4개의 사전 연산 값이 배열 A에 저장되어 있으며, 비밀 윈도우 값 $wvalue$에 따라 한 가지를 선택한다. [그림 2]는 $wvalue$값에 따른 out 계산 과정이다. $out=A[wvalue]$, 이지만 항상 배열 A의 모든 요소에 접근하여 [그림 3]과 같이 연산 시간과 패턴이 일정하여 시간 공격 및 단순 전력 분석에 안전하다. $mask_i$값은 $wvalue$값에 따라 결정되며, 배열에서 정확한 위치의 값을 참조하기 위해 사용된다. 따라서 각 $mask_i$값을 결정하거나 참조할 때 발생하는 소비 전력 특성을 정리하면 아래 [특성 1]과 같다.

[특성 1] $wvalue$값이 i일 때 $mask_i=0xff$이므로 $mask_i$의 HW 값인 8에 비례하는 전력 소비가 발생한다. 반면, $wvalue$값이 i가 아닐 때 $mask_i=0x00$이므로 0에 비례하는 전력 소비가 발생한다.

[그림 3]에 강조된 부분은 각 $mask_i$에 대한 PoI로, 해당 영역에서 [그림 2]의 set에 따른 $mask_i$의 소비 전력 파형 분포는 [그림 4]와 같다. $wvalue=0$일 때 $mask_0=0xff$이고 $wvalue=i\,(1\le i<4)$일 때 $mask_0=0x00$이므로 [특성 1]에 따라 $mask_0$에 대한 PoI에서 set 0은 8에 비례하는 전력 소비가 발생하고, set 1~3은 0에 비례하는 전력 소비가 발생한다. 따라서 [그림 4] (a)와 같이 $mask_0$에 대한 PoI에서 set 1~3의 분포는 서로 유사하며 set 0의 분포와 다르다.

[그림 4]와 같이 $mask_i$값에 따른 두 분포의 차이가 명확히 존재하기 때문에 군집화를 통해 PoI 집합을 두 개의 그룹 G_1, G_2로 분류 오류 없이 나눌 수 있다. 따라서 단일 파형 공격으로 비밀키를 100% 성공률로 찾을 수 있다.

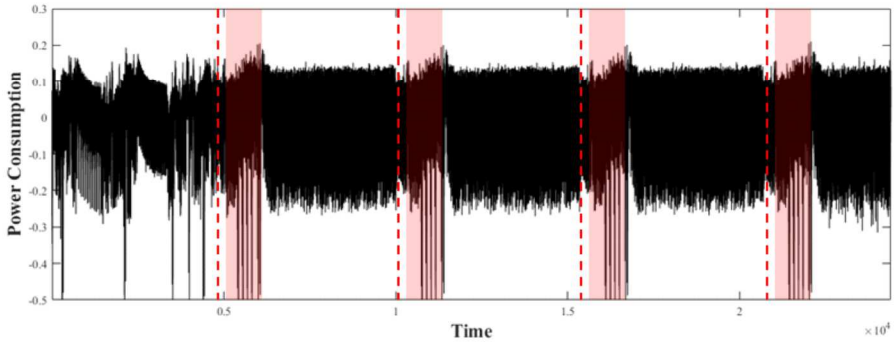
3.2. 윈도우 크기 $w=6$

윈도우 크기 $w=6$일 때 OpenSSL 상수 시간 메모리 접근 알고리즘은 8비트 프로세서에서 아래 수식을 기반으로 동작한다.

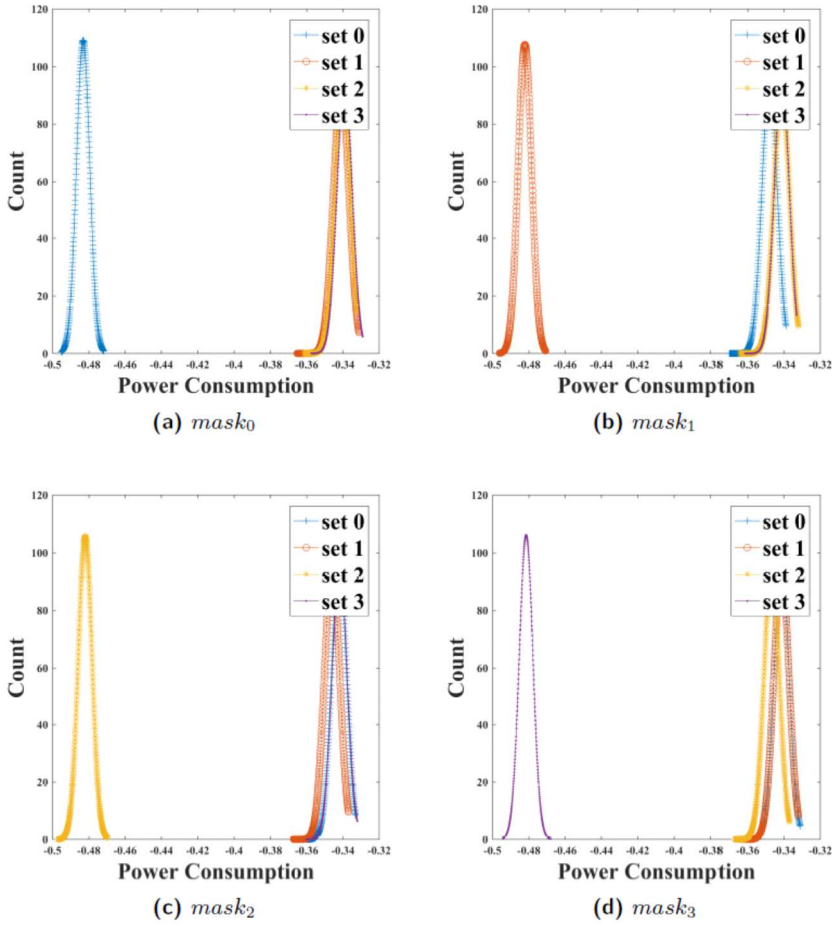
$$\begin{aligned} out| = &\ (A[0x00+j]\wedge mask_0)| \\ &(A[0x10+j]\wedge mask_1)| \\ &(A[0x20+j]\wedge mask_2)| \\ &(A[0x30+j]\wedge mask_3)| \\ &\wedge MASK_j \end{aligned}$$

set	Refer to precomputed value
0	out = (A[0] & 0xff) | (A[1] & 0x00) | (A[2] & 0x00) | (A[3] & 0x00)
1	out = (A[0] & 0x00) | (A[1] & 0xff) | (A[2] & 0x00) | (A[3] & 0x00)
2	out = (A[0] & 0x00) | (A[1] & 0x00) | (A[2] & 0xff) | (A[3] & 0x00)
3	out = (A[0] & 0x00) | (A[1] & 0x00) | (A[2] & 0x00) | (A[3] & 0xff)

〔그림 2〕 OpenSSL 상수 시간 메모리 접근 알고리즘 ($w=2$, 8비트 프로세서)



〔그림 3〕 OpenSSL 상수 시간 메모리 접근 알고리즘 소비 전력 파형 ($w=2$, 8비트 프로세서, -O0)



(a) $mask_0$ (b) $mask_1$

(c) $mask_2$ (d) $mask_3$

〔그림 4〕 $wvalue$값에 따른 소비 전력 파형 분포 ($w=2$, 8비트 프로세서, -O0)

$$mask_i = \begin{cases} 0xff, \text{if}\,(wvalue \wedge 0x30) = i \ll 4 \\ 0x00, \text{if}\,(wvalue \wedge 0x30) \neq i \ll 4 \end{cases}$$

$$MASK_j = \begin{cases} 0xff, \text{if}\,(wvalue \wedge 0x0f) = j \\ 0x00, \text{if}\,(wvalue \wedge 0x0f) \neq j \end{cases}$$

($\wedge$=&, bitwise AND,
$0 \le i < 2^{w-4} = 4,\ 0 \le j < 2^{w-2} = 16$)

$w=6$이므로 총 64개의 사전 연산 값이 배열 A에 저장되어 있으며, 비밀 윈도우 값 $wvalue$에 따라 한 가지를 선택한다. [그림 5]는 $wvalue$값에 따른 out 계산 과정이며, 예로 $wvalue=0x07$, $wvalue=0x2d$ 일 때를 보인다. $out=A[wvalue]$, 이지만 항상 배열 A의 모든 요소에 접근하기 때문에 연산 시간과 패턴이 일정하여 시간 공격 및 단순 전력 분석에 안전하다.

$mask_i$값은 $wvalue \wedge 0x30$값에 따라 결정되고, $MASK_j$값은 $wvalue \wedge 0x0f$값에 따라 결정되며, 배열에서 정확한 위치의 값을 참조하기 위해 사용된다. 각 $mask_i$값과 $MASK_j$값을 결정하거나 참조할 때 발생하는 소비 전력 특성을 정리하면 아래 [특성 2], [특성 3]과 같다.

〔**특성 2**〕 $(wvalue \wedge 0x30)$값이 $i \ll 4$일 때 $mask_i$ 값이 $0xff$이므로 $mask_i$의 HW 값인 8에 비례하는 전력 소비가 발생한다. 반면에, $(wvalue \wedge 0x30)$값이 $i \ll 4$가 아닐 때 $mask_i=0x00$이므로 0에 비례하는 전력 소비가 발생한다.

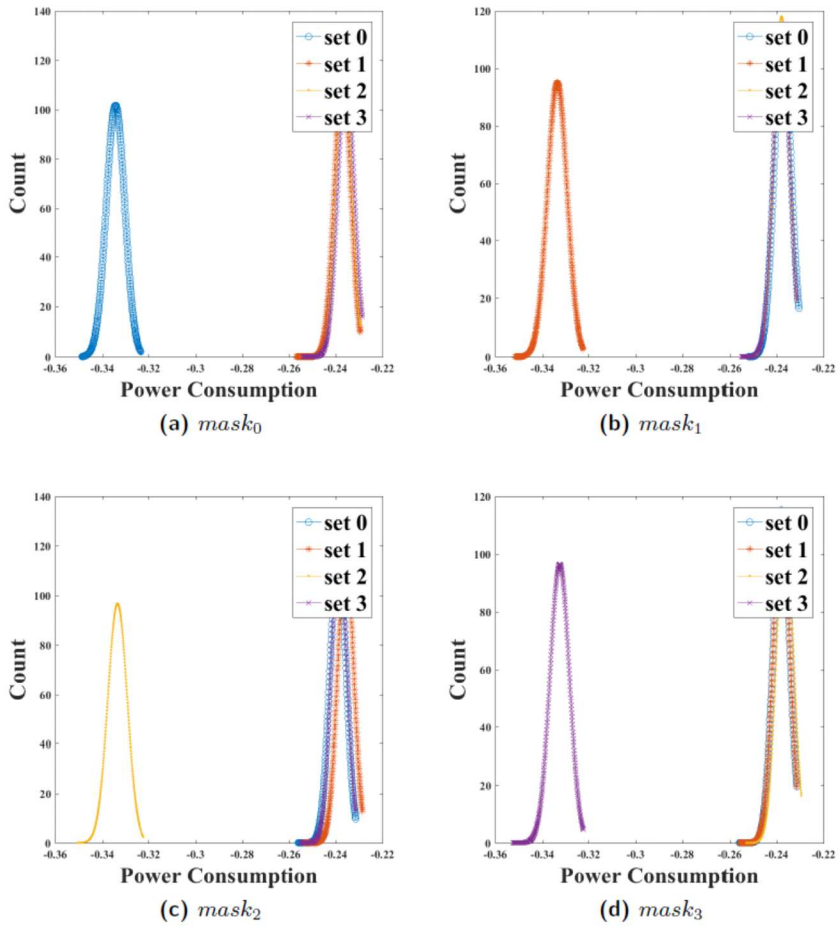
〔**특성 3**〕 $(wvalue \wedge 0x0f)$값이 j일 때 $MASK_j$ 값이 $0xff$이므로 $MASK_j$의 HW 값인 8에 비례하는 전력 소비가 발생한다. 반면에, $(wvalue \wedge 0x0f)$값이 j가 아닐 때 $MASK_j=0x00$이므로 0에 비례하는 전력 소비가 발생한다.

$(wvalue \wedge 0x30)$값에 따른 $mask_i$의 소비 전력 파형 분포는 [그림 6]과 같고, [특성 2]에 따라 $mask_i$값

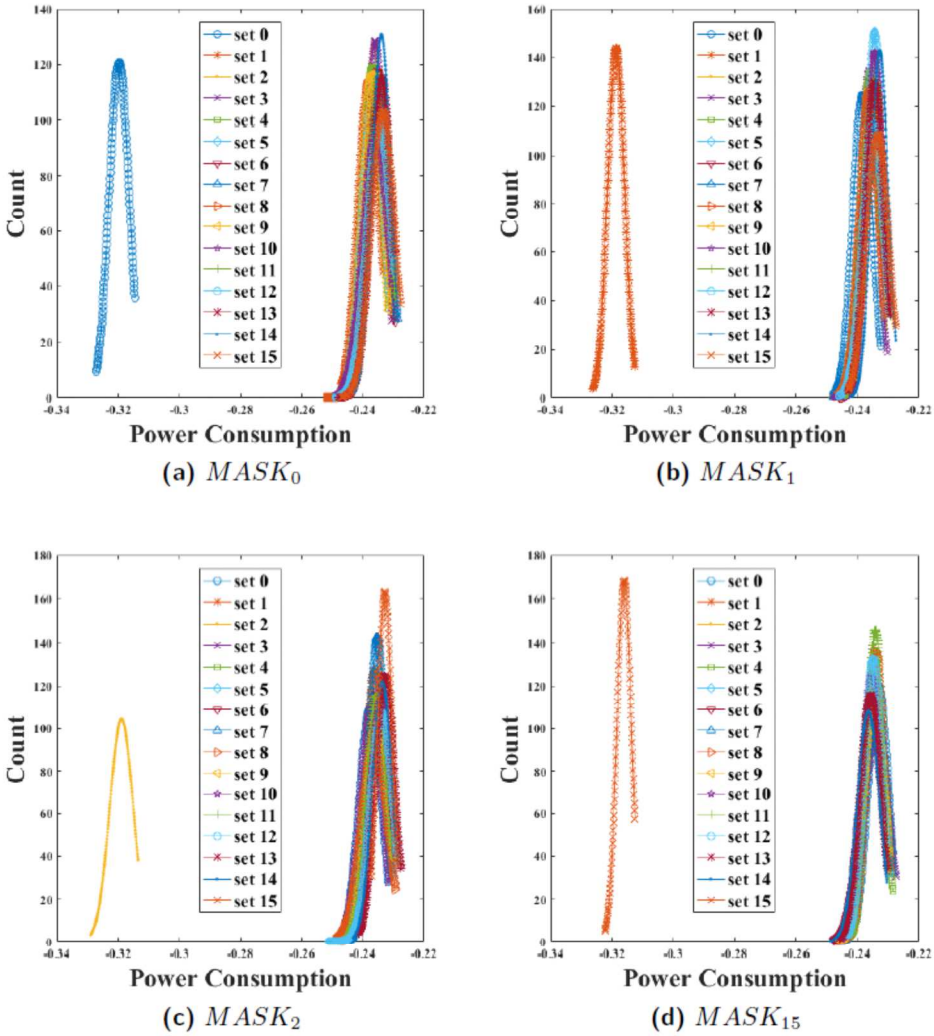
0x07		Refer to precomputed value
0x0 *	0	$out = (A[0x00] \,\&\, 0xff) \mid (A[0x10] \,\&\, 0x00) \mid (A[0x20] \,\&\, 0x00) \mid (A[0x30] \,\&\, 0x00) \,\&\, 0x00$
	1	$out = (A[0x01] \,\&\, 0xff) \mid (A[0x11] \,\&\, 0x00) \mid (A[0x21] \,\&\, 0x00) \mid (A[0x31] \,\&\, 0x00) \,\&\, 0x00$
	⋮	⋮
	7	$out = (A[0x07] \,\&\, 0xff) \mid (A[0x17] \,\&\, 0x00) \mid (A[0x27] \,\&\, 0x00) \mid (A[0x37] \,\&\, 0x00) \,\&\, 0xff$
	⋮	⋮
	f	$out = (A[0x0f] \,\&\, 0xff) \mid (A[0x1f] \,\&\, 0x00) \mid (A[0x2f] \,\&\, 0x00) \mid (A[0x3f] \,\&\, 0x00) \,\&\, 0x00$

0x2d		Refer to precomputed value
0x2 *	0	$out = (A[0x00] \,\&\, 0x00) \mid (A[0x10] \,\&\, 0x00) \mid (A[0x20] \,\&\, 0xff) \mid (A[0x30] \,\&\, 0x00) \,\&\, 0x00$
	1	$out = (A[0x01] \,\&\, 0x00) \mid (A[0x11] \,\&\, 0x00) \mid (A[0x21] \,\&\, 0xff) \mid (A[0x31] \,\&\, 0x00) \,\&\, 0x00$
	⋮	⋮
	d	$out = (A[0x0d] \,\&\, 0x00) \mid (A[0x1d] \,\&\, 0x00) \mid (A[0x2d] \,\&\, 0xff) \mid (A[0x3d] \,\&\, 0x00) \,\&\, 0xff$
	⋮	⋮
	f	$out = (A[0x0f] \,\&\, 0x00) \mid (A[0x1f] \,\&\, 0x00) \mid (A[0x2f] \,\&\, 0xff) \mid (A[0x3f] \,\&\, 0x00) \,\&\, 0x00$

〔그림 5〕 OpenSSL 상수 시간 메모리 접근 알고리즘 ($w=6$, 8비트 프로세서)



〔그림 6〕 $wvalue \wedge 0x30$값에 따른 소비 전력 파형 분포 ($w=6$, 8비트 프로세서, -O0)



〔그림 7〕 $wvalue \wedge 0x0f$값에 따른 소비 전력 파형 분포 ($w=6$, 8비트 프로세서, -O0)

이 $0xff$일 때와 $0x00$인 경우에 따른 두 분포의 차이가 명확히 존재하기 때문에 군집화를 통해 PoI 집합을 두 개의 그룹 G_1, G_2로 분류 오류 없이 나눌 수 있다.

$(wvalue \wedge 0x0f)$값에 따른 $MASK_j$의 소비 전력 파형 분포는 [그림 7]과 같다. $(wvalue \wedge 0x0f) = 0$일 때 $MASK_0 = 0xff$이고 $(wvalue \wedge 0x0f) = j$일 때 $(1 \le j < 16)$ $MASK_0 = 0x00$이므로 [특성 3]에 따라 $MASK_0$에 대한 PoI에서 set 0은 8에 비례하는 전력 소비가 발생하고, set 1~15는 0에 비례하는 전력 소비가 발생한다. 따라서 [그림 4] (a)와 같이 $MASK_0$에 대한 PoI에서 set 1~15의 분포는 서로 유사하며 set 0의 분포와 다르다.

따라서 단일 파형 공격으로 $(wvalue \wedge 0x30)$값과 $(wvalue \wedge 0x0f)$값을 100% 성공률로 찾을 수 있다. 즉, 비밀키를 100% 성공률로 찾을 수 있다.

Ⅳ. 양자 내성 암호 알고리즘에 대한 민감 정보 종속 공격 [6,9,10]

대칭키 암호 알고리즘을 기반으로 두 사용자가 안전한 암호 통신을 위해서는 사전에 비밀키를 공유해야 한다. 이때 공개키 암호 알고리즘 기반 비밀키 교환 체계(KEM, key encapsulation mechanism)가 많이 사용되고 있다. NIST에서 양자 컴퓨팅 시대를 대비하여 2016년도부터 진행된 PQC 표준화 공모 사업의 3라운드 후보가 2020년 7월 22일에 발표되었으며, 7개의 최종 후보와 8개의 대안 후보가 있다. 총 18개의 후보 중 9개가 KEM이며, 그중 5개가 격자 기반 KEM, 3개가 부호 기반 KEM이다.

〔표 1〕 컴파일러 옵션 : 최적화 수준

최적화 수준	설명
-O0	최적화 없음
-O1	속도 최적화(낮음)
-O2	속도 최적화(중간)
-O3	속도 최적화(높음)
-Os	크기 최적화

만약 제 3자가 KEM의 취약점을 이용하여 비밀키를 획득한다면, 두 사용자간 전송되는 모든 메시지를 도청할 수 있다. 이에 부채널 분석을 기반으로 비밀키를 획득하기 위한 공격들이 제시되고 있다 [3-5, 7-11]. 특히 KEM의 캡슐화 과정은 비밀키 유도를 위한 임의의 비밀 메시지를 공개키로 암호화하기 때문에 비밀 메시지 복구 시 단일 파형만을 공격에 사용 가능하다. 주로 곱셈 연산을 대상으로 하는 기존 연구[3,4]와 달리 2020년 Amit 등은 NewHope 캡슐화 과정의 메시지 인코딩 연산을 공격 대상으로 하여 비밀 메시지를 단일 파형으로 복구하는 공격을 처음 제시하였다[7].

[9]에서는 NIST PQC 표준화 3라운드 후보 격자 기반 KEM NTRU Prime의 한 종류인 LPRime의 디캡슐화 과정의 메시지 디코딩 연산을 공격 대상으로 하여 비밀 메시지를 단일 파형으로 복구하는 공격을 제시하였다. 실험을 통해 최적화 옵션에 상관없이 100%의 성공률로 비밀 메시지를 복구할 수 있음을 보였다. [8]에서는 NIST PQC 표준화 3라운드 후보 중 5개의 격자 기반 KEM의 캡슐화 과정의 메시지 인코딩 연산을 공격 대상으로 하였으며 실험을 통해 CRYSTALS-KYBER, SABER의 경우 최적화 옵션에 상관없이 100%의 성공률로 비밀 메시지를 복구할 수 있음을 보였다. FrodoKEM은 79% 이상, NTRU PRime과 NTRU는 96% 이상 비밀 메시지를 복구할 수 있었다. 더불어 [9,10]에서는 셔플링과 마스킹을 적용하여 논문에서 제안한 단일 파형 공격 복잡도를 높이는 대응기법을 권고하였다.

[6]에서는 상수 시간 곱셈을 사용하는 준순환 부호 기반 암호 알고리즘에 대한 다중 및 단일 파형 공격을 제안하였다. 시간 공격에 안전한 신드롬 계산을 위해 Chou가 제안한 상수 곱셈 알고리즘[1]이 공격 대상이다. Rossi 등이 제안한 차분 전력 분석 공격[2]과 달리 [6]에서 제안하는 다중 파형 공격은 부채널 정보만을 이용하여 후보 없이 정확한 비밀 값의 해독이 가능하다. 따라서 프로세서 연산단위에 상관없이 실현 가능한 공격이다. 그리고 [6]에서 제안하는 단일 파형 공격은 마스킹과 같은 차분 전력 분석 대응기법이 적용되어 있거나, 암호 체계가 임시 키를 사용하는 제한적인 경우에도 적용 가능하다. NIST PQC 표준화 2라운드 후보 알고리즘 중 준순환 부호 기반 암호 알고리즘인 LEDACrypt 및 BIKE의 신드롬 연산을 시간 공격 및 차분 전력 분석에 안전하게 구현하기 위해서 Chou가 제안한 상수 시간 곱셈 알고리즘 및 Rossi 등이 제안한 차분 전력 분석 대응기법을 적용하는 경우 [6]에서 제시하는 공격을 통해 비밀 정보 해독이 가능하다(현재 BIKE만 3라운드 후보로 선정되었다).

본 논문에서는 [9]의 공격 결과를 간략히 소개한다.

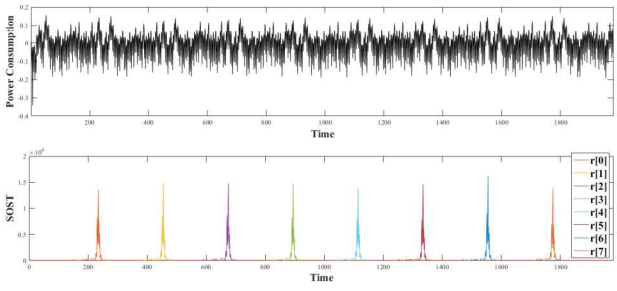
4.1. NTRU LPRime의 메시지 디코딩 연산에 대한 단일 파형 공격

[9]에서는 NTRU LPRime의 메시지 디코딩 연산에 대한 단일 파형 공격을 제시하였다. 공격 위치는 l비트 비밀 메시지 r의 각 비트 값 r_i을 계산하는 위치이다.

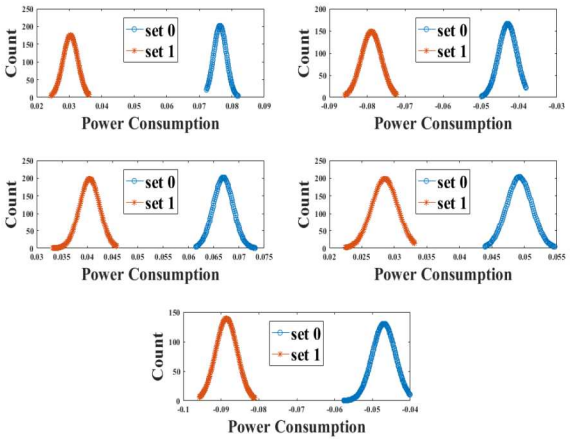
int16_negative_mask($r'[i]$)$=0xFFFF$이면 $r_i=1$이고, int16_negative_mask($r'[i]$)$=0x0000$이면 $r_i=0$이다. 따라서 비밀 메시지 비트 r_i값에 따라 발생하는 전력 소비 특성을 정리하면 다음과 같다 $(0 \le i < l)$.

〔**특성 4**〕 $r_i=1$일 때 중간값 $0xFFFF$의 HW값인 16에 비례한 전력 소비가 발생하고, $r_i=0$일 때 중간값 $0x0000$의 HW값인 0에 비례하는 전력 소비가 발생한다.

32비트 ARM Cortex-M4 프로세서를 탑재한



〔그림 8〕 -O3 전력 소비 파형 (상), SOST 값 (하)



〔그림 9〕 r_0 PoI에서의 소비 전력 분포 (-O0, -O1, -O2, -O3, -Os/ set 0, 1은 각각 G_1, G_2)

ChipWhisperer UFO STM32F3보드에서 알고리즘이 동작할 때 29.54MS/s으로 수집한 전력 파형에 대한 실험 결과를 보였다. gcc-arm-none-eabi-6-2017-q2-update 컴파일러를 사용하였으며, [표 2]과 같은 컴파일러 옵션에 따른 실험 결과를 비교한다.

PoI 식별을 위해 -O3 최적화 옵션으로 500개의 파형을 수집하여 SOST를 계산한 결과는 [그림 8]과 같다. SOST 값이 높은 점을 각 r_i에 대한 PoI로 선택하며, 비밀 메시지 비트 r_i를 계산하기 위해 l번의 동일 연산이 수행되므로 PoI가 규칙적으로 발생한 것을 알 수 있다.

최적화 수준에 따른 PoI에서의 소비 전력 분포는 [그림 9]와 같으며, 두 집합의 평균 및 평균의 차는 [표 2]와 같다. 최적화를 적용하지 않은 -O0일 때 두 집합의 평균의 차가 가장 크며, 속도 최적화가 가장 강하게 적용된 -O3일 때 두 집합의 평균의 차가 가장 작다.

[그림 9]에서와 같이 PoI에서 r_i 값에 따른 두 분포의 차이가 명확히 존재하기 때문에 군집화를 통해 PoI 집합을 두 개의 그룹 G_1, G_2로 분류 오류 없이 나눌 수 있다. 즉, 최적화 수준에 관계없이 100% 성공률로 비밀 메시지 r을 획득할 수 있다. 따라서 획득한 r과 공개된 정보를 이용하여 두 사용자간 공유된 비밀키 K를 계산할 수 있다.

〔표 2〕 두 집합 평균 및 평균의 차

최적화 수준	$E(G_1)$	$E(G_2)$	$\lvert E(G_1)-E(G_2)\rvert$
-O0	7.6411e-02	3.0403e-02	4.6008e-02
-O1	-4.2961e-02	-7.8910e-02	3.5949e-02
-O2	6.6894e-02	4.0483e-02	2.6411e-02
-O3	4.9231e-02	2.8530e-02	2.0701e-02
-Os	-4.6974e-02	-8.8565e-02	4.1591e-02

Ⅴ. 대칭키 암호 알고리즘 AES에 대한 민감 정보 종속 공격 [6]

AES 암호 알고리즘의 MixColums 함수 출력 값 계산 시 4개의 항을 갖는 두 개의 3차 다항식 $a(x)$와 $b(x)$의 모듈러 $m'(x)=x^4+1$ 곱셈 연산이 수행된다. 이때, 다항식의 각 계수는 $GF(2^8)$의 원소이며, $a(x)=3x^3+x^2+x+2$이다($b(x)$의 각 계수는 S-Box 출력 값).

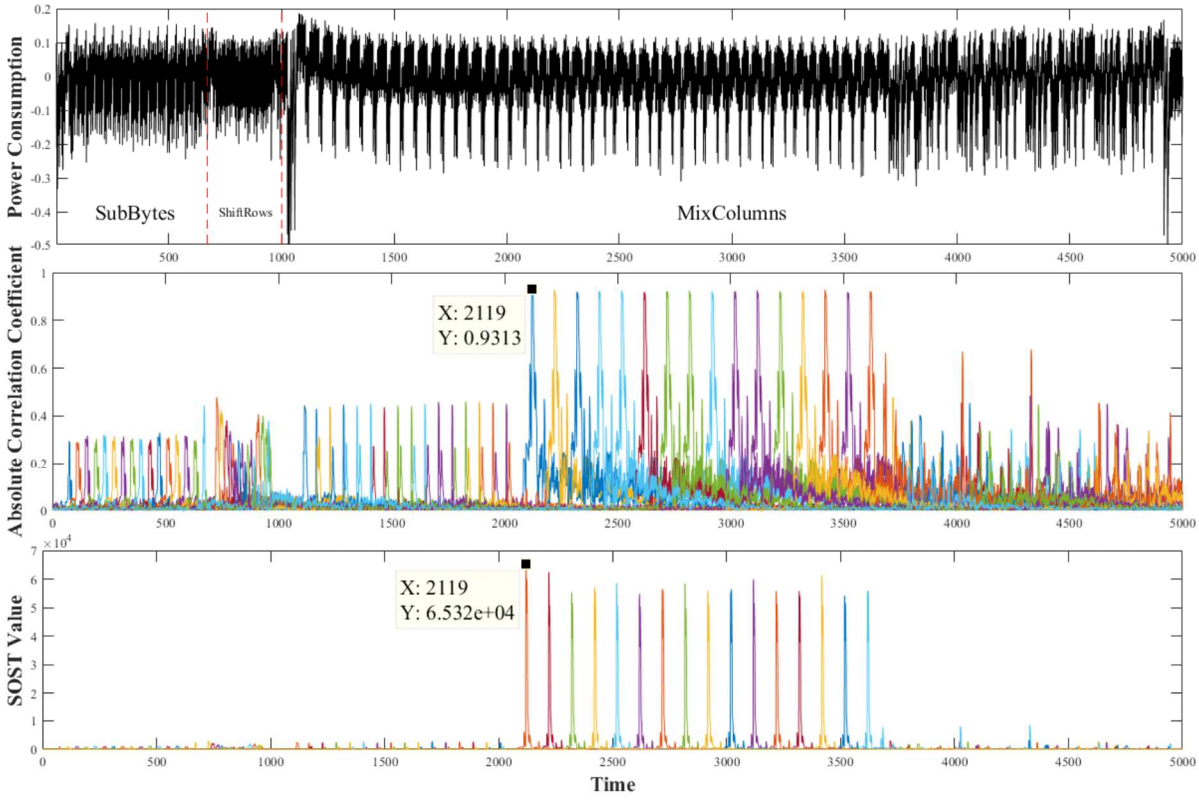
따라서 $b(x)$의 계수 $s\,INGF(2^8)$에 대한 갈루아 체 $GF(2^8)\cong GF(2)[x]/(x^8+x^4+x^3+x+1)$에서의 2배 연산(xtimes)이 수행되며 아래와 같다.

$$xtimes(s)=(s\ll 1)\oplus(0x1b\times b_7)$$
$$=\begin{cases} s\ll 1 & ,\mathrm{if}\, b_7=0 \\ (s\ll 1)\oplus 0x1b & ,\mathrm{if}\, b_7=1 \end{cases}$$

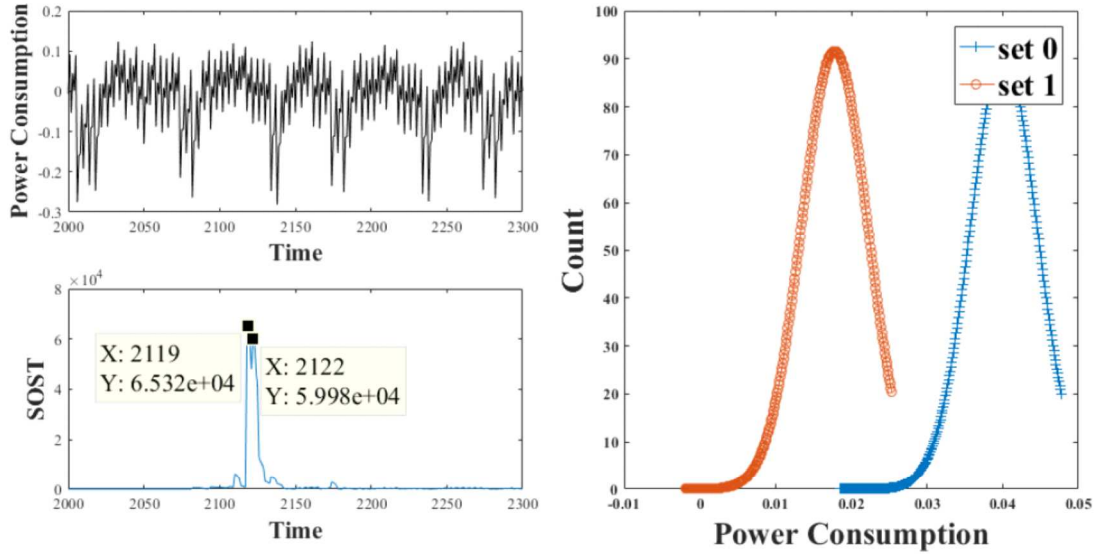
피연산자 $s=(b_7,b_6,\cdots,b_0)_2$의 최상위 비트(MSB, most significant bit) b_7에 따라 xtimes 연산 결과가 달라지므로, xtimes 연산 수행 시 s의 MSB b_7을 추출하여 레지스터에 저장하고 읽는 연산이 필요하다. 따라서 MixColums 연산 시 발생하는 소비 전력 특성을 정리하면 아래 [특성 5]와 같다.

〔**특성 5**〕 $b_7=1$일 때 중간값 $0x1b$의 HW값인 4에 비례한 전력 소비가 발생하고, $b_7=0$일 때 0에 비례하는 전력 소비가 발생한다.

소비 전력 파형은 8비트 AVR 프로세서를 탑재한 ChipWhisperer-Lite XMEGA 보드를 이용하여 29.54MS/s으로 수집하였다. gcc-arm-none-eabi-6-2017-q2-update 컴파일러를 사용하였으며, 기본 옵션인 크기 최적화 -Os에 대한 실험 결과를 보인다. EM(expectation-maximization) 군집 알고리즘을 이용해 [그림 10]과 [그림 11] (좌)를 기반으로 선택한 PoI 집합을 99.51% 성공률로 두 개의 그룹 G_1, G_2로 분류할 수 있다. 두 그룹에 속하는 원소들의 HW 차이는 4



〔그림 10〕 -Os 소비 전력 파형 (상), 16개의 S-Bos 출력 값과의 절대 상관 계수 값 (중), 16개의 S-Box 출력 값의 MSB에 따른 SOST 값 (하)



〔그림 11〕 첫 번째 S-Box 출력 값의 MSB에 따른 SOST 값 (좌), PoI 2119에서의 소비 전력 분포 (우)

로 3, 4장에서 각각 8, 16이었던 것보다 작다. 이에 3, 4장에서 100% 성공률로 오류 없이 군집화 할 수 있는 것과 달리 오분류된 비트가 존재한다. 이는 [그림 11] (우)와 같이 두 분포가 겹치는 부분이 존재하기 때문이다. 따라서 임의의 분류 오류 기준을 적용한 경우 분류 성공률이 84.86%이다.

S-Box 출력의 각 비트에 대해 대수 정규 형식(ANF, algebraic normal form)이 존재하므로, 민감 정보 종속 공격을 기반으로 추출한 16개의 S-Box 출력 값의 MSB 정보를 기반으로 대수 공격 기법을 적용하여 비밀키와 관련된 정보를 찾는 것은 흥미로운 향후 연구 주제 중 하나가 될 것이다.

Ⅵ. 결 론

본 논문에서는 공개키, 양자 내성, 대칭키 암호 알고리즘에 대한 민감 정보 종속 공격 동향을 소개하였다. 암호 알고리즘의 입·출력 값에 대한 정보 없이 부채널 정보만을 이용한 공격이며, 공개키와 양자 내성 알고리즘의 경우 단일 파형으로 100%의 성공률로 민감 정보를 획득할 수 있다. 이는 시간 공격에 안전한 구현을 위해 발생 가능한 두 중간값의 HW 차이가 큰 결정자를 이용하기 때문이다. 즉, 시간 공격에 대응하기 위해 적용한 방법이 더 심각한 보안 위협을 초래할 수 있음을 시사한다. 따라서 결정자를 사용하는 기존 시간 공격 대응기법을 보완한 새로운 대응기법에 대한 연구가 필요하다.

참 고 문 헌

[1] T. Chou. Qcbits: Constant-time small-key code-based cryptography. In Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings, pages 280 – 300, 2016.

[2] M. Rossi, M. Hamburg, M. Hutter, and M. E. Marson. A side-channel assisted cryptanalytic attack against qcbits. In Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings, pages 3 – 23, 2017.

[3] A. Aysu, Y. Tobah, M. Tiwari, A. Gerstlauer and M. Orshansky, “Horizontal side-channel vulnerabilities of post-quantum key exchange protocols”, HOST 2018, pp. 81-88, April, 2018.

[4] J.W. Bos, S. Friedberger, M. Martinoli, E. Oswald and M. Stam, “Assessing the feasibility of single trace power analysis of frodo”, SAC 2018, pp. 216-234, August, 2018.

[5] W.-L. Huang, J.-P. Chen and B.-Y. Yang, “Power Analysis on NTRU Prime”, TCHES 2020, No. 1, pp. 123-151, November, 2019.

[6] 심보연, “민감 정보 종속 공격 및 그 응용”, 국민대학교 일반대학원 2020.

[7] D. Amit, A. Curiger, L. Leuenberger and P. Zbinden, “Defeating newhope with a single trace”, PQCrypto 2020, pp. 189-205, April, 2020.

[8] P. Ravi, S. Bhasin, S.S. Roy and A. Chattopadhyay, “Drop by Drop you break the rock – Exploiting generic vulnerabilities in Lattice-based PKE/KEMs using EM-based Physical Attacks”, ePrint 2020-549.

[9] 심보연, 한재승, 이태호, 김일주, 한동국, “NIST Round 2 후보 격자 기반 KEM NTRU LPRime에 대한 신규 단일 파형 공격”, 한국정보보호학회 하계 학술대회 2020.

[10] B.-Y. Sim, J. Kwon, J. Lee, I.-J. Kim, T. Lee, J. Han, H. Yoon, J. Cho and D.-G. Han, “Single-Trace Attacks on Message Encoding in Lattice-Based KEMs”, IEEE Access 2020, Vol. 8, pp. 183175-183191, October, 2020.

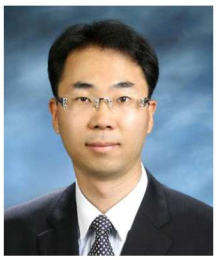
[11] P. Ravi, S. Bhasin, S.S. Roy and A. Chattopadhyay, “On Exploiting Message Leakage in (few) NIST PQC Candidates for Practical Message Recovery and Key Recovery Attacks”, ePrint 2020-1559.

〈저 자 소 개〉

심 보 연 (Bo-Yeon Sim)

정회원

2013년 2월 : 국민대학교 수학과 학사
2015년 2월 : 국민대학교 금융정보보안학과 이학석사
2020년 2월 : 국민대학교 수학과 이학박사
2020년 3월~2021년 1월 : 국민대학교 산학협력단 연구교수
2021년 2월~현재 : 한국전자통신연구원 지능화융합연구소 연구원
<관심분야> 공개키 암호 시스템, 부채널 분석 및 대응기법 설계, 경량 저전력 정보보호 기술



한 동 국 (Dong-Guk Han)

정회원

1999년 2월 : 고려대학교 수학과 학사
2002년 2월 : 고려대학교 수학과 이학석사
2005년 2월 : 고려대학교 정보보호대학원 공학박사
2004년 4월~2005년 4월 : 일본 Kyushu Univ. 방문연구원
2005년 4월~2004년 4월 : 일본 Future Univ.-Hakodate Post. Doc.
2006년 6월~2009년 2월 : 한국전자통신연구원 정보보호연구단 선임연구원
2009년 3월~현재 : 국민대학교 정보보안암호수학과 교수
<관심분야> 공개키 암호 시스템 안전성 분석 및 고속 구현, 부채널 분석 및 대응법 설계, IoT 정보보호 기술