Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.6, Dec. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.6.1059

양자내성암호 NTRU에 대한 전력 부채널 공격 및 대응방안*

장 재 원,[1†] 하 재 철[2‡]
[1,2]호서대학교 (대학원생, 교수)

Power-Based Side Channel Attack and Countermeasure on the Post-Quantum Cryptography NTRU*

Jaewon Jang,[1†] Jaecheol Ha[2‡]
[1,2]Hoseo University (Graduate student, Professor)

요 약

양자 컴퓨터의 계산 능력을 고려하여 설계된 양자 내성 암호 NTRU는 수학적으로 안전한 암호 조건을 만족하지만 하드웨어 구현 과정에서는 전력 분석 공격과 같은 부채널 공격 특성을 고려해야 한다. 본 논문에서는 NTRU의 복호화 과정 중 발생하는 전력 신호를 분석할 경우 개인 키가 노출될 가능성이 있음을 검증한다. 개인 키를 복구하는 데에는 단순 전력 분석 공격(Simple Power Analysis, SPA), 상관 전력 분석 공격(Correlation Power Analysis, CPA)과 차분 딥러닝 분석 공격(Differential Deep Learning Analysis, DDLA)을 모두 적용할 수 있었다. 이러한 전력 부채널 공격에 대응하기 위한 기본적인 대응책으로 셔플링 기법이 있으나 보다 효과적인 방법을 제안한다. 제안 방식은 인덱스별로 곱셈(multiplication)후 누산(accumulation)을 하는 것이 아니라 계수별로 누산 후 덧셈만 하도록 함으로써 곱셈 연산에 대한 전력 정보가 누출되지 않도록 하여 CPA 및 DDLA 공격을 방어할 수 있다.

ABSTRACT

A Post-Quantum Cryptographic algorithm NTRU, which is designed by considering the computational power of quantum computers, satisfies the mathematically security level. However, it should consider the characteristics of side-channel attacks such as power analysis attacks in hardware implementation. In this paper, we verify that the private key can be recovered by analyzing the power signal generated during the decryption process of NTRU. To recover the private keys, the Simple Power Analysis (SPA), Correlation Power Analysis (CPA) and Differential Deep Learning Analysis (DDLA) were all applicable. There is a shuffling technique as a basic countermeasure to counter such a power side-channel attack. Nevertheless, we propose a more effective method. The proposed method can prevent CPA and DDLA attacks by preventing leakage of power information for multiplication operations by only performing addition after accumulating each coefficient, rather than performing accumulation after multiplication for each index.

Keywords: Side channel Attack, Correlation Power Analysis, Differential Deep Learning Analysis, Countermeasures

Received(08. 29. 2022), Modified(10. 14. 2022) Accepted(10. 14. 2022)
* 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2020R1F1A1074358)
* 본 논문은 2022년도 한국정보보호학회 하계학술대회에 발표한 우수논문을 개선 및 확장한 것임
† 주저자, 58580454@naver.com
‡ 교신저자, jcha@hoseo.edu(Corresponding author)

I. 서 론

현재 컴퓨터의 발전은 비트(bit)로 연산을 수행하는 기존 컴퓨터에서 양자역학적 원리를 사용한 큐빗(qubit)으로 0과 1이 중첩된 상태로 연산을 수행하여 초고속으로 데이터를 처리할 수 있는 양자 컴퓨터로 발전하는 중이다. 이러한 발전은 기상 예측, 빅데이터 분석 등 높은 연산량이 필요했던 슈퍼컴퓨터를 양자 컴퓨터로 대체가 가능하게 되었다. 하지만 양자 컴퓨터의 등장으로 인해 부정적 요소도 발생하게 되는데 그 대표적인 것이 기존의 암호화 시스템을 악의적 목적으로 해독하는 것이다.

현재 정보에 대한 기밀성과 무결성을 보장하기 위해 AES[1]와 RSA와 같은 대칭 키, 공개 키 암호 알고리즘을 사용 중에 있다. 그러나 대칭 키 암호 알고리즘 같은 경우에는 양자 컴퓨터를 이용한 Grover 알고리즘을 이용한다면 비밀 키를 찾는데 기존 $O(N)$의 연산량에서 $O(\sqrt{N})$으로 연산량이 축소되어 안정성을 위협받게 된다[2]. RSA와 같은 소인수 분해 문제를 이용한 공개 키 암호 알고리즘 또한 양자 컴퓨터를 이용한 Shor 알고리즘을 이용하면 다항식 시간 내에 소인수 분해 문제를 해결할 수 있다[3].

이와 같이 양자 컴퓨터 등장에 따라 상용 암호 알고리즘의 안전성이 위협받게 되자 NIST(National Institute of Standards and Technology)에서는 양자 내성 암호(Post Quantum Cryptography) 표준화 작업을 시작하여 현재까지 총 3번의 평가 분석을 진행하였고 2024년경에 최종 선정 예정에 있다[4].

양자 내성 암호 알고리즘의 종류는 공개 키 암호(Public Key Encryption)/키 교환 메커니즘(Key Encapsulation Mechanism)과 전자서명(Digital Signature) 분야가 있으며 표준화 과정 Round3까지 PKE/KEM에서는 NTRU, SABER 등 5개의 후보와 전자서명 분야에는 FALCON, Rainbow 등 3개 알고리즘이 선정되었다. 이와 같은 양자 내성 암호 알고리즘들은 각 수학적 난제에 기반하여 IND-CCA2(Indistinguishability under adaptive chosen ciphertext attack), INDCPA(Indistinguishability under chosen-plaintext attack)와 같은 안전성 조건을 만족하여야 한다[5]. 그러나 이러한 수학적인 안전 조건을 갖추더라도 이 암호 알고리즘을 구현할 때 발생하는 취약점으로 인해 부채널 정보가 누출될 수 있고 이를 이용한 비밀 정보에 대한 공격이 시도되고 있어 이에 대한 대책이 필요하다.

본 논문에서는 NIST 3Round의 PKE/KEM 분야의 알고리즘 중 격자 기반의 암호 알고리즘인 NTRU에 대해 부채널 공격을 수행하며 알고리즘의 물리적 안전성을 검증해 본다. 다만 NTRU 알고리즘이 연구가 진행된 시점에 Round 4의 후보에는 선정되지 못했지만, 지속적으로 보완 작업을 진행하면 NTRU 알고리즘이 다양한 암호 기술에 활용될 수 있기 때문에 NTRU에 대한 부채널 공격 분석 및 대응 방안 연구는 필요하다고 할 수 있다[6, 7].

NTRU 암호 알고리즘에 기반한 KEM 메커니즘에서 송신자는 세션 키를 상대방의 공개 키로 암호화하여 수신자에게 보내고, 수신자는 자신의 개인 키로 암호문을 복호하여 공통된 세션 키를 공유하게 된다. 이 경우 개인 키로 복호화를 수행하는 과정에서 사용하는 전력 정보를 누출할 수 있는데 논문에서는 단순 전력 분석 공격(Simple Power Analysis, SPA) 상관 전력 분석 공격(Correlation Power Analysis, CPA)과 차분 딥러닝 분석 공격(Differential Deep Learning Analysis, DDLA)[8-10]을 사용하여 비밀 키를 복구할 수 있음을 확인하였다. 또한 이러한 전력 기반 부채널 공격(Side Channel Attack, SCA)에 대응하기 위한 대응책으로 새로운 다항식 곱셈 방식을 제안한다. 이 방식은 인덱스별로 곱셈(multiplication)후 누산(accumulation)을 하는 것이 아니라 계수별로 누산 후 덧셈만 하도록 함으로써 곱셈 연산에 대한 전력 정보가 누출되지 않도록 함으로써 기존 셔플링 방법보다 효과적으로 CPA 및 DDLA 공격을 방어할 수 있다.

II. 배경지식

2.1 부채널 공격

부채널 공격은 암호 알고리즘의 물리적인 구현 과정에서의 취약점을 공격하여 비밀정보를 획득하는 공격을 말한다. 이때 발생하는 대표적인 부채널 누출로는 소비 전력, 연산 소요 시간, 전자파 등이 있으며, 이 중에서도 전력이나 전자파를 이용한 부채널 공격이 가장 효과적인 공격 방법으로 알려져 있다. 초기에 제안된 NTRU에 대한 단순 전력 분석 공격 상관

전력 분석 공격이 시도되었으며 이에 대한 대응책도 제시된 바 있다[11, 12]. 초기 NTRU 알고리듬은 다항식 곱셈을 위해 오퍼랜드(operand scanning) 기법을 사용하였으나 NIST 표준으로 제안된 기법은 곱셈 결과 스캐닝(product scanning) 기법을 사용하고 있다는 것이 차이가 있다.

본 논문에서도 NTRU 알고리즘 중 복호화 구현 과정에서 발생하는 소비 전력 누출을 이용하여, 단순 전력 분석 공격, 상관 전력 분석 공격과 딥러닝을 이용한 차분 딥러닝 분석 공격 등 다양한 공격을 수행하여 개인 키를 복구하였다. 이러한 부채널 공격 중에서 DDLA 공격이란, 2018년 B. Timon이 제안한 비프로파일링 공격 모델로서 가정한 키에 대해 계산된 중간 값을 해당 연산 소비 전력 파형의 레이블로 학습시키는 과정에서의 학습 속도, 정확도 등의 추이를 이용하여 키를 복구할 수 있는 공격 방법을 말한다. 이때 사용되는 레이블의 종류는 MSB(Most Significant Bit), LSB(Least Significant Bit), 해밍 웨이트(Hamming Weight, HW) 등이 있으며, 이러한 레이블을 사용하여 소비 전력을 학습 시켰을 때 올바른 키는 소비 전력과 연관성이 있어 올바르지 않은 키보다 상대적으로 정확도는 더 높고, 손실 값은 더 낮은 값을 갖게 된다.

본 논문에서는 DDLA를 통해 개인 키를 복구할 때 사용되는 레이블 값을 기존 해밍 웨이트 기반에서 해밍 웨이트의 중간 값을 기준으로 1 또는 0 레이블을 생성하는 해밍 웨이트 기반 2진 레이블(HW-based binary label)을 학습에 사용하였다.

2.2 NTRU 알고리즘

다음 Fig. 1은 NTRU의 키 생성, 암호화, 복호화 과정을 나타낸 것이다[13]. NTRU에서 사용하는 다항식 링 $\mathbb{R} = Z_q[x]/(\Phi_1\Phi_n) = Z_q[x]/(x^N-1)$ 이고 이 때 $\Phi_1 = (x-1), \Phi_n = (x^N-1)/(x-1)$으로 표기 하였다. 여기서 N은 환 $\mathbb{R}$ 위의 다항식의 차수이며, 본 연구에서 사용한 ntruhps2048509 버전에서는 $N=509$로 설정하였다. 또한, 다항식의 계수는 모듈러 q상에서 연산되는데 $q=2^{11}=2048$ 설정하였다.

키를 생성하는 알고리즘인 Keygen에서 $Sample_fg$는 랜덤한 두 개의 삼진 다항식

KeyGen(seed)
1. $(f, g) \leftarrow Samplc\_fg(sccd)$
2. $f_q \leftarrow (1/f) \bmod (q, \Phi_n)$
3. $h \leftarrow (3 \cdot g \cdot f_q) \bmod (q, \Phi_1, \Phi_n)$
4. $h_q \leftarrow (1/h) \bmod (q, \Phi_n)$
5. $f_q \leftarrow (1/f) \bmod (3, \Phi_n)$
6. return $((f, f_q, h_q), h)$

Encrypt$(h, (r, m))$
1. $m' \leftarrow Lift(m)$
2. $c \leftarrow (r \cdot h + m') \bmod (q, \Phi_1, \Phi_n)$
3. return c

Decrpyt$((f, f_q, h_q), c)$
1. if $c \neq 0 (\bmod (q, \Phi_1)$ return $(0,0,1)$
2. $a \leftarrow (c \cdot f) \bmod (1, \Phi_1, \Phi_n)$
3. $m \leftarrow (a \cdot f_p) \bmod (3, \Phi_n)$
4. $m' \leftarrow Lift(m)$
5. $r \leftarrow ((c-m') \cdot h_q) \bmod (q, \Phi_n)$
6. if $(r, m) \in L_r \times L_m$ return $(r, m, 0)$
7. else return $(0,0,1)$

Fig. 1. Description of NTRU PKE

(ternary polynomial)을 생성하며 이를 이용하여 개인 키 f 와 공개 키 h를 생성하여 반환한다. 삼진 다항식이란 각 다항식의 계수가 3개를 가지는 다항식으로서 NTRU에서는 {-1, 0, 1}의 값만 가진다. 이때 f_p, h_q는 f와 h의 역원을 나타낸다. 송신자는 공개 키를 이용하여 평문 정보를 암호화하고 Decrypt에서 수신자는 개인 키를 이용하여 비밀 정보를 복호화한다. 이때 연산되는 다항식 곱셈은 모두 환 $\mathbb{R}$ 위에서 이루어진다.

2.3 부채널 공격 지점 분석

NTRU에 대한 부채널 공격은 복호화 함수인 Decrypt 과정 중 $c \cdot f \bmod (q, \Phi_1\Phi_n)$에서 개인 키 f 가 사용되므로 이 연산 과정에서 발생하는 전력 소비 신호를 이용한다. 이 과정을 구체적으로 살펴보면 다음과 같이 개인 키 f와 암호문 c의 곱셈 연산이 수행됨을 알 수 있다.

$$r(x) = c(x) \cdot f(x) \bmod (x^N - 1)$$
$$= \left(\sum_{i=0}^{N-1} c_i x^i \right) \cdot \left(\sum_{i=0}^{N-1} f_i x^i \right) \bmod (x^N - 1)$$
$$= \sum_{k=0}^{N-1} \left(\sum_{i+j=k \bmod N} c_i \cdot f_j \right) x^k$$
$$= \sum_{k=0}^{N-1} r_k \cdot x^k$$

여기서 두 다항식의 곱셈 결과를 나타내는 결과 값의 각 계수는 다음과 같이 표현할 수 있다.

$$r_k = \sum_{i=1}^{N-k-1} c_{k+i} \cdot f_{N-i} + \sum_{i=0}^{k} c_{k-i} \cdot f_i$$

위 다항식 곱셈 과정을 수행하는 Poly_Rq_mul 함수를 나타낸 것이 다음 Fig. 2와 같다.

NTRU에서 사용하는 다항식 곱셈은 메모리 효율을 높이기 위해 오퍼랜드 스캐닝(operand scanning) 방식이 아니라 결과 값을 스캐닝(product scanning) 하는 방식을 채택하고 있다.

본 논문에서는 NTRU에 대한 전력 부채널 공격은 SPA 공격, CPA 공격 그리고 DDLA 공격을 진행하였다. 우선 SPA 공격에서는 전력 소비 파형을 직관적으로 관찰함으로써 개인 키인 삼진 다항식 f의 계수 중 0의 위치를 찾아낼 수 있었다. 그러나 계수 1과 -1을 구별하기 단순 파형만으로는 구별하기 어려웠다. 따라서 CPA 공격을 통해 정확히 개인 키를 복구할 수 있었다

CPA 공격은 Fig. 2의 3과 4번 라인 연산에서 개인 키와 암호문의 곱셈 연산의 결과가 $r[k]$의 배열에 하나씩 더해지며 중첩되기 때문에 키의 마지막 부분부터 하나씩 순차적으로 복구한다. 이를 순서화하면 다음과 같다.

Algorithm 1 : *poly_Rq_mul*(c, f, r)

Input : *Secret Key* f
Input : *Ciphertext* c
Output : *Polynomial* r

1. *for* $0 \le k < NTRU\_N$
2. $r[k] = 0$
3. *for* $1 \le i < NTRU\_N\text{-}k$
4. $r[k] += c[k+i] * f[NTRU\_N\text{-}i]$
5. *for* $0 \le i < k+1$
6. $r[k] += c[k-i] * f[i]$

Fig. 2. Polynomial multiplication in NTRU

1) $r_0[0]$이 0으로 초기화된 상태에서 $r_1[0] += c[1] * f[NTRU_N-1]$연산 후 업데이트되는 $r_1[0]$에 대한 해밍 웨이트를 구하여 CPA공격을 통해 $f[NTRU_N-1]$의 값을 구할 수 있다. 여기서 $r_i[0]$는 i번째 반복문을 수행한 후의 중간 값을 나타낸다.

2) $r_2[0] += c[2] * f[NTRU_N-2]$에 대한 CPA 공격을 수행하기 위해서는 $r_2[0]$값에 대한 해밍 웨이트가 필요하다. 이때 $r_2[0]$는 이전 라운드의 $r_1[0]$값이 더해진 값이므로 이전 라운드에서 구한 키를 고정시킨 후 연산한 결과인 $r_1[0]$값을 더해 준 해밍 웨이트를 구하여 CPA 공격을 수행하여 두 번째 비밀 정보인 $f[NTRU_N-2]$의 값을 구한다.

3) 위와 같은 방법을 반복하여 CPA 공격을 수행하여 f의 마지막부터 $f[1]$까지 구할 수 있다.

4) poly_Rq_mul 연산의 3, 4번 라인 연산 지점에서는 $f[NTRU\_N]$ 부터 $f[NTRU_N-i]$까지만 쓰이기 때문에 $f[0]$번째 키를 CPA 공격으로 복구할 수 없다. 그러나 $f[0]$는 5, 6번 라인의 연산에서 사용하기 때문에 복구가 가능하다.

III. NTRU에 대한 전력 분석 공격

3.1 실험 환경

본 논문에서 NTRU 공격을 수행하기 위해 사용한 코드는 NIST PQC Round 3 최종 후보로 올라 공개되어 있는 참조용 소스 코드 중 ntruhps2048509 버전을 사용한다[14]. 전력을 측정하기 위해 Chipwhisperer Lite를 사용하였으며 측정 속도는 29.5MS/s 이다. 부채널 공격 실험은 ARM-Coretex-M4 코어가 탑재된 32비트 프로세서인 STM32F MCU에 ntruhps2048509 복호 코드를 구현한 후 구동하면서 개인 키에 대한 소비 전력을 측정하여 수행하였다. 다음 Fig. 3은 MCU가 탑재된 실험 보드와 Chipwhisperer Lite와의 인터페이스를 나타낸 것이다.

ChipWhisperer Lite를 사용하여 poly_Rq_mul 연산 과정에서 측정한 파형을 나타낸 것이 Fig. 4이다. 개인 키의 계수 중 20개가 사용된 파형은 약 3000샘플 안에 종료되며 본 연구에서는 이 키를 대상으로 SPA, CPA 공격을 수행하였다. 실험의 편의를 위해 20개의 키 계수는 key =

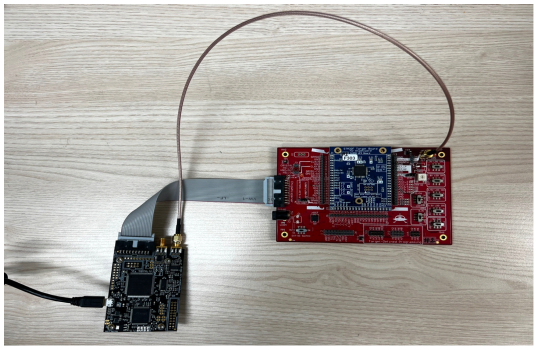

Fig. 3. STM32F MCU and ChipWhisperer platform

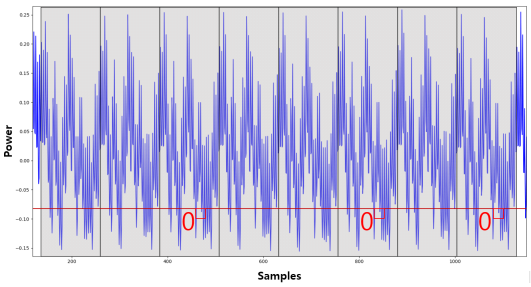

Fig. 4. Result of SPA on polynomial multiplication of NTRU

{1,1,-1,0,-1,0,1,-1,0,1,-1,1,0,-1,0,1,-1,0,-1,1}로 가정하였다. 여기서 주의할 점은 개인 키 계수중 -1은 $q-1$을 의미하므로 2047로 표현이 가능하다는 것이다.

3.2 SPA 공격

곱셈 연산 중 개인 키 0에 대한 연산은 1과 -1의 연산에 비해 전력 소모량이 적기 때문에 특정 위치에서 개인 키를 쉽게 관찰할 수 있다. 본 논문에서는 100개 까지의 파형을 측정한 후 이를 평균 내어 단일 파형으로 확인해 본 결과 Fig. 4와 같이 적은 데이터로도 0의 값을 구별할 수 있었다.

그림에서 볼 수 있듯이 개인 키의 계수 중 0이 사용된 위치를 확인할 수 있었다. 그러나 나머지 1과 -1은 단순 전력 파형으로는 구별하기 어려워 CPA 공격을 추가적으로 시도하였다.

3.3 CPA 공격

앞서 수행한 SPA 공격을 통해 0의 위치를 찾은 후 CPA 공격을 통해 나머지 키를 복구한다. 공격을 위해 수집한 전력 파형은 업데이트되는 $r[0]$ 값을 중간 값으로 가정하고 파형과 중간 값의 상관도를 조사하여 개인 키를 순차적으로 복구한다.

논문에서는 약 250개 정도의 전력 파형을 이용하여 CPA 공격을 수행하여 충분히 개인 키를 복구할 수 있음을 확인하였다. 상기한 바와 같이 개인 키의 마지막 계수를 복구하면 다음 계수를 순차적으로 복구한다. 실제 마지막 키에 대한 연산 시작 지점은 132샘플이며 끝 지점은 256샘플이었다. 이후로 키 하나의 연산은 124샘플 동안 이루어짐을 확인하였다. 다음 Fig. 5는 공격 대상이 되는 키와 전력 파형의 상관도 그래프를 나타낸 것이다. 그림에서 보는 바와 같이 비밀 키의 계수 값이 1인지 -1인지 확연히 구별할 수 있었다.

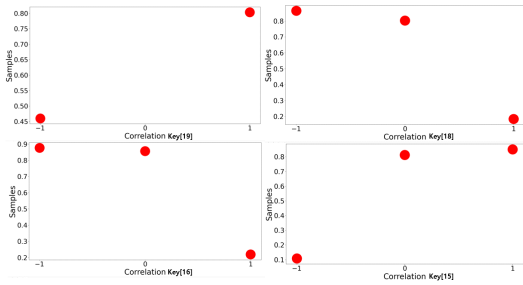

Fig. 5. Result of CPA on polynomial multiplication of NTRU

3.4 DDLA 공격

DDLA 공격에서도 개인 키는 사용되는 알고리즘에 따라 마지막 계수부터 순차적으로 복구한다. DDLA 공격에 사용되는 딥러닝 모델은 MLP(Multi Layer Perceptron), CNN(Convolutional Neural Network) 모델이 주로 사용되는데 본 논문에서는 MLP 모델을 이용하여 DDLA 공격을 수행하였으며, 학습의 정확도(accuracy)와 손실(loss) 값을 실제 키를 찾는 지표로 사용하였다.

DDLA 공격에 사용된 레이블은 앞서 언급한 해밍 웨이트 기반 이진 레이블을 사용하였다. 이 방법은 레이블을 정하는 데이터의 중간 값의 해밍 웨이트를 구하고 중간보다 높은 해밍 웨이트를 가지면 1로 낮으면 0으로 레이블을 결정하여 학습 정도를 보고 개인 키를 복구하는 기법이다.

하지만 NTRU의 Poly_Rq_mul 함수에서 중간 값을 두 바이트를 사용하는 반면 q가 2048이므로 실제 11비트가 넘어서 저장되는 경우 일부 해밍 웨이트 불균형 문제가 발생할 수 있다. 따라서 해밍 웨이트를 높은 경우와 낮은 경우로 나누는 기준을 해당 중간 값의 하위 10비트를 기준으로 나누었다. DDLA 공격 모델의 세부적인 구조는 Table 1과 같다.

Table 1. Deep leaerning parameters for DDLA

Category	Specification	
Deep Learning Model (MLP)	Input layer	248 nodes
	Hidden layer	32 nodes (ReLU)
	Output layer	1 nodes (Sigmoid)
Loss function	Binary_Crossentropy	
Optimizer	Adam (lr=0.001)	
Labeling method	HW-based Binary classification	
Epoch / batch size	50 / 512	
Training / Validation ratio	0.8	
Scaling	Zero mean	

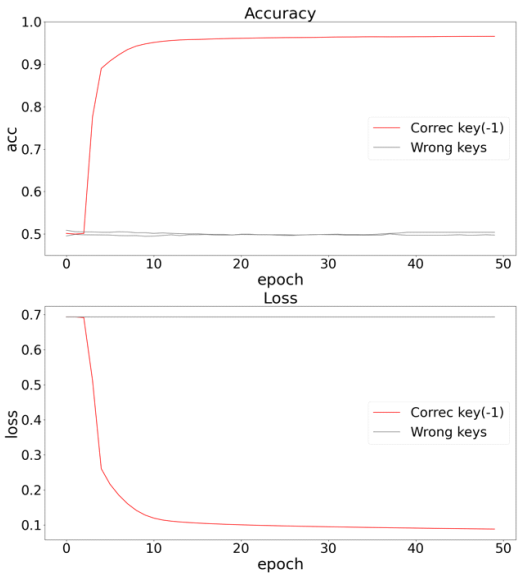

Fig. 6. Result of DDLA on polynomial multiplication of NTRU

DDLA 공격에서는 총 10,000개까지의 전력 파형으로 학습해 본 결과, 2,000개의 파형만으로 개인키를 충분히 복구해 낼 수 있음을 확인하였다. 다음 Fig. 6은 개인 키의 18번째 계수에 대해 DDLA 공격을 수행한 결과이다. 실제 키 값에 대한 결과는 높은 학습 정확도와 낮은 손실 값을 보이는 것을 확인할 수 있으며, 그렇지 않은 키는 낮은 학습률과 높은 손실 값을 보여 DDLA 공격을 통해 개인 키를 복구할 수 있음을 확인하였다.

IV. 전력 부채널 공격 대응 방안

4.1 셔플링 기법

SPA 공격과 CPA 공격 방어를 위해 사용되는 대표적인 방법으로는 개인 키의 각 계수가 사용되는 순서를 알 수 없게 하는 셔플링(shuffling) 기법이 있다[15]. 본 논문에서 수행했던 부채널 공격들을 적용해도 셔플링 기법을 적용한다면 개인 키를 복구하는 것은 불가능하다.

다음 Fig. 7은 NTRU 복호화 알고리즘 중 공격 지점인 poly_Rq_mul 함수에 Fisher-Yates 셔플링 기법을 적용한 코드이다. 즉, 다항식 곱셈의 결과 값의 계수를 구하는 순서를 셔플링함으로써 사용되는 개인 키의 계수 순서를 랜덤화 하는 것이다. 본 논문의 구현에서 중요한 점은 변수 $r[k]$의 계산 순서를 셔플링하는 것에 따라 사용되는 키의 계수 순서도 셔플링 되기 위해 다항식 c와 f의 곱셈 위치를 바꾸었다는 것이 중요하다. 즉, 그림에서 f의 값을 사용하는 순서가 셔플링 값이 저장된 $s[k]$값에 의존하도록 해야 한다.

Algorithm 2 : poly_Rq_mul_Shuffle(c, f, r)

Input : Secret Key f
Input : Ciphertext c
Output : Polynomial r

1. *for* $0 \le k < NTRU\_N$
2. $s[k] = k$
3. *Shuffle(s[k], NTRU_N-1)*
4. *for* $0 \le k < NTRU\_N$
5. $r[s[k]] = 0$
6. *for* $1 \le i < NTRU\_N - s[k]$
7. $r[s[k]] \mathrel{+}= f[s[k]+i] * c[NTRU\_N-i]$
8. *for* $0 \le i < s[k] + 1$
9. $r[s[k]] \mathrel{+}= f[s[k]-i] * c[i]$

Fig. 7. Polynomial multiplication with shuffling countermeasure

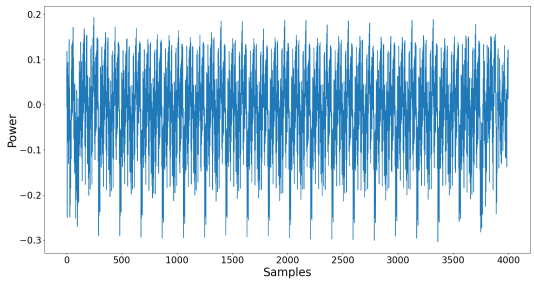

Fig. 8. Power trace of polynomial multiplication with shuffling countermeasure

위와 같이 셔플링을 적용한 후, 개인 키와 암호문이 연산되는 지점의 파형은 Fig. 8과 같다. 상기한 실험과 마찬가지로 개인 키가 곱셈이 되는 구간을 선정하고 CPA와 DDLA 공격을 수행하였다.

다음 Fig. 9는 셔플링 대응 기법을 적용한 후 CPA와 DDLA 공격을 수행한 결과를 비교한 것이다. 셔플링 대응책을 적용 전에는 CPA와 DDLA 공격을 통해 개인 키를 복구할 수 있었지만, 적용 후에는 두 가지 공격으로 개인 키를 복구할 수 없음을 확인하였다.

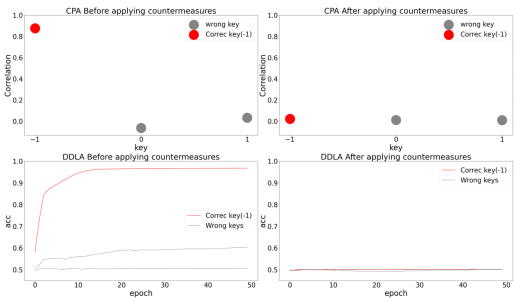

Fig. 9. Result of DDLA and CPA after applying Shuffling countermeasure

4.2 제안하는 대응 기법

상기한 셔플링 기법을 통한 대응책은 SPA와 CPA에 모두 대응할 수 있다는 장점이 있지만, 곱셈 연산 이전에 연산하는 개인 키의 위치를 랜덤하게 셔플링하는 과정에서 연산량이 증가한다는 단점이 있다. 특히, Fisher-Yates 셔플링 과정에서 N개의 랜덤 수를 발생해야 한다. 따라서 본 논문에서는 poly_Rq_mul 연산 시 적용 가능한 새로운 부채널 공격 대응 알고리즘을 제안한다.

기본적으로 NTRU 복호화 알고리즘 중 부채널 누출은 개인 키와 암호문의 곱셈 연산 과정에서 발생하게 된다. 이 연산에서 개인 키는 삼진 다항식이라는 사실에 주목할 필요가 있다. 제안하는 대응책에서는 SPA를 방어하기 위해 더미(dummy) 연산을 추가함은 물론 CPA나 DDLA 공격에 대응하기 위해 곱셈 연산에서 개인 키 계수가 -1과 곱하는 항, 1과 곱하는 항, 그리고 0과 곱하는 항을 구별한 후 암호문의 계수를 독립적으로 먼저 누산(accumulation)하도록 하였다.

암호문에 대한 누산을 한 후 0과 곱하는 누산 항은 버리고 1을 곱하는 누산 항에서 -1을 곱하는 누산 항을 빼면 된다. 제안하는 대응 알고리즘에서는 두 다항식의 곱셈 결과를 나타내는 결과 값의 각 계수는 다음과 같이 표현할 수 있다.

$$r(x) = c(x) \cdot f(x) \bmod (x^N - 1)$$
$$= \sum_{k=0}^{N-1} r_k \cdot x^k$$
$$r_k = \sum_{i=1}^{N-k-1} c_{k+i} \cdot f_{N-i} + \sum_{i=0}^{k} c_{k-i} \cdot f_i$$
$$r_k = \sum_{(i=1, f_{N-i}=1)}^{N-k-1} c_{k+i} - \sum_{(i=1, f_{N-i}=-1)}^{N-k-1} c_{k+i} + \sum_{(i=0, f_i=1)}^{k} c_{k-i} - \sum_{(i=0, f_i=-1)}^{k} c_{k-i}$$

제안하는 부채널 공격 대응 알고리즘을 나타낸 것이 Fig. 10이다. 여기서 누산 값도 공격자가 예측할 수 없도록 초기에 랜덤한 값을 저장한 후 연산이 끝

Algorithm 3 : poly_Rq_mul _proposal(c, f, r)

Input : Secret Key f
Input : Ciphertext c
Output : Polynomial r

```
1.  for 0 ≤ k < NTRU_N
2.      r[k] = 0, temp = 0, random[3] = {0,}, r_temp[3] = {0,}
3.      for 0 ≤ i < 3
4.          random[i] = r_temp[i] = rand( )% 2^16
5.      for 1 ≤ i < NTRU_N-k
6.          temp = f[NTRU_N-i]
7.          if(temp == 2047) r_temp[0]+=c[k + i]
8.          if(temp == 0) r_temp[1]+=c[k + i]
9.          if(temp == 1) r_temp[2]+=c[k + i]
10.     for 1 ≤ i < k + 1
11.         temp = f[i]
12.         if(temp == 2047) r_temp[0]+=c[k − i]
13.         if(temp == 0) r_temp[1]+=c[k − i]
14.         if(temp == 1) r_temp[2]+=c[k − i]
15.     r[k] = (r_temp[2] − random[2]) - (r_temp[0] - random[0])
```

Fig. 10. Polynomial multiplication with proposed countermeasure

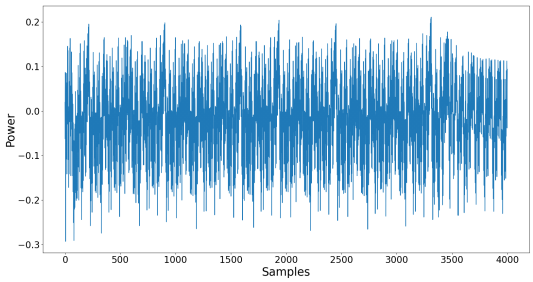

Fig. 11. Power trace of polynomial multiplication with a proposed countermeasure

난 후 이 랜덤 값을 제거하는 기법을 같이 사용하면 공격자의 중간 값 예측을 더 어렵게 할 수 있다.

개인 키 계수에 따른 암호문 누산이 종료되면, 해당하는 개인 키 계수가 1일때의 누산 값과 -1일때의 누산 값을 뺄셈을 하게 된다. 최종적으로 개인 키와 암호문의 곱셈 연산을 제거하고 누산과 덧셈으로 다항식 곱셈을 수행하게 된다. 따라서 개인 키는 처음 한 번만 읽어오게 되고 실제 연산에는 사용되지 않아 전력 소비 누출이 없어지게 된다. 제안 알고리즘을 사용하여 다항식을 곱셈을 수행할 경우의 전력 파형을 나타낸 것이 Fig. 11이다.

본 논문에서는 제안하는 대응 방식에 상기한 CPA나 DDLA 공격을 그대로 적용하여 보았다. 공격 지점을 설정하고 공격을 수행한 결과, Fig. 12와 같이 CPA와 DDLA 공격에 효과적으로 대응할 수 있음을 확인하였다.

제안 대응책의 수행 시간 측면에서의 비교를 위해 Intel(R) Core(TM) i3-7100 CPU @ 3.90GHz에서 C-언어로 구현한 결과, 대응책이 적용되지 않은 경우 키 생성에는 약 43.882ms, 암호화 과정에서는 약 0.833ms 복호화 과정에서는 약 2.178ms가 소요되었다. 그리고 셔플링 대응책을 적용했을 경우 복호화 시간은 약 2.337ms가 소요된 반면, 제안 대응책은 2.234ms가 소요되어 대응책이 없는 경우에 비해 약 2.6%정도 증가함을 확인하였다.

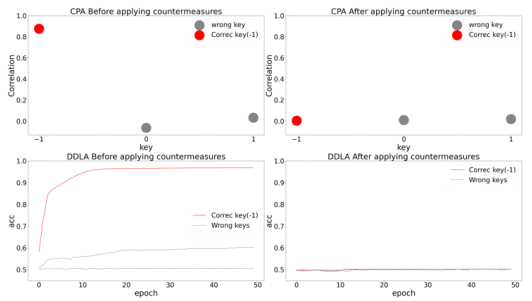

Fig. 12. Result of DDLA and CPA after applying a proposed countermeasure

V. 결 론

본 논문에서는 양자 내성 암호 NTRU 복호화 과정 중 발생하는 부채널 누출 정보를 이용하여 SPA, CPA, DDLA 공격을 수행하였으며, 그 결과 개인 키를 복구할 수 있음을 확인하였다. 특히, 개인 키가 삼진 다항식 구조로 되어 있다는 것이 NTRU에서 순차적으로 키의 계수를 복구하기가 용이하다는 취약성을 가지고 있다.

이와 같은 부채널 공격에 대응하기 위해서는 계산 과정을 셔플링하는 알고리즘을 사용할 수 있으나 매번 랜덤 수를 다항식 항의 수 만큼 발생해야 하여 연산량이 부담이 될 수 있다.

본 논문에서는 NTRU의 다항식 곱셈이 인덱스별로 두 다항식 계수를 곱셈 후 누산하는 것을 개선하였다. 제안하는 대응책은 개인 키의 계수에 따라 암호문을 먼저 누산 후 덧셈과 뺄셈만 수행하도록 함으로써 곱셈 연산에 따른 전력 정보 누출이 최소화되도록 설계함으로써 CPA 및 DDLA 공격을 방어할 수 있다.

References

[1] M. J. Dworkin, E. B. Barker, J. R. Nechvatal, J. Foti, L. E. Bassham, E. Roback, J. F. Dray Jr et al., "Advanced encryption standard (AES)," 2001.

[2] L. Grover, "A fast quantum mechanical algorithm for database search," ACM Symposium on Theory of Computing, STOC '96, pp.212-219, 1996.

[3] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM review, Vol. 41, No.2, pp. 303-332, Apr. 1999.

[4] G. Alagic et al., “Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process,” US Department of Commerce, National Institute of Standards and Technology, 2021.

[5] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, ″Relations among notions of security for public-key encryption schemes,″ Advances in Cryptology - CRYPTO′98, LNCS 1462, pp. 26-45, 1998.

[6] K. Ahmad, A. Kamal, A., K. Ahmad, K. A. B., M. Khari, and R. G. Crespo, ″Fast hybrid-MixNet for security and privacy using NTRU algorithm,″ Journal of Information Security and Applications, 2021.

[7] NIST, ″PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates,″ Available at https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4. 2022.

[8] E. Brier, C. Clavier, and F. Olivier, ″Correlation Power Analysis with a Leakage Model,″ CHES′04, LNCS 3156, pp. 16-29, 2004.

[9] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” CRYPTO’99, LNCS 1666, pp. 388-397, 1999.

[10] B. Timon, ″Non-Profiled Deep Learning-based Side-Channel attacks with Sensitivity Analysis,″ IACR Transactions on Cryptographic Hardware and Embedded Systems-TCHES ′19, Vol. 2019, no. 2, pp. 107-131, Feb. 2019.

[11] J. Song, D. Han, M. Lee and D. Choi, “Power analysis attacks against NTRU and their countermeasures,” Journal of the Korea Institute of Information Security & Cryptology, Vol. 19, No. 2, pp. 11-21, 2009.

[12] A. Askeland and S. Rønjom, “A Side-Channel Assisted Attack on NTRU,” IACR ePrint Archive, Available at https://eprint.iacr.org/2021/790, 2021.

[13] C. Chen, O. Danba, J. Hoffstein, A. Hulsing, J. Rijneveld, J. M. Schanck, P. Schwabe, W. Whyte and Z. Zhang, ″NTRU Algorithm Specifications And Supporting Documentation,″ Second PQC Standardization Conference, March 2019.

[14] NIST. “Round 3 Finalists: Public-key Encryption and Key-establishment Algorithms,” Available at https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions. 2022.

[15] M. Saarinen, “Arithmetic Coding and Blinding Countermeasures for Lattice Signatures,” Journal of Cryptographic Engineering. Vol. 8, No. 3, pp. 71-84, 2018.

〈저자소개〉

장 재 원 (Jaewon Jang) 학생회원
2022년 2월: 호서대학교 컴퓨터정보공학부 학사
2022년 3월~현재: 호서대학교 정보보호학과 학·석사연계과정
〈관심분야〉 인공지능 보안, 부채널 공격, 양자암호

하 재 철(Jaecheol Ha) 종신회원
1989년 2월: 경북대학교 전자공학과 학사
1993년 8월: 경북대학교 전자공학과 석사
1998년 2월: 경북대학교 전자공학과 박사
1998년 3월~2007년 2월: 나사렛대학교 정보통신학과 교수
2007년 3월~현재: 호서대학교 컴퓨터공학부 교수
2013년 1월~현재: 한국정보보호학회 상임부회장
2009년 1월~현재: 한국산학기술학회 이사
〈관심분야〉 정보보호, 네트워크보안, 부채널 공격