Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.3, Jun. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.3.439

8 비트 구현 Ring-LWE 암호시스템의 SPA 취약점 연구*

박 애 선,† 원 유 승, 한 동 국‡
국민대학교 금융정보보안학과

A Study of SPA Vulnerability on 8-bit Implementation of Ring-LWE Cryptosystem*

Aesun Park,† Yoo-Seung Won, Dong-Guk Han‡
Dept. of Financial Information Security, Kookmin University

요 약

포스트 양자 암호라 할지라도 실제 디바이스에 이를 적용 할 때 부채널 분석 취약점이 존재한다는 것은 이미 알려져 있다. 코드 기반 McEliece 암호와 격자 기반 NTRU 암호에 대한 부채널 분석 연구 및 대응책 연구는 많이 이루어지고 있으나, ring-LWE 암호에 대한 부채널 분석 연구는 아직 미비하다. 이에 본 논문은 8비트 디바이스에서 ring-LWE 기반 암호가 동작할 때 적용 가능한 선택 암호문 SPA 공격을 제안한다. 제안하는 공격은 $\lceil \log_2 q \rceil$ 개의 파형으로 비밀키를 복구 할 수 있다. q는 보안 레벨과 관련된 파라미터로 128비트 또는 256비트의 보안 레벨을 만족하기 위해 각각 7681 또는 12289를 사용한다. 또한, 우리는 실제 디바이스에서 동작되는 ring-LWE 복호화 과정의 모듈러 덧셈에서 비밀키를 드러낼 수 있는 취약점이 존재함을 실험을 통해 보이고, 공격 시간 단축을 위한 두 벡터의 유사도 측정 방법을 이용한 공격에 대해 논한다.

ABSTRACT

It is news from nowhere that post-quantum cryptography has side-channel analysis vulnerability. Side-channel analysis attack method and countermeasures for code-based McEliece cryptosystem and lattice-based NTRU cryptosystem have been investigated. Unfortunately, the investigation of the ring-LWE cryptosystem in terms of side-channel analysis is as yet insufficient. In this paper, we propose a chosen ciphertext simple power analysis attack that can be applied when ring-LWE cryptography operates on 8-bit devices. Our proposed attack can recover the key only with $\lceil \log_2 q \rceil$ traces. q is a parameter related to the security level. It is used 7681 and 12289 to match the common 128 and 256-bit security levels, respectively. We identify the vulnerability through experiment that can reveal the secret key in modular add while the ring-LWE decryption performed on real 8-bit devices. We also discuss the attack that uses a similarity measurement method for two vectors to reduce attack time.

Keywords: ring-LWE cryptosystem, simple power analysis, post-quantum cryptography, side-channel analysis

Received(03. 15. 2017), Modified(1st: 05. 02. 2017), Accepted(05. 04. 2017)
* 본 논문은 2016년 동계학술대회에 발표한 우수논문을 개선 및 확장한 것임
* 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00520,(ICT 기초연구실) SCR-Friendly 대칭키 암호 및 응용 모드 개발)
† 주저자, aesons@kookmin.ac.kr
‡ 교신저자, christa@kookmin.ac.kr(Corresponding author)

I. 서 론

현재 널리 사용되고 있는 공개키 암호시스템의 안전성은 큰 수의 소인수분해, Z_p^* 또는 유한체에서의 이산 로그 문제 등 수학적 난제를 기반으로 한다. 그러나 양자컴퓨팅 기술을 이용하면 현재의 컴퓨팅 능력으로는 오랜 시간이 걸려 해결이 어려운 문제를 쉽고 빠르게 풀 수 있을 뿐만 아니라, 1995년 Shor는 소인수분해 문제와 이산 로그 문제를 양자 컴퓨터상에서 다항식 시간에 풀기 위한 알고리즘을 제시했다[1]. 제안된 알고리즘으로 인해 양자 컴퓨터의 존재는 RSA, Diffie-Hellman 키 교환, DSA, ECDSA 등 현재 널리 사용되는 암호를 무력화 시키는 현실적인 위협이 될 수 있다. 따라서 이러한 위협에 대한 대안이 될 수 있는 포스트 양자 암호시스템에 대한 요구가 증가되고 있다.

이론적으로 안전한 암호 알고리즘이라도 실제 디바이스에 적용되었을 때 소비 전력 및 전자파 등 부가적인 정보를 발생한다. 이러한 부가적인 정보들의 노출을 이용하여 암호화 키 같은 비밀 정보를 알아내는 부채널 분석(side-channel analysis)이 1996년 Kocher에 의해 제안된 이후, 많은 암호 알고리즘에 대한 부채널 분석 연구가 이루어져왔다[2]. 특히 전력 분석(power analysis)은 디바이스의 암호 알고리즘이 실행되는 동안의 소비 전력을 분석하여 비밀 정보를 얻어 내는 것으로 효율적인 분석 방법으로 알려져 있다. 전력 분석 공격은 디바이스가 '0' 또는 '1'을 처리하는데 소비되는 전력이 서로 다르다는 점을 이용한다.

전력 분석은 단지 몇 번의 알고리즘이 실행되는 동안 발생하는 전력 신호를 관찰하여 비밀키 값을 유도해 내는 단순전력분석(SPA, Simple Power Analysis)과 수차례 알고리즘이 반복 실행되는 동안 수집된 전력 신호들을 두 집단으로 나누어 분석하는 차분전력분석(DPA, Differential Power Analysis)이 존재한다[3].

포스트 양자 암호 알고리즘이라 할지라도 실제 디바이스에 적용될 때 부채널 분석 취약점이 존재한다. 특히 코드 기반 McEliece 암호와 격자 기반 NTRU 암호에 대한 부채널 분석 연구 및 대응책 연구는 많이 이루어졌다[4,5]. 그러나 최근 ring-LWE 문제 기반 공개키 암호 알고리즘의 관심이 높아졌으나, 대부분 효과적인 구현을 위한 연구가 많이 진행되었고 1차 DPA 공격에 대한 대응법을 제안한 2가지 논문만 존재 할 뿐 그 연구는 미비하다[13,14].

또한 인터넷이 스마트 기기 등에 널리 사용되고 있고 이러한 기기들은 컴퓨팅 전력, 메모리 용량 등이 제한된다. 뿐만 아니라 사물인터넷의 발달로 인해 이러한 장치들의 사용은 날로 증가하고 있다. 이러한 이유로 8 비트 플랫폼에서 사용되는 포스트 양자 암호 기술 등이 제안되었다[9].

본 논문에서는 효과적인 구현을 위해 NTT(Number Theoretic Transform)를 적용한 ring-LWE 암호 알고리즘의 8 비트 구현시 복호화 단계에서 나타나는 취약점을 이용한 선택 암호문 SPA 공격을 제안한다. 또한 실제 디바이스에서의 실험을 통해 복호화 단계에서 사용되는 모듈러 덧셈에 취약점이 존재함을 보이고, 공격의 효율성을 높이기 위해 유사도 측정 방법을 이용한 취약점 확인 방법에 대해 논한다.

본 논문의 구성은 2장에서 ring-LWE 암호시스템에 대해 설명 후, 3장에서 제안하는 선택 암호문 SPA 공격 방법을 서술한다. 4장에서는 실제 획득한 전력에서 모듈러 연산(modular arithmetic) 발생 여부의 차이를 구별할 수 있음을 실험을 통해 보이고, 공격 시간 단축을 위한 방법에 대해 언급한 후 결론을 맺는다.

II. Ring-LWE 암호시스템

2.1 기호 정의

암호시스템 설명에 앞서 본 논문에서는 다음과 같은 기호를 사용한다. 영문 소문자는(예: r_i, c_i, a) 다항식을 의미한다. 단, 소문자 m은 메시지로 비트 문자열을 의미하며 q는 정수이다. 다항식은 다음의 식과 같이 표현된다.

$$r_2(x) = \sum_{j=0}^{n-1} r_2[j]x^j = r_2[0]x^0 + r_2[1]x^1 + \cdots + r_2[n-1]x^{n-1}$$

여기에서 $r_2[j]$는 $r_2(x)$의 x^j의 계수를 의미하며 Z_q의 원소이다. 또한 본 논문에서 사용하는 다항식은 환 $R_q = Z_q[x]/(f(x))$의 원소이며, 여기에서 $f(x)$는 n차 기약 다항식이다. 또한 기호 • 는 두 다항식의 계수별 곱을 의미한다.

2.2 Ring-LWE 기반 암호시스템

Ring-LWE 기반 암·복호화는 키생성, 암호화, 복호화 세 단계로 구분된다[6]. Ring-LWE 암호시스템의 효과적인 구현을 위해 2014년 NTT를 이용한 방법이 Roy 등에 의해 제안되었다[7]. 다음은 NTT를 이용한 ring-LWE 기반 암호 알고리즘의 키생성 및 암·복호화 과정을 설명한 것이다. 틸더(~) 표시의 영문 소문자는($\tilde{r}$) 다항식(r)의 NTT 적용 결과 값이다.

1. $KeyGen(a)$: 두 개의 오류 다항식 r_1, r_2를 이산 가우시안 분포에서 랜덤하게 추출 후 a, r_1, r_2에 NTT 알고리즘을 적용한다. 여기에서 a는 랜덤 다항식으로 공개 가능한 다항식이다.

$$\widetilde{r_1} = NTT(r_1)\ ,\ \widetilde{r_2} = NTT(r_2)\ ,\ \tilde{a} = NTT(a)$$

이후 공개키로 사용될 다항식 $\tilde{p} = \widetilde{r_1} - \tilde{a} \bullet \widetilde{r_2}$를 계산한다. 개인키는 $\widetilde{r_2}$이며, 다항식 쌍 $(\tilde{a}, \tilde{p})$를 공개키로 사용한다.

2. $Enc((\tilde{a},\tilde{p}),m)$: n비트 문자열 메시지 m의 암호화 첫 번째 단계는 메시지 m을 R_q의 원소 $(=\overline{m})$로 인코딩하는 것이다. R_q의 원소로 인코딩된 메시지는 아래의 과정을 거쳐 암호문 $(\widetilde{c_1},\widetilde{c_2})$로 계산된다. 이산 가우시안 분포에서 오류 다항식 e_1, e_2, e_3를 랜덤(random)하게 추출 후, e_1, e_2에 NTT 알고리즘을 적용한다.

$$\widetilde{e_1} = NTT(e_1)\ ,\ \widetilde{e_2} = NTT(e_2)$$

이후 e_3, 공개키 $(\tilde{a}, \tilde{p})$ 및 인코딩된 메시지 $\overline{m}$을 이용하여 암호문 $(\widetilde{c_1}, \widetilde{c_2})$을 아래와 같이 계산한다.

$$\widetilde{c_1} = \tilde{a} \bullet \widetilde{e_1} + \widetilde{e_2},$$
$$\widetilde{c_2} = \tilde{p} \bullet \widetilde{e_1} + NTT\left(e_3 + \overline{m}\right)$$

3. $Dec((\widetilde{c_1},\widetilde{c_2}),\widetilde{r_2})$: 개인키 $\widetilde{r_2}$를 이용한 메시지 복호화 단계로 아래의 식으로 계산된다.

$$m = th\left(INTT\left(\widetilde{c_1} \bullet \widetilde{r_2} + \widetilde{c_2}\right)\right)$$

여기에서 INTT는 역 NTT를 의미한다. 메시지를 올바르게 복호화하기 위해서는 역 NTT 이후 얻게 되는 $\overline{m}$를 디코딩하여야 한다. 본 논문에서는 디코딩 함수를 [7]에서 사용한 기호(th)를 따르며, 디코딩 함수는 아래와 같이 정의된다.

$$th(x) = \begin{cases} 0 \text{ if } x \in \left\{\left(0, \frac{q}{4}\right) \cup \left(\frac{3q}{4}, q\right)\right\} \\ 1 \text{ if } x \in \left(\frac{q}{4}, \frac{3q}{4}\right) \end{cases}$$

III. 선택 암호문 SPA 공격

본 장에서는 ring-LWE 기반 암호의 복호화 과정에서 비밀키를 찾기 위한 선택 암호문 SPA 공격을 제안한다. 본 공격의 목적은 개인키 r_2를 찾는 것이다. 2장에서 설명한 바와 같이 ring-LWE 암호는 효과적인 구현을 위해 NTT 알고리즘을 사용하므로 제안하는 선택 암호문 SPA 공격은 $\widetilde{r_2} = NTT(r_2)$를 획득 후 역 NTT를 이용해 r_2를 복원한다.

편의성을 위해 이후 부분에서는 c_1, c_2, r_2의 NTT 변환 값을 틸더(~) 기호 제외 후 사용한다.

3.1 공격 지점

Algorithm 1.은 복호화 단계를 간략히 나타낸 알고리즘으로 디코딩 적용 전까지의 흐름을 나타내고 있다. 제안하는 공격 기법은 Algorithm 1.의 7~9 단계의 조건문 존재 여부를 활용하여 수행한다. 두 다항식의 덧셈에서 계수별로 덧셈이 이루어질 때 덧셈의 결과 값에 대한 모듈러 연산 발생 여부 판단 이 가능하다면, 3.2절에서 설명하고 있는 공격을 통해 각 시행에서 비밀키의 후보를 약 $\frac{1}{2}$배 감소시킬 수 있기 때문에 오직 $\lceil \log_2 q \rceil$번의 시행으로 비밀키를 알아낼 수 있다. 여기에서 $\lceil x \rceil$은 천장 함수(ceiling function)로 x보다 작지 않은 최소 정수를 의미하며 q는 보안 레벨과 관련된 파라미터로 128비트 또는

256비트의 보안 레벨을 만족하기 위해 각각 7681 또는 12289를 사용한다. 공격 방법에 대한 자세한 내용은 3.2절에서 설명한다.

Algorithm1. Pseudo-code of Decryption

Input : $(c_1, c_2), r_2$

Output : $\bar{m}$

1. **For** i from 0 to $n-1$ **do**
2. $\quad c_1[i] \leftarrow r_2[i] \cdot c_1[i]$
3. $\quad$ **if** $c_1[i] \geq q$ **then**
4. $\quad\quad c_1[i] \leftarrow c_1[i] \mod q$
5. $\quad$ **end if**
6. $\quad \bar{m}[i] \leftarrow c_1[i] + c_2[i]$
7. $\quad$ **if** $\bar{m}[i] \geq q$ **then**
8. $\quad\quad \bar{m}[i] \leftarrow \bar{m}[i] \mod q$
9. $\quad$ **end if**
10. **end for**

Ring-LWE 암호 알고리즘을 사용 할 때 파라미터 (n, q, σ)는 일반적으로 $\left(256, 7681, \frac{11.31}{\sqrt{2\pi}}\right)$ 또는 $\left(512, 12289, \frac{12.18}{\sqrt{2\pi}}\right)$로 사용하는데, 이는 각각 128비트 또는 256비트 보안 레벨을 만족하는 파라미터이다. 여기에서 n은 다항식 환의 차수를 의미하고, q는 모듈러스(modulus), σ는 오류 다항식을 선택하는 가우시안 분포의 표준 편차를 의미한다[8].

본 논문에서 제안하는 선택 암호문 SPA 공격은 n과 σ에 의존하지 않는다.

3.2 공격 방법

본 절에서는 비밀키의 n개 계수를 모두 찾기 위해 i번째 계수를 타겟으로 제안하는 공격 방법을 자세히 설명한다. 비밀키 r_2를 알아내기 위해 공격자는 선택된 암호문의 입력이 필요한데, 우리는 모든 단계에서 암호문 c_1의 모든 n개의 계수를 1로 고정한다. 또한 암호문 c_2는 이전 단계의 결과에 따라 각 시행 단계별로 다르게 설정한다. r_2의 x^i의 계수 $r_2[i]$를 추측하는 방법은 다음과 같다. 여기에서 $r_2[i]$는 공격자가 모르는 비밀 정보이며, q값은 공개 되어 있는 정보이다.

$c_1[i] = 1$이므로 역 NTT이전 계산은 다음과 같이 이루어진다.

$$c_1[i] \bullet r_2[i] + c_2[i] = r_2[i] + c_2[i]$$

첫 번째 시행에서 $c_2^1[i]$의 값을 $\left\lceil \frac{q-1}{2} \times (1) \right\rceil$로 설정한다. 여기에서 $c_2^t[i]$는 t번째 시행에서의 $c_2[i]$를 의미한다. $r_2[i] + c_2^1[i]$ 값이 모듈러스 q에 대해 모듈러 연산이 발생하지 않는다는 것은 $r_2[i] + c_2^1[i] = r_2[i] + \left\lceil \frac{q-1}{2} \right\rceil$ 값이 모듈러스 q보다 작다는 것을 의미한다. 따라서 $r_2[i]$값은 $\left[0, \frac{q-1}{2}\right]$에 존재함을 알 수 있다. 반대로 모듈러 연산이 발생하는 경우 $r_2[i]$값은 $\left[\frac{q-1}{2}+1, q-1\right]$ 사이에 존재함을 알 수 있다.

예를 들어 $r_2[i] = 2784$(비밀정보)이고 $q = 7681$(공개된 정보)인 경우를 생각해보자. 첫 번째 시행에서 $c_2^1[i]$는 아래와 같이 설정한다.

$$c_2^1[i] = \left\lceil \frac{7681-1}{2} \times 1 \right\rceil = 3840$$

이 경우 $r_2[i] + c_2^1[i] = 2784 + 3840 = 6624$이므로 모듈러 연산이 일어나지 않는다. 그러므로 공격자는 비밀키 $r_2[i]$의 값이 $[0, 3840]$ 사이에 존재함을 알 수 있다. 이로써 공격자는 비밀키의 후보를 약 $\frac{1}{2}$배 감소시킬 수 있다.

만약 첫 번째 시도에서 $r_2[i]$의 값이 $\left[0, \frac{q-1}{2}\right]$ 사이에 존재함을 알았다면, 두 번째 시행에서는 $\left[0, \frac{q-1}{2}\right]$의 중간값을 기준으로 모듈러 연산 발생 여부를 판단하기 위해 $c_2^2[i]$를 아래와 같이 선택한다.

$$c_2^2[i] = \left\lceil \frac{q-1}{2^2} \times (3) \right\rceil$$

이때, 첫 번째 시행과 비슷한 방법으로 $r_2[i] + c_2^2[i]$

값이 모듈러스 q에 대해 모듈러 연산 발생하지 않았다면 $r_2[i]$는 $\left[0, \frac{q-1}{4}\right]$ 사이의 값으로, 모듈러 연산이 발생 했다면 $\left[\frac{q-1}{4}+1, \frac{q-1}{2}\right]$ 사이의 값으로 추정할 수 있다. 반대로 첫 번째 시행에서 알아낸 $r_2[i]$의 정보가 $\left[\frac{q-1}{2}+1, q-1\right]$ 사이에 존재하였다면 $c_2^2[i]$는 아래와 같이 선택한다.

$$c_2^2[i] = \left\lceil \frac{q-1}{2^2} \times (1) \right\rceil$$

마찬가지로 $r_2[i]+c_2^2[i]$ 값의 모듈러 연산이 발생하지 않았다면 $r_2[i]$는 $\left[\frac{q-1}{2}+1, \frac{q-1}{4}\times(3)\right]$ 사이의 값으로, 반대는 $\left[\frac{q-1}{4}\times(3)+1, q-1\right]$ 사이의 값으로 추정할 수 있다. 마지막 시행($\lceil \log_2 q \rceil$ 번째)를 제외한 이후의 모든 시행에서 앞의 방법과 동일한 방법으로 $r_2[i]$의 후보들을 줄여간다.

마지막 $\lceil \log_2 q \rceil$ 번째 시행에서는 이전 단계에서 사용된 $c_2^{\lceil \log_2 q \rceil - 1}[i]$를 사용한다. 이전 단계의 $r_2[i]+c_2^{\lceil \log_2 q \rceil - 1}[i]$ 값에 대해 모듈러 연산이 발생하지 않았다면 이전 단계의 $c_2^{\lceil \log_2 q \rceil - 1}[i]$에 1을 더한 값으로 설정하고, 이전 단계에서 모듈러 연산이 발생했다면 전 단계의 $c_2^{\lceil \log_2 q \rceil - 1}[i]$에 1을 뺀 값으로 설정한다. 이후, 모듈러 연산 발생 여부에 따라 다음과 같이 $r_2[i]$를 추출한다.

$$\text{발생} : r_2[i] = q - c_2^{\lceil \log_2 q \rceil}[i],$$
$$\text{발생 안함} : r_2[i] = q - c_2^{\lceil \log_2 q \rceil}[i] - 1$$

Algorithm 2.는 제안하는 선택 암호문 SPA 공격의 암호문 선택 방법에 대한 흐름을 나타내고 있는 의사코드(pseudocode)이다. Algorithm 2.의 1~3 단계는 선택 암호문 SPA 공격의 첫 번째 시행으로 암호문 (c_1, c_2)의 모든 계수를 $c_1 = 1$, $c_2^1 = \frac{q-1}{2}$로 각각 설정하는 단계이다. 이후 모든 시행의 암호문 c_1은 1이므로 이후 단계에서의 설정은 생략하였다. Algorithm 2.의 5~12 단계는 두 번째 $\lceil \log_2 q \rceil - 1$번째 시행까지의 암호문 c_2^t, $t \in \{2, \lceil \log_2 q \rceil - 1\}$의 설정 방법을 나타내고 있으며, 마지막 시행에서 사용될 암호문은 15~19 단계와 같이 설정하고 이후 모듈러 연산 결과를 이용하여 비밀키를 복원한다.

앞서 언급했던 것처럼 ring-LWE 암호시스템은 효과적인 구현을 위해 NTT를 적용하는데, 이렇게 NTT가 적용된 다항식의 곱은 같은 차수의 계수 곱으로 계산된다. 이것은 모든 계수들은 각각 독립적으로 계산된다는 것을 의미한다. 즉, 비밀 정보 r_2의 모든 n개의 계수를 구하기 위해 n배의 시행을 더 할 필요가 없이 총 $\lceil \log_2 q \rceil$ 번의 시행($q = 7681$인 경우 13번, $q = 12289$인 경우 14번)으로 모든 계수를 알아낼 수 있다.

Algorithm 2. Proposed Chosen Ciphertext SPA Attack

Input : q, n

Output : $r_2[i]$ where $i \in [0, n-1]$

1. **for** i from 0 to $n-1$ **do**
2. $c_1[i] \leftarrow 1, c_2^1[i] \leftarrow \frac{q-1}{2}, k[i] \leftarrow 1$
3. **end for**
4. **for** j from 2 to $\lceil \log_2 q \rceil$ **do**
5. **if** $j \neq \lceil \log_2 q \rceil$ **then**
6. **for** i from 0 to $n-1$ **do**
7. **if** modular reduction of $\left(c_1[i] \cdot r_2[i] + c_2^{j-1}[i]\right)$ does*n't occur* **then**
8. $k[i] \leftarrow 2 \cdot k[i] + 1, c_2^j[i] \leftarrow \left\lfloor \frac{q-1}{2^j} \times k[i] \right\rfloor$
9. **else**
10. $k[i] \leftarrow 2 \cdot k[i] - 1, c_2^j[i] \leftarrow \left\lfloor \frac{q-1}{2^j} \times k[i] \right\rfloor$
11. **end if**
12. **end for**
13. **else**
14. **for** i from 0 to $n-1$ **do**
15. **if** modular reduction of $\left(c_1[i] \cdot r_2[i] + c_2^{j-1}[i]\right)$ does*n't occur* **then**
16. $c_2^j[i] \leftarrow c_2^{j-1}[i] + 1$
17. **else**
18. $c_2^j[i] \leftarrow c_2^{j-1}[i] - 1$
19. **end if**
20. **if** modular reduction of $\left(c_1[i] \cdot r_2[i] + c_2^j[i]\right)$ does*n't occur* **then**
21. **Return** $r_2[i] \leftarrow q - c_2^j[i] - 1$
22. **else**
23. **Return** $r_2[i] \leftarrow q - c_2^j[i]$
24. **end if**
25. **end for**
26. **end if**
27. **end for**

IV. 실험 결과

4.1 모듈러 연산 구별 실험

제안하는 선택 암호문 SPA 공격은 모듈러 덧셈 연산에서 모듈러 연산의 발생 여부를 구별 할 수 있어야 공격이 가능하다. 따라서 본 절에서는 8비트로 구현된 모듈러 덧셈에서 모듈러 연산 발생 여부를 확실히 구별할 수 있음을 실험을 통해 보인다. 우리는 8 비트 ring-LWE 암호시스템의 복호화에서 사용되는 모듈러 덧셈이 실제 디바이스에서 동작 할 때 비밀값 추측이 가능한 취약성이 존재하지 알아보기 위해 NewAE Technology의 ChipWhisperer-Lite를 이용해 실험을 진행하였다. ChipWhisperer-Lite에서 사용하는 공격 대상 칩은 XMEGA128D4로 8/16-bit AVR XMEGA 마이크로컨트롤러이다. 신호는 7.37MHz 클록 주파수(clock frequency)로 모듈러 덧셈이 실행 될 때 발생 되는 전력신호를 29.5MHz 샘플링 레이트로 획득하였다.

Fig.1.은 모듈러 연산 발생에 따른 수집 전력의 차이를 보여주고 있다. 그림의 가로축은 시간을 나타내고, 세로축은 수집된 전력의 크기를 나타낸다. 실선은 모듈러 연산이 발생한 전력이고, 점선은 발생하지 않은 전력으로 모듈러 연산이 발생 할 때 더 많은 시간이 필요함을 알 수 있다. 이는 작은 차이지만 두 연산의 차이가 확실히 존재함을 확인 할 수 있다.

Fig.2.의 위쪽 그림은 실제 복호화 부분의 모듈러 덧셈이 동작할 때 수집한 전력을 보여준다. 동일한 연산이 여러 번 반복됨을 확인할 수 있으며 이를 확대해보면(Fig.2.의 하단 참조) Fig.1.과 같이 모듈러 연산 발생 여부에 따라 수집된 전력의 포인트 차이가 존재함을 알 수 있다.

이처럼 실제 모듈러 발생 여부에 따라 차이가 존재함을 확인할 수 있다. 그러나 본 논문에서 제안하는 선택 암호문 SPA 공격을 수행하기 위해 공격자는 각 시행 단계에서 $q=7681$인 경우 $n=256$ 번의 모듈러 연산 발생 여부를 확인하여야 한다. 즉, 비밀키 r_2를 복원하기 위해 13번의 시행이 필요하므로 총 $256\times 13=3328$번($q=12289$인 경우 총 $512\times 14=7168$번)의 모듈러 연산 확인 과정이 필요하다. 이를 단순히 시각적으로만 판단하기에는 공격 시간 증가의 문제점이 존재한다. 이러한 문제점을 해결하기 위해 본 논문에서는 두 벡터의 유사도 측정 방법을 이용하여 공격시간을 단축하고자 한다.

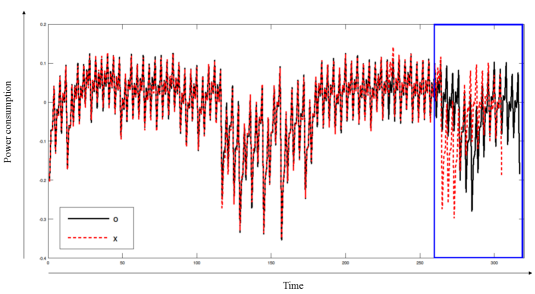


Fig. 1. Difference of power consumption according to whether the reduction occurs.

4.2 효과적인 모듈러 연산 구별 방법

앞 절에서 언급된 문제점을 해결하기 위해 본 절에서는 두 벡터의 유사도를 측정하는 방법의 비교를 통해 효과적인 모듈러 연산 구별 방법에 대해 살펴본다.

길이가 n인 두 벡터 $X=\{X_1,X_2,\cdots,X_n\}$ 및 $Y=\{Y_1,Y_2,\cdots,Y_n\}$의 유사도를 측정하는 방법은 다음과 같이 나눌 수 있다.

(1) 유클리드 거리 (euclidean distance)
: 두 벡터의 상대적인 거리 차를 측정하는 방법으로 거리 값이 작을수록 두 벡터가 유사함을 나타낸다.

$$ED(X,Y)=\sqrt{\sum_{k=1}^{n}(X_k-Y_k)^2}$$

(2) 코사인 계수 (cosine coefficient)
: 코사인 유사도는 두 벡터의 코사인 값을 계산해 얼마나 유사한 성향인지 확인하는 방법으로 코사인 값을 구하는 방법은 아래와 같다.

$$Cos(X,Y)=\frac{\sum_{k=1}^{n}X_k\times Y_k}{\sqrt{\sum_{k=1}^{n}(X_k)^2}\sqrt{\sum_{k=1}^{n}(Y_k)^2}}$$

이 값은 벡터의 크기가 아닌 방향의 유사도를 판단하는 목적으로 사용되며, 두 벡터의 방향이 완전히 같을 경우 1, 90°의 각을 이룰 경우 0, 180°로 완전히 반대 방향인 경우 -1의 값을 갖는다.

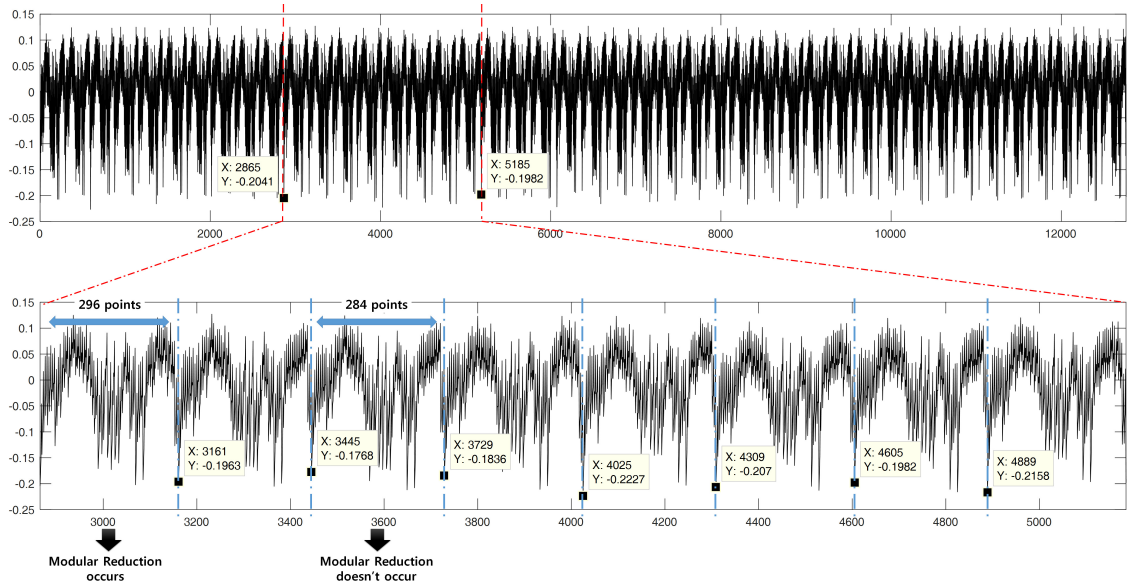


Fig. 2. Top: Power consumption trace of modular additions in decryption. Bottom: Expansion of 8 modular addition operation of the top trace.

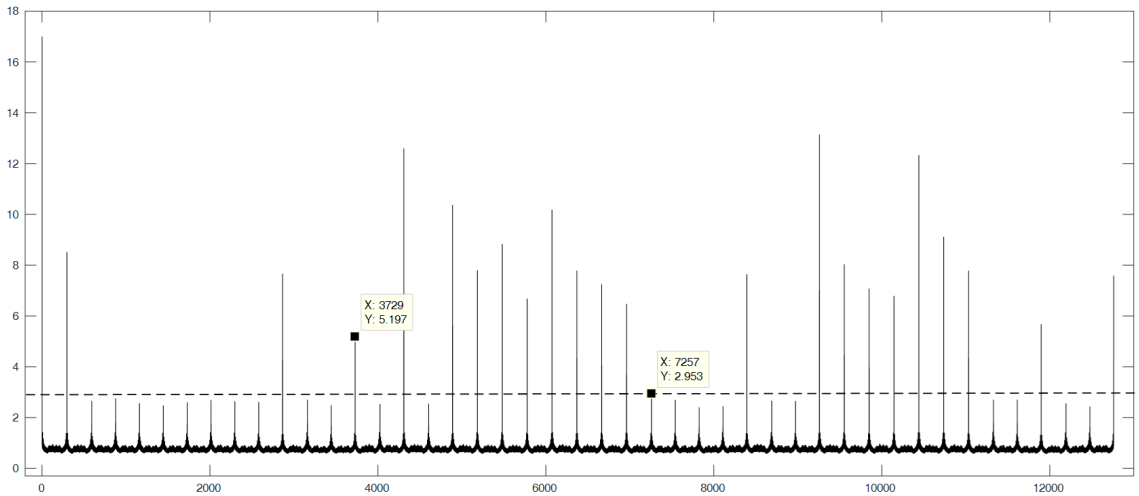


Fig. 3. The result of PCDC.

(3) 상관계수(correlation coefficient)

: 상관계수는 두 분포의 선형성을 의미하며, 두 분포가 $Y = aX + b$의 형태와 유사한 관계로 양의 상관관계를 가지게 되면 상관계수는 1에, 음의 상관관계가 있으면 -1에 가까운 값을 갖게 되고 두 분포간의 연관성이 없고 독립적일수록 그 값은 0에 가까워진다.

$$Corr(X, Y) = \frac{E(XY) - E(X)E(Y)}{\sqrt{VAR(X)\,VAR(Y)}}$$

(4) 차이의 평균(mean of difference)

: 두 벡터의 차이의 평균을 계산하는 것으로 유사할수록 0에 가까워진다.

$$MoD(X,Y)=\frac{\sum_{k=1}^{n}(X_k-Y_k)}{n}$$

(5) PCDC(Plotting Collision Detection Criterion) : 부채널 분석의 한 종류인 충돌쌍 공격 시 사용되는 유사도 측정 방법으로 아래와 같은 식을 이용해 계산된다[10].

$$PCDC(X,Y)=\frac{\sigma(X)}{\sigma(X-Y)}$$

효과적인 모듈러 연산 구별 방법을 찾기 위해 위에 언급된 5가지 방법을 사용하여 수집된 모듈러 덧셈(Fig.2. 상단) 연산에 대해 모듈러 연산 발생 여부를 Algorithm 3.과 같이 조사하였다. Algorithm 3.의 첫 번째 단계는 입력된 파형의 특정부분을 기준 파형으로 사용하기 위해 구간을 설정하는 단계이다. 예를 들어, 본 실험은 파형의 길이(Algorithm 3.에서의 m)가 13057인 경우에 대하여, 처음 모듈러 연산이 발생한 부분을 기준으로 사용하기 위해 $k=1, l=296$으로 설정 후 진행하였다. k, l은 공격자에 따라 다르게 설정 가능하다.

Algorithm 3.의 $len(\ )$은 입력된 파형의 길이(포인트 수)를 의미하며 T_i는 파형 T의 i번째 포인트를 의미한다.

Algorithm 3. Detection Criteria

Input : trace of modular additions $T=\{T_1, T_2 \cdots, T_m\}$

Output : result trace of detection criteria

1. **Set** $T'=\{T_k, T_{k+1} \cdots, T_l\}$ where $1 \le k < l \le m$
2. **For** j from 1 to $\left(len(T)-len(T')\right)$ **do**
3. **Set** $T''=\{T_j, T_{j+1} \cdots, T_{j+l-k}\}$
4. **Compute** Detection Criteria (such as PCDC etc.)
5. **end for**

Fig. 3.은 모듈러 연산을 구별하기 위한 검출방법으로 PCDC 사용의 결과를 보여준다. 가로의 점선은 연산구분이 가능한 문턱값(threshold value)을 표시한 것으로 쉽게 연산이 구별됨을 알 수 있다. 즉, 문턱값 이하의 값을 갖는 부분은 모듈러 연산이 일어나지 않은 부분이며 문턱값 초과 부분은 모듈러 연산이 일어난 부분이다.

Table 2.은 각 방법의 모듈러 연산 발생 여부에 따른 문턱값의 최대(또는 최소)값 및 이에 따른 $Ratio=\frac{Max\ Value}{Min\ Value}$ 값을 나타내고 있다. $Ratio$값이 크다는 것은 모듈러 연산 발생여부를 시각적으로 보다 쉽게 확인 할 수 있음을 의미한다. 예를 들어 Fig. 3.에서 모듈러 연산 발생 부분의 PCDC 최소값은 5.197이며 발생하지 않은 부분의 PCDC 최대값은 2.953으로 PCDC를 이용한 검출방법의 $Ratio$는 1.7599 이다.

Table 1.에서 확인 할 수 있듯이 5가지 모든 경우에 대해 모두 모듈러 연산 발생 구분이 가능 했으나, PCDC를 이용할 때 가장 쉽게 모듈러 연산 발생 여부 확인이 가능함을 알 수 있다.

Table 1. threshold value and ratio

Detection Criteria	Max Value	Min Value	Ratio
ED	0.3996	0.2558	1.1562
Cos	0.9781	0.9482	1.0315
Corr	0.9827	0.9462	1.0385
MoD	0.0121	0.01151	1.0512
PCDC	5.197	2.953	1.7599

V. 결 론

본 논문에서는 ring-LWE 암호 알고리즘 NTT 구현의 복호화 단계 시 사용되는 모듈러 덧셈의 취약점을 이용한 선택 암호문 SPA 공격을 제안하였다. 본 공격은 DPA 공격 보다 적은 $\lceil \log_2 q \rceil$ 번의 복호화 시도로 비밀키를 모두 찾을 수 있다는 이점이 존재한다.

또한 실제 디바이스에서 동작되는 ring-LWE 복호화 과정의 모듈러 덧셈에서 모듈러 발생 여부에 따른 차이가 존재 한다는 취약점이 발생함을 실험을 통해 보였다. 또한 모듈러 연산 발생 여부 확인에 있어 보다 쉽게 구별할 수 있는 방법에 대한 비교를 통해 효과적인 공격이 가능하도록 하였다.

향후 제안한 공격 방법을 다양한 8비트 디바이스에 적용하여 실제 모듈러 덧셈 연산에 대한 취약점이 다른 디바이스에서도 존재하는지에 대한 연구 및 부채널 분석 대응방법과 다른 취약점의 존재에 대한 연구가 더 필요할 것이다.

References

[1] P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," Proceedings of the 35th Annual Symposium on Foundations of Computer Science, pp. 124-134, Nov. 1994.

[2] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," Proceedings of the 16th Annual International Cryptology Conference, pp. 104-113, Aug. 1996.

[3] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," Proceedings of the 19th Annual International Cryptology Conference, pp. 388-397, Aug. 1999.

[4] C. Chen, T. Eisenbarth, I.V. Maurich, and R. Steinwandt, "Differential Power Analysis of a McEliece Cryptosystem," Proceedings of the 13th International Conference on Applied Cryptography and Network Security, pp. 538-556, Jun. 2015.

[5] M.K. Lee, J.E. Song, D.H. Choi, and D.G. Han, "Countermeasures against the power analysis attack for the NTRU public key cryptosystem," IEICE Transactions on Fundamentals of Electronics on Communications and Computer Sciences, vol.E93-A, no.1, pp.153 - 163, Jan. 2010.

[6] V. Lyubashevsky, C. Peikert, and O. Regev, "On Ideal Lattices and Learning with Errors over Rings," Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 1 - 23, Jun. 2010.

[7] S. Roy, F. Vercauteren, N. Mentens, D. Chen, and I. Verbauwhede, "Compact ring-LWE cryptoprocessor," Proceedings of the 16th Workshop on Cryptographic Hardware and Embedded Systems, pp. 371-391, Sep. 2014.

[8] N. Göttert, T. Feller, M. Schneider, J. Buchmann, and S. Huss, "On the Design of Hardware Building Blocks for Modern Lattice-Based Encryption Schemes," Proceedings of the 14th Workshop on Cryptographic Hardware and Embedded Systems, pp. 512-529, Sep. 2012.

[9] Z. Liu, H. Seo, S. Roy, J. Großschädl, H. Kim, and I. Verbauwhede, "Efficient Ring-LWE encryption on 8-bit AVR processors," Proceedings of the 17th Workshop on Cryptographic Hardware and Embedded Systems, pp. 663-682, Sep. 2015.

[10] G. Perin, L. Imbert, L. Torres, and P Maurine, "Practical analysis of rsa countermeasures against side-channel electromagnetic attacks," Proceedings of the 12th Smart Card Research and Advanced Application Conference, pp. 200 - 215, Nov. 2013.

[11] A. Park and D,G. Han, "Chosen ciphertext Simple Power Analysis on software 8-bit implementation of ring-LWE encryption," Proceedings of the Hardware-Oriented Security and Trust (AsianHOST), pp. 1 - 6, Dec. 2016.

[12] A. Park, Y.S. Won and D,G. Han, "Chosen Ciphertext SPA attack on ring-LWE cryptosystem," CISC-W'16, D1-3, Dec. 2016.

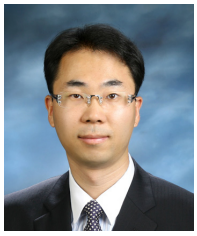
[13] O. Reparaz, S. Roy, F. Vercauteren, and I. Verbauwhede, "A masked ring-LWE implementation," Proceedings of the 17th Workshop on Cryptographic Hardware and Embedded Systems, pp. 683-702, Sep. 2015.

[14] O. Reparaz, R. de Clercq, S. Roy, F. Vercauteren, and I. Verbauwhede, " Additively homomorphic ring-LWE masking," Proceedings of the 7th International Conference on Post-Quantum Cryptography, pp. 233-244, Feb. 2016.

〈저자소개〉

박 애 선 (Aesun Park) 학생회원
2011년 2월: 국민대학교 수학과 학사
2013년 2월: 국민대학교 수학과 석사
2014년 3월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 부채널 분석 및 대응법, 스마트 카드 평가, Post-quantum cryptography 등

원 유 승 (Yoo-Seung Won) 학생회원
2012년 2월: 국민대학교 수학과 학사
2014년 2월: 국민대학교 수학과 석사
2014년 2월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 정보보호, 부채널 분석, 대칭키 암호 알고리즘, 스마트 카드 보안



한 동 국 (Dong-Guk Han) 종신회원
1999년 2월: 고려대학교 수학과 졸업(학사)
2002년 2월: 고려대학교 수학과 석사 (이학석사)
2005년 2월: 고려대학교 정보보호대학원 박사 (공학박사)
2004년 4월~2005년 4월: 일본 Kyushu Univ., 방문연구원
2005년 4월~2006년 4월: 일본 Future Univ.-Hakodate, Post.Doc.
2006년 6월~2009년 2월: 한국전자통신연구원 정보보호연구단 선임연구원
2009년 3월~현재: 국민대학교 수학과 부교수
〈관심분야〉 공개키 암호시스템 안전성 분석 및 고속 구현, 부채널 분석 및 대응법 설계, IoT 정보보호 기술