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.2022.32.6.1045

PQC 표준화 알고리즘 CRYSTALS-KYBER에 대한 비프로파일링 분석 공격 및 대응 방안*

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

Non-Profiling Analysis Attacks on PQC Standardization Algorithm CRYSTALS-KYBER and Countermeasures*

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

요 약

최근 양자 내성 암호 표준화 사업을 진행 중인 미국의 국립표준기술연구소는 표준화가 확정된 4개의 알고리즘을 발표하였다. 본 논문에서는 PKE/KEM 분야에서 표준화가 확정된 CRYSTALS-KYBER 알고리즘의 복호화 과정 중 비프로파일링 기반 전력 분석 공격인 CPA(Correlation Power Analysis)와 DDLA(Differential Deep Learning Analysis)에 의해 개인 키가 노출될 수 있음을 보이고자 한다. 실험 결과 개인 키의 일차 다항식 계수 복구에 성공하였으며, 특히 DDLA에서는 중간 값의 해밍 웨이트(Hamming Weight)를 라벨로 사용하는 모델에서 평가 기법인 NMM(Normalized Maximum Margin)의 값이 13.0으로 가장 높은 값을 가져 개인 키를 복구할 수 있는 것을 확인하였다. 또한, 복호화 과정 중 암호문을 랜덤하게 분할하고 계수별 곱셈 연산의 시작 지점을 랜덤화하는 방어 기법을 적용하면 상기한 공격을 방어하는 것을 확인하였다.

ABSTRACT

Recently, the National Institute of Standards and Technology (NIST) announced four cryptographic algorithms as a standard candidates of Post-Quantum Cryptography (PQC). In this paper, we show that private key can be exposed by a non-profiling-based power analysis attack such as Correlation Power Analysis (CPA) and Differential Deep Learning Analysis (DDLA) on CRYSTALS-KYBER algorithm, which is decided as a standard in the PKE/KEM field. As a result of experiments, it was successful in recovering the linear polynomial coefficient of the private key. Furthermore, the private key can be sufficiently recovered with a 13.0 Normalized Maximum Margin (NMM) value when Hamming Weight of intermediate values is used as a label in DDLA. In addition, these non-profiling attacks can be prevented by applying countermeasures that randomly divides the ciphertext during the decryption process and randomizes the starting point of the coefficient-wise multiplication operation.

Keywords: Post-Quantum Cryptography, CRYSTALS-KYBER, Power Analysis Attack, Deep Learning Analysis attack

Received(08. 29. 2022), Modified(1st: 10. 20. 2022, 2nd: 11 .16. 2022), Accepted(11. 17. 2022)
* 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2020R1F1A1074358)
† 주저자, wkdehreo55@naver.com
‡ 교신저자, jcha@hoseo.edu(Corresponding author)

I. 서 론

현재 상용되고 있는 컴퓨터는 트랜지스터로 구현된 여러 게이트를 이용해 정보를 0과 1의 비트(bit) 단위로 연산을 수행한다. 그에 반해, 현재 연구개발이 계속되고 있는 양자 컴퓨터는 양자역학을 연산 법칙으로 사용해 정보를 0과 1의 상태를 동시에 가지는 큐비트(qubit) 단위로 초고속 연산을 수행하며 기존의 컴퓨터로는 풀 수 없었던 문제들도 간단히 해결할 수 있게 된다. 이러한 양자 컴퓨팅 환경은 나날이 발전하고 있으며 현재 사용 중인 공개 키 암호시스템의 수학적 난제는 Shor가 제안한 양자 알고리즘에 의해 다항식 시간 안에 풀릴 수 있으므로 더는 안전성을 보장받을 수 없게 되었다[1].

미국의 국립표준기술연구소(NIST, National Institute of Standards and Technology)는 2016년부터 양자 컴퓨팅 환경에서 안전한 양자 내성 암호(PQC, Post Quantum Cryptography) 표준화 사업을 진행 중이며, 최근 3라운드가 종료되면서 표준화 확정 알고리즘이 발표되었다[2]. 공개 키 암호화(PKE, Public Key Encryption)/키 교환 메커니즘(KEM, Key Encapsulation Mechanism) 분야에서는 CRYSTALS-KYBER 알고리즘이 유일하게 확정되었으며, 전자서명(Digital Signature) 분야에서는 CRYSTALS-Dilithium, FALCON, SPHINCS$^+$로 총 3개 알고리즘의 표준화가 확정되었다. 또한, PKE/KEM 분야의 또 다른 후보 알고리즘 4개는 이어지는 4라운드를 진행하여 그 결과에 따라 추후 표준화가 진행될 예정이다.

1996년 P.Kocher에 의해 처음 제안된 부채널 분석(SCA, Side Channel Analysis) 공격은 암호용 디바이스에서 누설되는 부채널 정보를 기반으로 개인 키 등의 비밀 정보를 복구하는 공격을 의미한다[3]. 이때 이용하는 부채널 정보는 암호용 디바이스에서 암호 알고리즘이 동작할 때 정채널이 아닌 부가적인 채널을 통해 측정할 수 있는 정보이며 소비 전력, 전자기파, 연산 소요시간, 소리 등이 있다. 그 중, 암호용 디바이스 동작 시 발생하는 소비 전력 파형을 ChipWhisperer, 오실로스코프와 같은 장비를 이용해 수집한 뒤 분석하여 비밀 정보를 복구하는 전력 분석 공격(power analysis attack)이 부채널 분석 분야에서 가장 활발히 연구되고 있다[4, 5].

전력 분석 공격은 크게 프로파일링(profiling) 기반 전력 분석 공격과 비프로파일링(non-profiling) 기반 전력 분석 공격으로 나뉜다. 먼저 프로파일링 기반 전력 분석 공격은 공격 대상 디바이스와 동일하거나 비슷한 사양의 디바이스를 이용해 소비 전력 프로파일을 생성하고 공격 대상 디바이스에서 측정한 소비 전력 파형과의 비교를 통해 비밀 정보를 알아내는 공격이며 대표적으로 템플릿 공격(TA, Template Attack)[6]이 있다.

한편, 비프로파일링 기반 전력 분석 공격은 오직 공격 대상 디바이스에서 측정한 소비 전력 파형만을 이용하여 비밀 정보를 알아내는 공격이다. 비프로파일링 기반 전력 분석 공격에는 대표적으로 다수의 소비 전력 파형을 분석 기법을 이용하여 분석하는 차분 전력 분석(DPA, Differential Power Analysis) 공격[7], 상관 전력 분석(CPA, Correlation Power Analysis) 공격[8]이 있으며 딥러닝(deep learning)을 기반으로 한 DDLA(Differential Deep Learning Analysis) 공격[9]이 있다.

본 논문에서는 PKE/KEM 분야에서 표준화가 확정된 CRYSTALS-KYBER를 대상으로 비프로파일링 기반 전력 분석 공격인 CPA와 DDLA 공격을 시도하여 구현상의 취약점을 분석해 본다. CRYSTALS-KYBER는 송신자가 수신자의 공개 키로 평문을 암호화하고 수신자는 자신의 개인 키로 암호문을 복호화하게 되므로 공격은 복호화 단계에서 수행하여 개인 키를 복구한다. 이때 DDLA 공격은 CRYSTALS-KYBER 공격 지점의 중간 값 특징을 고려하였을 때 신경망 학습에 사용될 라벨(label)의 종류에 따라 공격 성능이 크게 달라질 것으로 예측하고 그 결과를 비교하였다. 또한, 비프로파일링 기반 전력 분석 공격에 대응할 수 있는 방어 기법을 제안하며 이 대응책이 부채널 누출을 제거하고 상기한 부채널 공격을 방어할 수 있음을 확인하였다.

II. 관련 연구

CRYSTALS-KYBER를 대상으로 한 부채널 공격은 현재까지 활발히 연구되고 있으며, 최근 표준화가 발표됨에 따라 해당 연구의 관심도는 더욱 높아질 것이라고 예상된다.

A. Karlov 등은 pqm4 라이브러리에서 구현된 CRYSTALS-KYBER의 1차 마스킹 기법을 제안하였다[10]. 마스킹 기법은 KYBER의 decapsulation 과정에 적용하였으며, 그중에서도 크게 복호화, 재 암호화, KYBER용 SHA3-512 해

시 G 그리고 암호문 비교 과정에 각각 적용하였다.

D. Heinz 등은 pqm4 라이브러리에서 구현된 CRYSTALS-KYBER에 대해 두 개의 basemul 연산을 진행하는 함수인 doublebasemul에서 첫 번째 basemul 연산의 smultt 명령어와 pkhtb 명령어를 공격 지점으로 하여 CPA 공격을 진행하였다[11]. 이때 smultt와 pkhtb 명령어는 ARM 내장 함수이며, 두 레지스터의 상위 16비트 곱셈을 진행하는 명령어인 smultt를 통해 비밀 키의 일차 항을 복구하고, 이어서 두 레지스터의 상위 16비트를 합치는 명령어인 pkhtb를 통해 비밀 키의 상수 항을 복구하는 실험을 진행하였다.

본 논문에서는 상기한 연구들에서 사용한 CRYSTALS-KYBER의 pqm4 라이브러리 코드가 아닌 공식 참조 코드를 이용하였으며 CPA 공격뿐만 아니라 또 다른 비프로파일링 공격인 DDLA를 시도하여 알고리즘의 안전성을 검증한다. 또한, 다항식 계수별 곱셈 연산 지점의 부채널 누출을 방어할 수 있는 새로운 방어 기법을 제안한다.

III. CRYSTALS-KYBER 알고리즘

CRYSTALS-KYBER는 CPA에 안전한 PKE 방식을 Fujisaki-Okamoto 변환을 통해 IND-CCA2 조건을 달성한 Module-LWE (Learning With Errors) 문제를 수학적 난제로 하는 격자(lattice) 기반 암호이다[12]. 1개의 환(ring)을 이용하는 Ring-LWE 문제는 보안 강도를 유지하기 위해 매우 큰 크기의 파라미터 n과 q를 사용하여야 하는 단점이 존재한다. 그에 반해 Module-LWE 문제는 상대적으로 작은 크기의 파라미터 n과 q를 사용하며 격자 차원을 n의 배수로 설정하는 파라미터 k의 값에 따라 보안 강도를 다양한 수준으로 설정할 수 있다는 장점이 존재한다.

Table 1. CRYSTALS-KYBER parameters

version	KYBER 512	KYBER 768	KYBER 1024
security level	1 (≈AES128)	3 (≈AES192)	5 (≈AES256)
n	256	256	256
k	2	3	4
q	3329	3329	3329
η_1	3	2	2
η_2	2	2	2
δ	2^{-139}	2^{-164}	2^{-174}
sk size	1632	2400	3168
pk size	800	1184	1568
c size	768	1088	1568

KYBER PKE 시스템에서는 다항식 환 $\mathcal{R}_q$를 수식 (1)과 같이 정의한다.

$$\mathcal{R}_q = \mathbb{Z}_q[x]/(X^n+1) \tag{1}$$

Table 1은 k값에 따른 KYBER 버전과 파라미터를 나타낸다. k값에 따라 보안 강도가 올라가는 것을 확인할 수 있으며, δ는 복호화 실패확률이다.

3.1 KYBER 키 생성 알고리즘

KYBER의 키 생성 알고리즘에서는 공개 키 pk와 개인 키 sk를 생성한다. 이때 공개 키에 포함되는 공개 행렬 A는 균등분포를 이용하여 샘플링된다. 공개 행렬 A의 차원은 $(k\times k)$로 이루어진다. 먼저 랜덤한 seed를 XOF(eXtendable-Output Function) 함수를 이용하여 랜덤 byte string을 생성한다. 이후 rejection sampling을 이용하여 공개 행렬 A를 생성한다. 키 생성에 사용되는 비밀 벡터와 에러는 PRF(Pseudo Random Function) 함수를 통해 샘플링한 뒤, 중심이항분포(CBD, Centered Binomial Distribution) 함수를 통해 생성한다. 이때 η_1 값이 2인 경우 (-3, 3) 사이의 값을 샘플링하며, η_1 값이 3인 경우 (-7, 7) 사이의 값을 샘플링한다. 또한, KYBER의 알고리즘에는 인코딩과 디코딩 과정이 존재한다.

인코딩은 메모리상에 표현된 다항식 $f=f_0+f_1X+\cdots f_{255}X^{255}$의 계수를 바이트 배열로 직렬화하는 과정이다. 이때 소수 $q=3329$는 약 12비트이므로 각 다항식의 계수는 16비트 자료형에 저장된다. 디코딩은 인코딩과 반대의 과정으로, 바이트 배열에 저장된 다항식 계수를 다항식으로 역직렬화한다. 최종적으로 공개 행렬 A, 비밀 벡터 s와 에러 e를 이용하여 수식 (2)와 같이 공개 키 pk와 개인 키 sk를 생성한다. 키 생성 알고리즘의 전체 과정은 Fig. 1과 같다.

KYBER.CPAPKE.KeyGen() : key generation from [13]

Output: Secret key $sk \in \mathcal{B}^{12 \cdot k \cdot n/8}$
Output: Public key $pk \in \mathcal{B}^{12 \cdot k \cdot n/8+32}$
1: $d \leftarrow \mathcal{B}^{32}$
2: $(\rho, \sigma) := \mathrm{G}(d)$
3: $N := 0$
4: **for** i from 0 to $k-1$ **do**
5: **for** j from 0 to $k-1$ **do**
6: $\hat{\mathbf{A}}[i][j] := \mathsf{Parse}(\mathsf{XOF}(\rho, j, i))$
7: **end for**
8: **end for**
9: **for** i from 0 to $k-1$ **do**
10: $\mathbf{s}[i] := \mathsf{CBD}_{\eta_1}(\mathsf{PRF}(\sigma, N))$
11: $N := N+1$
12: **end for**
13: **for** i from 0 to $k-1$ **do**
14: $\mathbf{e}[i] := \mathsf{CBD}_{\eta_1}(\mathsf{PRF}(\sigma, N))$
15: $N := N+1$
16: **end for**
17: $\hat{\mathbf{s}} := \mathsf{NTT}(\mathbf{s})$
18: $\hat{\mathbf{e}} := \mathsf{NTT}(\mathbf{e})$
19: $\hat{\mathbf{t}} := \hat{\mathbf{A}} \circ \hat{\mathbf{s}} + \hat{\mathbf{e}}$
20: $pk := (\mathsf{Encode}_{12}(\hat{\mathbf{t}} \bmod^{+} q) \| \rho)$
21: $sk := \mathsf{Encode}_{12}(\hat{\mathbf{s}} \bmod^{+} q)$
22: **return** (pk, sk)

Fig. 1. Key generation algorithm of KYBER PKE

$$pk : (A, t = A \bullet s + e)$$
$$sk : (s) \tag{2}$$

3.2 KYBER 암호화 알고리즘

KYBER의 암호화 알고리즘에서는 공개 키 pk와 메시지 m, 그리고 랜덤 값 r을 이용하여 암호문 c를 생성한다. 먼저 공개 키를 디코딩 과정을 거쳐 다항식 형태로 변환한 후 seed를 통해 공개 행렬 A를 재생성한다. 다음으로 암호문 생성에 필요한 r과 e_1, e_2는 PRF 함수를 통해 샘플링한 뒤, 중심이항분포 함수를 통해 생성한다. 이후 암호문 생성에 필요한 u, v 값을 수식 (3)과 같이 생성한다. 마지막으로 인코딩 과정을 거친 u, v를 이용하여 c_1, c_2를 생성하고 이를 연접하여 암호문 c를 생성한다. 암호화 알고리즘의 전체 과정은 Fig. 2와 같다.

$$(u, v) = (A^T r + e_1, t^T r + e_2 + \left\lceil \frac{q}{2} \right\rfloor \bullet m) \tag{3}$$

KYBER.CPAPKE.Enc(pk, m, r) : encryption from [13]

Input: Public key $pk \in \mathcal{B}^{12 \cdot k \cdot n/8+32}$
Input: Message $m \in \mathcal{B}^{32}$
Input: Random coins $r \in \mathcal{B}^{32}$
Output: Ciphertext $c \in \mathcal{B}^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$
1: $N := 0$
2: $\hat{\mathbf{t}} := \mathsf{Decode}_{12}(pk)$
3: $\rho := pk + 12 \cdot k \cdot n/8$
4: **for** i from 0 to $k-1$ **do**
5: **for** j from 0 to $k-1$ **do**
6: $\hat{\mathbf{A}}^T[i][j] := \mathsf{Parse}(\mathsf{XOF}(\rho, i, j))$
7: **end for**
8: **end for**
9: **for** i from 0 to $k-1$ **do**
10: $\mathbf{r}[i] := \mathsf{CBD}_{\eta_1}(\mathsf{PRF}(r, N))$
11: $N := N+1$
12: **end for**
13: **for** i from 0 to $k-1$ **do**
14: $\mathbf{e}_1[i] := \mathsf{CBD}_{\eta_2}(\mathsf{PRF}(r, N))$
15: $N := N+1$
16: **end for**
17: $e_2 := \mathsf{CBD}_{\eta_2}(\mathsf{PRF}(r, N))$
18: $\hat{\mathbf{r}} := \mathsf{NTT}(\mathbf{r})$
19: $\mathbf{u} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}}^T \circ \hat{\mathbf{r}}) + \mathbf{e}_1$
20: $v := \mathsf{NTT}^{-1}(\hat{\mathbf{t}}^T \circ \hat{\mathbf{r}}) + e_2 + \mathsf{Decompress}_q(\mathsf{Decode}_1(m), 1)$
21: $c_1 := \mathsf{Encode}_{d_u}(\mathsf{Compress}_q(\mathbf{u}, d_u))$
22: $c_2 := \mathsf{Encode}_{d_v}(\mathsf{Compress}_q(v, d_v))$
23: **return** $c = (c_1 \| c_2)$

Fig. 2. Encryption algorithm of KYBER PKE

3.3 KYBER 복호화 알고리즘

KYBER의 복호화 알고리즘에서는 개인 키 sk와 암호문 c를 이용하여 메시지 m을 복구한다. 먼저 암호문 c를 이용하여 암호화 과정에서 생성한 바이트 배열형태의 u, v와 개인 키 s를 다항식 형태로 변환한다. 이후 다항식 u는 NTT(Number Theoretic Transform) 변환을 거치며, 개인 키와 다항식 계수별 곱셈을 수행한 뒤 메시지 m을 복호한다. 이 과정을 수식으로 표현하면 수식 (4)와 같다.

$$m = Compress_q(v - s^T u, 1) \tag{4}$$

여기서 중요한 연산은 복호화 알고리즘을 나타낸 Fig. 3의 line 4에 표시된 것과 같이 개인 키 $\hat{s}$과 $\hat{u}$의 다항식 계수별 곱셈 연산이다. 이때 $\hat{u}$은 NTT가 적용된 다항식 u를 나타낸다. 즉 다항식 계수별 곱셈 연산에서는 비밀 정보인 개인 키와 공개 정보인 다항식 u가 사용되므로 공격자는 해당 지점의 전력 분석 공격을 통해 개인 키를 알아낼 수 있다.

$KYBER.CPAPKE.Dec(sk,c)$: *decryption* from [13]

Input: Secret key $sk \in \mathcal{B}^{12 \cdot k \cdot n/8}$
Input: Ciphertext $c \in \mathcal{B}^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$
Output: Message $m \in \mathcal{B}^{32}$
1: $\mathbf{u} := \mathsf{Decompress}_q(\mathsf{Decode}_{d_u}(c), d_u)$
2: $v := \mathsf{Decompress}_q(\mathsf{Decode}_{d_v}(c + d_u \cdot k \cdot n/8), d_v)$
3: $\hat{\mathbf{s}} := \mathsf{Decode}_{12}(sk)$
4: $m := \mathsf{Encode}_1(\mathsf{Compress}_q(v - \mathsf{NTT}^{-1}(\hat{\mathbf{s}}^T \circ \mathsf{NTT}(\mathbf{u})), 1))$
5: **return** m

Fig. 3. Decryption algorithm of KYBER PKE

IV. 비프로파일링 기반 전력 분석 공격 기법

4.1 CPA 공격

상관 전력 분석(CPA) 공격은 디바이스에서 측정한 소비 전력 파형과 예상 소비 전력 간의 상관 관계를 분석하는 공격 기법이다. 먼저 암호용 디바이스 안에 내장된 비밀 키를 사용하여 연산을 수행할 때의 소비 전력 파형을 측정한다. 이후 암호문과 모든 경우의 수로 가정한 비밀 키를 이용해 공격 지점 연산을 진행하여 중간 값(IV, Intermediate Value)을 계산한다. 다음으로, 얻어낸 중간 값에 전력 소비 모델을 적용해 예상 소비 전력을 계산한다. 일반적으로 전력 모델에는 해밍 웨이트(HW, Hamming Weight) 모델과 해밍 거리(HD, Hamming Distance) 모델이 존재하며 본 논문에서는 해밍 웨이트 모델을 사용하여 예상 소비 전력을 계산하였다. 최종적으로 디바이스에서 측정한 실제 전력 파형과 중간 값에 전력 소비 모델을 적용해 계산한 예상 소비 전력 간의 피어슨 상관 계수(PCC, Pearson Correlation Coefficient)를 구한다. 이때 올바른 키는 높은 PCC 값을 가지는 것을 통해 실제 비밀 키를 복구할 수 있다.

4.2 DDLA 공격

2019년 B.Timon이 처음 제안한 DDLA 기법은 비프로파일링 환경에서 딥러닝 모델의 학습을 진행한다. 이 기법은 가정한 비밀 키로 계산된 라벨을 이용하여 학습을 진행할 때의 학습 경향성을 이용하는 공격 기법이다. DDLA는 디바이스에서 측정한 소비 전력 파형을 입력으로 하고 중간 값 또는 중간 값을 이용해 계산한 값을 라벨로 사용하여 딥러닝 모델을 학습한다. 라벨로 이용될 수 있는 값은 중간 값(IV), 중간 값의 해밍 웨이트(HW), 중간 값의 최상위 비트(MSB, Most Significant Bit), 최하위 비트(LSB, Least Significant Bit) 그리고 해밍 웨이트 기반 이진(HW-based binary) 레이블[14]이 있다. 이때 올바른 키를 이용하여 계산한 중간 값은 수집한 파형과 연관성이 있고, 잘못된 키를 이용하여 계산한 중간 값은 수집한 파형과 연관성이 없을 것이다. 즉, DDLA 공격은 소비 전력 파형과 추측 키를 이용해 계산한 라벨을 이용하여 딥러닝 모델을 학습시킬 때 연관성이 있는 경우 학습이 잘 이루어지고, 연관성이 없는 경우 학습이 이루어지지 않는 특징을 이용하여 비밀 키를 찾는다.

DDLA 공격은 딥러닝 모델의 학습 경향성을 이용하는 기법으로 모델 설계 시 파라미터의 설정이 중요하며, 그중에서도 모델의 라벨로 주어지는 값의 설정이 매우 중요하다. J. Han 등은 AES-128 1라운드 Subbytes 연산을 대상으로 진행한 DDLA 공격에서 각 라벨의 분석 성능을 예측하고 검증하는 연구를 진행하였으며, HW 라벨을 사용한 것이 가장 좋은 분석 성능을 가지는 것을 보였다[15]. 본 논문에서 공격 지점으로 선정한 $fqmul$연산은 그 출력 값을 중간 값으로 사용하는데, 그 범위가 수식 (5)와 같아 각 라벨의 특성과 중간 값의 특성을 고려하여야 하므로 분석 성능을 예측한다.

$$fqmul\,(a,b) \in \left[-\frac{q-1}{2}, \frac{q-1}{2}\right] \tag{5}$$

먼저 HW는 중간 값 16비트를 모두 고려하여 총 17개(HW:0~16)의 클래스를 가진다. 이때 DDLA 공격이 계산된 라벨 값이 소비 전력 파형과 연관성이 있는지를 보이는 공격임을 고려하였을 때, HW를 라벨로 사용하면 전력 소비 모델을 통한 값을 그대로 사용하는 것이며, 이는 실제 소비 전력과 예상 소비 전력의 연관성을 비교하는 것이 되므로 가장 효과적일 것이라고 예상한다.

MSB는 중간 값의 최상위 비트를 라벨로 사용한다. KYBER의 중간 값은 부호가 존재함을 고려하였을 때 MSB 라벨로 중간 값을 분류하면 부호 비트에 따라 분류하는 꼴이 되므로 0은 양수, 1은 음수임을 분류하는 것이 된다. 따라서 MSB는 HW와 일정 부분 선형관계가 존재하므로 약간의 성능을 기대해볼 수 있지만, HW를 라벨로 선정했을 때와 비교해 분석 성능은 더 낮을 것으로 예상한다. LSB는 중간 값의 최하위 비트를 라벨로 사용한다. 이는

HW와 일정 부분 선형관계가 존재하지만, MSB와 같이 음수와 양수를 구분할 수 있는 것이 아니며, 총 16비트의 중간 값 비트 중 최하위 1비트만을 고려하는 것이므로 분석 성능은 가장 좋지 않으리라고 예상된다.

마지막으로 HW-based binary는 중간 값 비트 열에서 0보다 1의 개수가 더 많으면 라벨을 1로, 1보다 0의 개수가 더 많으면 라벨을 0으로 구분하는 방법이다. 이때, 중간 값의 모든 비트를 고려한 뒤 2개의 클래스로 분류하는 것이므로 17개의 클래스로 분류하는 HW와 비교해 분석 성능은 더 낮을 것이며 MSB와 비슷한 결과가 나타날 것으로 예측한다.

한편, 본 논문에서는 라벨에 원-핫 인코딩을 적용하지 않았다. 원-핫 인코딩이 적용된 라벨은 클래스 간 선형성이 사라지고 독립적인 관계가 된다는 특징을 가진다. 하지만, 본 실험에서 사용한 라벨은 모두 선형 관계가 존재하므로 독립적이지 않다.

V. 비프로파일링 기반 전력 분석 공격 실험

전력 분석 공격을 수행하기 위해서는 고정된 개인 키가 연산에 사용되는 지점을 찾는 것이 중요하다. CRYSTALS-KYBER에 대한 효과적인 전력 분석 공격 지점으로는 수신자가 복호화를 수행하기 위해 실질적으로 개인 키가 사용되는 가장 구체화된 연산인 계수별 곱셈 알고리즘 $basemul(a,b,zeta)$ 과정이 있다. 본 알고리즘의 입력값인 a는 NTT가 적용된 $Z_{3329}[x]/(X^2-zeta)$ 상의 2바이트 크기를 갖는 1차 다항식 개인 키 계수 2개이며 b는 위와 동일한 환경의 1차 다항식 암호문 계수 2개이다. 개인 키 및 암호문은 각각 $a=a[0]a[1]$와 $b=b[0]b[1]$로 표현된다. 이때 $a[0]$, $b[0]$은 상수 항이며 $a[1]$, $b[1]$은 일차 항이다. $fqmul$ 함수에 입력된 개인 키와 암호문의 계수는 서로 곱해진 후 Montgomery reduction을 거쳐 $r=r[0]r[1]$에 저장된다.

Fig. 4는 KYBER 복호화 과정 중 개인 키와 암호문의 계수별 다항식 곱셈 알고리즘의 과정을 나타낸 것이다. Fig. 4의 (1)~(3)에서 확인할 수 있듯이 계수별 곱셈 알고리즘 $basemul$은 $k\times128$번 진행한다. $basemul$ 알고리즘에서는 일차 다항식 계수 간의 곱셈을 진행하는데, 이때 상수 항 간, 일차 항 간의 곱셈은 Fig. 4의 네 번째 알고리즘 line 1~3에 해당하고 상수 항과 일차 항 간 곱셈은 Fig. 4의 네 번째 알고리즘 line 4~5에 해당한다. 따라서 본 실험에서는 $a[0]$와 $a[1]$이 사용된 $basemul$ 알고리즘을 대상으로 전력 분석 공격을 진행하여 개인 키의 1차 다항식 계수 $a[0], a[1]$을 복구하고 이와 같은 과정을 반복함으로써 전체 개인 키 정보를 찾아낼 수 있음을 확인하고자 한다.

전력 분석 공격 시 $fqmul$ 연산의 출력 값을 중간 값으로 사용하는 경우(Fig. 4의 네 번째 알고리즘 line 1, 2, 4), 올바른 키의 계수와 비슷한 상관 계수를 가지는 유사 계수가 존재한다[16]. 중간 값의 범위는 상기한 수식 (5)와 같으며, 이때 수식 (6)과 같이 $HW(-c)$는 $HW(\sim c)$와 선형관계에 있다. 즉, 올바른 상관 계수 c를 이용해 계산한 $HW(c)$는 양의 상관 계수를 가지며, 그에 대응하는 $HW(-c)$는 반대로 음의 상관 계수를 가진다. 따라서 CPA 공격을 진행하여 상관 계수를 계산하였을 때 위와 같은 규칙에 의해 양의 상관 계수를 가지는 추측 값은 후보 계수로 분류하고, 음의 상관 계수를 가지는 추측 값은 후보 계수에 대응하는 유사 계수로 분류한다.

$$\begin{aligned} HW(-c) &= HW((\sim c)+1) \approx HW(\sim c) \\ &= 16-HW(c) \end{aligned} \tag{6}$$

$\hat{s}^T \circ NTT(u)$ (1)

1 $mp =$ **polynomial_multiplication**$(c[0], sk[0])$
2 **for** $i = 1$ *to* k **do**
3 $bp =$ **polynomial_multiplication**$(c[i], sk[i])$
4 $bp = mp + bp$

polynomial multiplication (a, b) (2)

Input: Secret polynomial a
Input: Polynomial part of the ciphertext b
Output: Polynomial m
1 **for** $i = 1$ *to* $\frac{n}{4}$ **do**
2 $m[4\cdot i : 4\cdot i+4] =$ **doublebasemul**$(b[4\cdot i : 4\cdot i+4], a[4\cdot i : 4\cdot i+4], zetas[i])$

doublebasemul (a, b) (3)

Input: Array of 4x2 bytes a
Input: Array of 4x2 bytes b
Input: Constant zeta
Output: Array of 4x2 bytes c
1 $c[0:2] =$**basemul**$(a[0:2], b[0:2], zeta)$
2 $c[2:4] =$**basemul**$(a[2:4], b[2:4], -zeta)$

basemul $(a, b, zeta)$ (4)

Input : First factor of polynomial a
Input : Second factor of polynomial b
Input : Integer defining the reduction polynomial
Output: Polynomial r
1 $r[0] \leftarrow fqmul(a[1], b[1])$
2 $r[0] \leftarrow fqmul(r[0], zeta)$
3 $r[0] \leftarrow r[0] + fqmul(a[0], b[0])$
4 $r[1] \leftarrow fqmul(a[0], b[1])$
5 $r[1] \leftarrow r[1] + fqmul(a[1], b[0])$

Fig. 4. Coefficient-wise multiplication algorithm

본 실험에는 NIST에서 제공하는 공식 참조 코드인 Kyber512 버전을 구현하여 사용하였다. 소비 전력 측정 실험 환경은 Fig. 5와 같이 ARM-Cortex-M4 코어가 탑재된 32비트 프로세서인 STM32F3 MCU 상에서 알고리즘을 구현하고 ChipWhisperer Lite를 이용하여 공격 지점으로 설정한 *basemul* 연산의 소비 전력 파형 10,000개를 29.5MS/s 속도로 측정하였다. 이때 개인 키는 MCU 내부에 내장되어 있다고 가정한다.

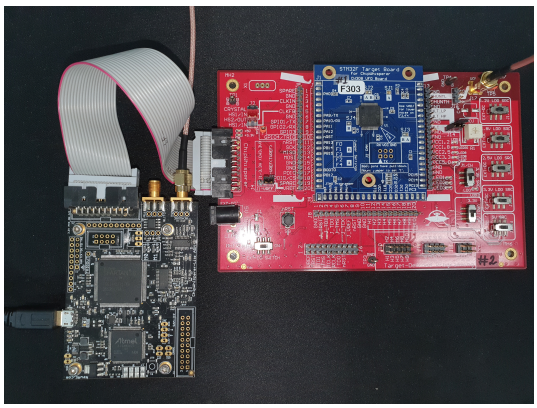


Fig. 5. Experimental setup using STM32F3 MCU and ChipWhisperer Lite platform

5.1 CPA 공격 결과

개인 키의 일차 항 계수 $a[1]$을 복구하기 위해 먼저 *basemul* 연산의 첫 번째 출력 값(Fig. 4의 네 번째 알고리즘 line 1)을 중간 값으로 하여 공격을 진행하였으며, $a[1]$의 범위는 0~3328이다. 이때 수식 (7)과 같이 가장 큰 상관 계수 값 $|PCC|_{\max}$를 가지는 계수의 상관 계수 값부터 $|PCC|_{\max}-0.3$ 사이의 값을 가지는 계수를 키 후보군으로 선정하며 키 후보의 개수가 1개일 경우 해당 키 후보를 실제 키로 판단한다.

$$|PCC|_{\max}-0.3 \le |PCC_{cand}|_{\max} \le |PCC|_{\max} \tag{7}$$

CPA 공격 결과, 상기한 키 후보군 선정 규칙에 따라 총 8개의 키 후보를 선정하였다. 이때, 음의 상관 계수 값을 가진 후보 계수 4개는 유사 계수 생성 규칙에 의해 후보 계수의 양의 상관 계수 값과 대칭

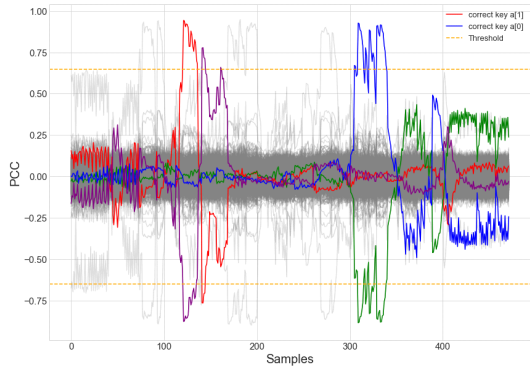


Fig. 6. CPA attack on the 1st output of basemul operation for all key candidates

인 음의 상관 계수 값을 가지는 유사 계수이다. Fig. 6은 *basemul* 연산의 첫 번째 출력 값에 대한 CPA 공격 결과이다. 키 후보군에 올바른 개인 키의 일차 항 계수 $a[1]$이 포함되었으며 대칭 형태의 상관 계수 값을 가지는 유사 계수가 존재하는 것을 확인할 수 있다. 또한, 동일한 암호문 계수 값과 연산을 진행하는 올바른 개인 키의 상수 항 계수 $a[0]$(Fig. 4의 네 번째 알고리즘 line 4) 역시 키 후보군에 포함된 것을 확인하였다.

다음으로 8개의 키 후보군에 대해 *basemul* 연산의 두 번째 출력 값(Fig. 4의 네 번째 알고리즘 line 2)을 중간 값으로 하여 공격을 진행하였다. *basemul* 두 번째 연산은 *basemul* 첫 번째 연산의 출력 값과 고정된 상수 *zeta*를 사용하므로 *basemul* 첫 번째 연산에 사용된 계수 값이 잘못된 값이라면 *basemul* 두 번째 연산의 중간 값 역시 잘못된 값이 된다. 그 결과 해당 중간 값은 이어지는 두 번째 CPA 공격 시 파형과의 상관도가 낮으므로 이 과정에서 실제 키와 그 유사 계수만을 구분할 수 있게 된다. Fig. 7은 *basemul* 두 번째 연산에 대한 CPA 공격 결과이다. 8개의 후보 키 중 2개의 후보 키만이 수식 (7)과 같은 키 후보군 선정 규칙을 만족하는 상관 계수 값을 가지는 것을 확인하였으며, 이는 올바른 계수 $a[1]$과 그에 대응하는 유사 계수 값이다.

마지막으로 2개의 후보 키 중 올바른 계수를 판단하기 위해 *basemul* 다섯 번째 연산(Fig. 4의 네 번째 알고리즘 line 5)의 출력 값을 중간 값으로 하여 공격을 진행하였다. 이때 올바른 개인 키의 상수 항 계수 $a[0]$가 사용된 연산(Fig. 4의 네 번째 알고리즘 line 4)이 선행되므로 상수 항 계수 $a[0]$를 먼

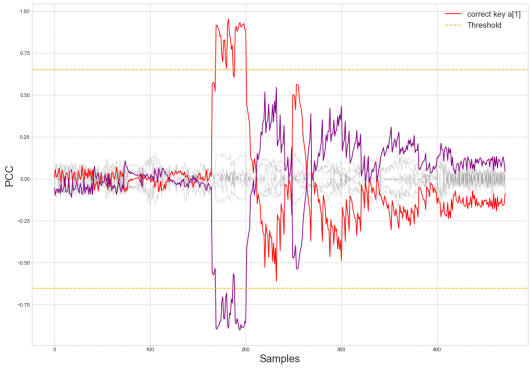


Fig. 7. CPA attack on the 2nd output of basemul operation for 8 key candidates

저 복구해야 한다. $a[0]$는 *basemul* 세 번째 연산의 출력 값과 $r[0]$가 더해진 값을 중간 값으로 하여 공격을 진행하여야 한다. 이때 $a[0]$는 상기한 첫 번째 CPA 공격 결과에서 알아낸 8개의 후보 키를 이용하며, 더해지는 $r[0]$ 값은 상기한 두 번째 CPA 공격 결과에서 알아낸 2가지 후보 키에 대해 각각 계산하여 이용한다. 본 실험에서는 상기한 과정에 따라 $a[0]$를 복구하였다고 가정하며, 최종적으로 $a[1]$을 복구하는 것을 보인다.

Fig. 8은 *basemul* 다섯 번째 연산에 대한 CPA 공격 결과이다. 올바른 계수 $a[1]$의 상관 계수 값이 가장 크며, 상기한 공격들에서 올바른 계수의 상관 계수 값과 대칭 형태의 상관 계수 값을 가지던 유사 계수의 상관 계수 값은 키 후보군 선정 규칙을 만족하지 못하는 것을 확인하였다. 이는 $a[1]$과 $b[0]$가 *fqmul* 연산을 통해 곱해진 후 네 번째 연산의 출력 값인 $r[1]$과 더해짐으로써 *fqmul* 출력 값에서 발생

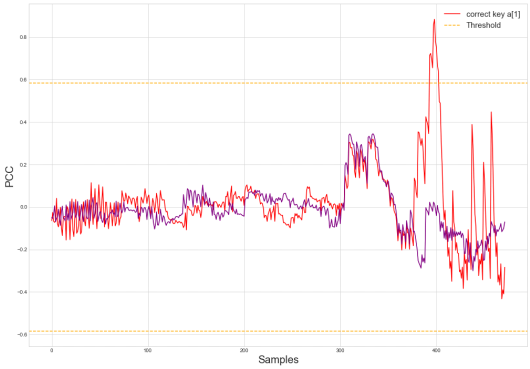


Fig. 8. CPA attack on the 5th output of basemul operation for 2 key candidates

Table 2. Comparison of $|PCC|_{\max}$ value on CPA attack result for basemul 5th output

	correct key	incorrect key
$\|PCC\|_{\max}$	0.8838	0.3450

하는 유사 계수 특징이 사라졌기 때문이다. 결과적으로 개인 키의 올바른 일차 항 계수 값 $a[1]$을 복구하는 데 성공하였으며, 올바른 키와 유사 계수의 $|PCC|_{\max}$ 값을 Table 2에서 비교하였다.

5.2 DDLA 공격 결과

본 실험에서는 개인 키의 일차 항 계수 $a[1]$을 DDLA 공격을 통해 복구하는 것을 보이기 위해 *basemul* 첫 번째 연산이 수행되는 지점의 파형을 이용하여 공격을 진행하였다. 공격에 사용된 딥러닝 모델은 가장 기본적인 신경망 모델인 다층 퍼셉트론(MLP, Multi-Layer Perceptron)을 사용하였다. 수집한 전력 파형을 모델의 입력 데이터로 사용하였으며, 라벨은 0을 제외한 모든 추측 키(1~3328)를 이용하여 생성한 HW, MSB, LSB, HW-based binary를 각각 사용하여 공격을 시도하였다. 단, IV는 본 실험에서 라벨로 사용하지 않았다. IV는 단사 함수 특성이 존재하며 이는 라벨의 경우의 수, 즉 출력층 노드의 개수가 기하급수적으로 늘어나게 되는 결과를 초래하므로 CRYSTALS-KYBER 대상 DDLA 공격에 적용할 수 없는 라벨로 설정하

Table 3. Detailed parameters for DDLA

Category	Specification	
Deep Learning Model (MLP)	Input layer	34 nodes
	Hidden layer	3 layers (ReLU)
	Output layer	17,2,2,2 nodes (Softmax)
Loss func.	Sparse_Categorical_Crossentropy	
Optimizer	Adam (lr=0.001)	
Labeling method	HW, MSB, LSB, HW-based binary	
Epoch / batch size	100 / 128	
Training / Validation ratio	0.7	
Scaling	Zero mean	

였다. 또한, 모델의 과적합 문제를 피하고 각 추측 키의 학습 능력을 일반화 성능으로 측정하기 위해서 validation loss를 학습 경향성 평가의 지표로 사용하여 그 성능을 분석하였다[17]. Table 3은 DDLA 공격 모델의 파라미터를 나타낸 것이다.

공격은 모든 라벨을 이용하여 수행하였으며, 그중 성능이 가장 좋을 것으로 예상한 HW를 라벨로 설정한 공격 결과를 확인한다. 공격 결과 Fig. 9와 같이 1개의 올바른 계수 값에서만 학습이 잘 진행되어 높은 정확도와 낮은 손실이 측정되고 나머지 후보 계수들은 학습이 일정 수준 진행되다가 더는 개선되지 않는 것을 확인하여 개인 키의 일차 항 계수 $a[1]$을 복구하는 데 성공하는 것을 확인하였다.

같은 방법으로 모든 라벨에 대해 DDLA 공격을 진행한 후 평가 기법인 NMM(Normalized Maximum Margin)[18]을 이용하여 공격 성능을 평가한다. NMM은 올바른 키와 잘못된 키의 학습 평가 지표 값의 차이를 표준편차 σ 단위로 나타낸 것이다. 만약 올바른 키의 NMM이 0보다 크다면 키를 찾을 수 있음을, 0보다 작다면 키를 찾을 수 없음을 의미한다.

다음 Fig. 10은 각 라벨 값을 이용해 학습한 모델의 공격 성능을 validation_loss를 이용해 계산한 NMM으로 평가한 결과이다. 100 epochs 지점에서 올바른 계수의 NMM 값이 HW는 13.0, MSB는 1.7, HW-based binary는 1.3으로 올바른 개인 키의 일차 항 계수를 복구하는 데 성공하였으며, LSB는 -0.6으로 복구에 실패한 것을 확인하였다.

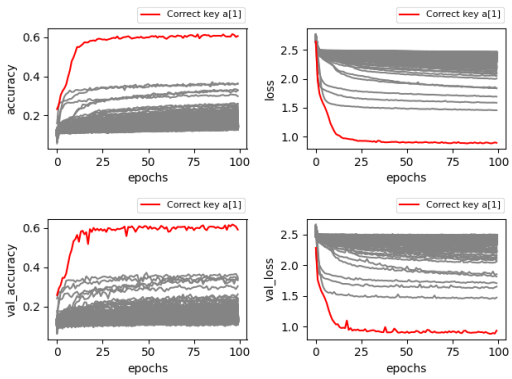
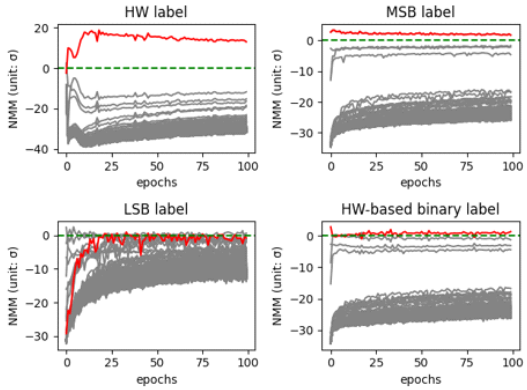
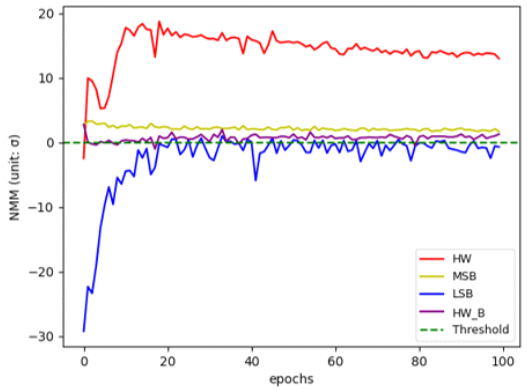


Fig. 9. DDLA attack on the 1st output of basemul operation using HW label for all key candidates



(a) NMM calculated in epochs for all key candidates



(b) NMM calculated in epochs for correct key a[1]

Fig. 10. NMM calculated according to each label

결론적으로 HW는 올바른 계수의 NMM 값이 0 이상이며, 다른 계수의 NMM 값과 큰 차이를 보여 DDLA 공격을 통해 올바른 일차 항 계수 $a[1]$을 복구하는 데 성공하였다. 반면, MSB와 HW-based binary는 올바른 계수의 NMM 값이 0보다 크긴 하지만 HW 라벨과 큰 차이를 보이며, 올바르지 않은 계수와의 NMM 값의 차이가 크게 나지 않았는데, 그 이유는 4.1절에서 진행한 CPA 공격 결과와 같이 후보 계수 및 유사 계수가 함께 측정되었기 때문이다. 실제로 올바른 계수 값을 제외한 3개의 계수 값이 다른 계수 값들에 비해 높은 NMM 값을 가지는 것을 Fig. 10-(a)에서 확인할 수 있다. 따라서 MSB와 HW-based binary를 이용하여 올바른 계수 값을 복구하기 위해선 CPA 공격 시나리오와 같이 후보 계수들에 대한 추가 공격을 진행하여야 하며, 추가 공격 시 비로소 올바른 계수 값을 복구할 수 있다. 이때 MSB와 HW-based binary를 라벨

로 진행한 DDLA 공격뿐만 아니라 HW를 이용한 CPA 공격에서도 존재하였던 유사 계수가 HW를 라벨로 이용한 DDLA 공격에서만 존재하지 않는 이유에 대해 정확히 분석하지 못하였으며, CPA와 DDLA 공격 과정의 차이에 의해 발생한다고 추측하였다.

VI. 부채널 공격 대응 방안

위의 5장에서 설명한 공격들은 KYBER 복호화 과정에서 개인 키와 암호문을 이용하여 진행되는 연산인 계수별 곱셈 알고리즘을 공격 지점으로 선정하여 진행되었다. 이때 암호문은 공격자가 알 수 있는 공개 정보이며, 공격자는 개인 키만을 추측하여 연산을 진행한 후 그 중간 값을 활용한다. 따라서 상기한 공격을 방어하기 위해서 두 가지 방어 기법을 제안한다.

첫 번째 기법은 공개 정보인 암호문을 복호화 과정 중 랜덤하게 두 개로 나누어 연산을 진행하는 것이다. 해당 기법을 적용하면 각 암호문과 개인 키에 대한 연산에서 처리되는 중간 값을 공격자가 예측하기 어렵게 되므로 공격을 방어할 수 있다.

두 번째 기법은 계수별 곱셈 연산에서 가장 먼저 계수별 곱셈을 수행하는 계수의 시작 위치를 랜덤하게 정하는 것이다. 이 경우 다항식 곱셈 연산을 시작하는 계수 위치가 매번 랜덤하게 변하므로 공격하려는 목표 계수의 연산 위치 역시 매번 달라지게 된다. 결과적으로 공격자는 소비 전력 파형을 특정 계수 값에 정렬하기 어려우므로 공격을 방어할 수 있다. Fig. 11은 기존의 복호화 과정과 부채널 공격 방어

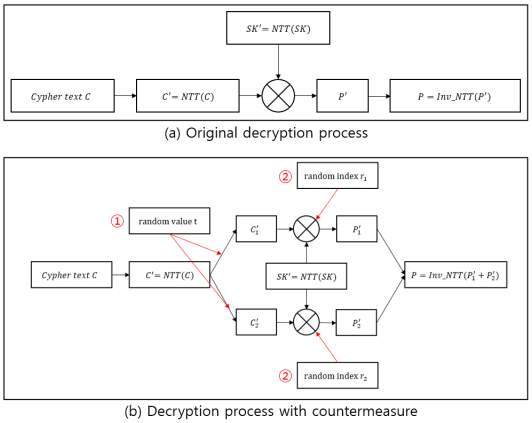


Fig. 11. Original decryption process and decryption process with countermeasures

기법을 적용한 복호화 과정을 각각 나타낸 것이다.

첫 번째 방어 기법은 Fig 11의 (b)-① 과정을 통해 이루어진다. 암호문에 NTT를 적용한 C'이 두 개로 나누어지는 것을 확인할 수 있는데, 이때 $-q < t < q$를 만족하는 $k \times n$개의 랜덤한 값 t를 구하여 C'의 각 계수에 대해 차분하여 암호문 다항식 C_1'을 생성하고, t 값을 계수로 하는 암호문 다항식 C_2'을 생성한다. 이후 두 개의 암호문 다항식을 각각 개인 키 다항식과 계수별 곱셈 연산을 진행한다.

두 번째 방어 기법은 Fig 11의 (b)-② 과정을 통해 이루어진다. 암호문 다항식과 비밀 키 다항식이 계수별 곱셈 연산을 진행할 때 $0 < r_1, r_2 < \frac{n}{4}$을 만족하는 랜덤한 인덱스 r_1, r_2를 이용해 연산이 시작될 첫 번째 계수의 인덱스를 랜덤하게 조정한다.

본 논문에서 제안한 두 가지 방어 기법은 알고리즘에 각각 적용할 수도 있고 모두 적용할 수도 있다. 먼저 첫 번째 기법만을 적용할 경우 분리된 암호문을 추측하기 위해서는 랜덤 값인 t를 알아내야 하는데, 한 개의 암호문 계수를 복구하기 위해서는 $(q-1) \times 2 + 1$개의 경우의 수를 가진 t를 알아내야 하며, 모든 암호문 계수를 복구하기 위해서는 $k \times n$개의 t를 알아내야 하므로 수많은 후보를 가져 모든 암호문 계수를 복구하기 어렵다. 하지만, 암호문을 두 개로 나누어 계수별 곱셈 연산을 두 번 진행하므로 기존의 복호화 방법과 비교해 연산 시간이 늘어난다는 단점이 존재한다. 본 실험에 사용한 KYBER512의 경우 $q = 3329, k = 2, n = 256$ 파라미터를 사용하는데, 이는 각각 소수 q, Module-LWE의 보안 강도를 나타내는 다항식 차원 k 그리고 다항식의 차수 n이다. 이때 랜덤 값 t는 6,657개의 경우의 수가 존재하며, 계수별 곱셈 연산이 두 번으로 나뉘어 진행되므로 비밀 키 계수 1개를 복구하는 데에는 방어 기법이 적용되지 않은 알고리즘을 공격할 때와 비교해 13,314배의 공격을 진행하여야 한다. 또한, 모든 비밀 키 계수를 복구하기 위해서는 2차원의 차수가 256인 다항식의 계수를 모두 복구하여야 하므로 6,657개의 경우의 수가 존재하는 t를 512개 찾아내야 하며, 두 번의 계수별 곱셈 연산을 모두 공격해야 하므로 모든 t를 찾아내려면 $6657 \times 512 \times 2$번의 추가 연산이 필요하다. 또한, KYBER 버전에 따라 더 높은 보안 강도의 k값을 가질수록 연산량도 비례하여 높아진다.

다음으로 두 번째 기법만을 적용할 경우

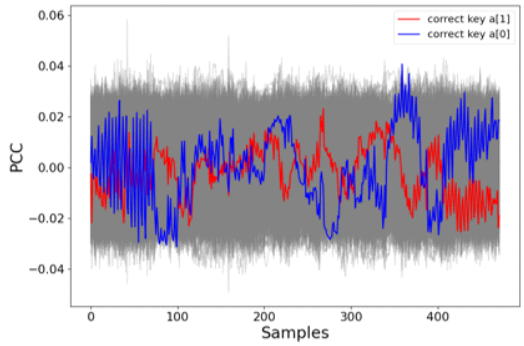
$f=f_0+f_1X+\cdots f_{255}X^{255}$ 형태인 다항식의 계수별 곱셈 연산에서 알고리즘의 시작 지점을 알아내기 위해선 $\frac{n}{4}$개의 경우의 수를 가진 k개의 랜덤 인덱스를 알아내야 한다. 이 경우 KYBER512는 모든 다항식 계수별 곱셈 연산의 시작 지점을 알아내는 데 최대 128번 반복만이 필요하다. 그럼에도 불구하고 CPA나 DDLA에서 사용되는 수 백개~수 천개 파형의 시작 지점을 모두 정렬할 가능성은 거의 없다. 또한, 이 방어 대책은 기존의 복호화 방법과 비교해 추가적인 반복문이 존재하지 않고 랜덤 인덱스를 이용하여 연산 순서만을 바꾸는 것이므로 추가적인 연산 시간이 발생하지 않는다는 장점이 있다.

마지막으로 두 가지 방어 기법을 모두 사용하는 경우 첫 번째 방어 기법의 수많은 랜덤 값 t를 알아내기 힘들 뿐만 아니라 다항식 계수별 곱셈 연산의 수가 2배로 늘어나므로 두 번째 방어 기법의 모든 랜덤 인덱스를 알아내기 어려워진다. 또한, 두 번째 방어 기법의 경우 연산 속도에 큰 영향을 주지 않기 때문에 두 가지 방어 기법을 모두 적용하여도 연산 시간은 첫 번째 방어 기법만을 적용하였을 때와 비슷하게 소요된다. Table 4는 모든 비밀 키 계수를 복구하는데 적용되는 randomness와 다항식 계수별 곱셈 알고리즘, 복호화 알고리즘이 각각 동작할 때 소요된 CPU 주기를 나타낸다. CPU 주기는 Intel i5-10500 (3.10GHz) CPU를 사용하는 PC에서 측정하였다.

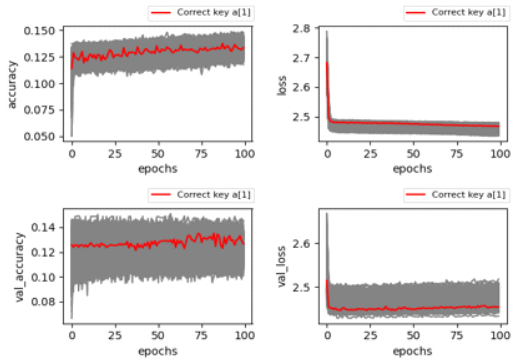
다음으로 5장에서 진행한 공격을 방어 기법이 모두 적용된 알고리즘을 대상으로 다시 진행하여 방어 기법을 검증하였다. Fig. 12는 CPA 공격과 HW를 라벨로 사용한 DDLA 공격 결과이다. 방어 기법 검증 실험 결과, CPA 공격에서 높은 PCC 값을 가졌던 올바른 계수 값들이 다른 계수 값들과 구분되지 않는 것과 DDLA 공격에서 올바른 계수가 다른 계수들과 구분되지 않는 것을 확인하였다. 따라서 본 논문에서 제안하는 방어 기법이 효과적으로 작동함을 검증하였다.

Table 4. Randomness and CPU cycles of KYBER512 with countermeasures

	method ①	method ②	method ①+②
randomness	6,816,768	128	1,745,092,608
CPU cycles	7,765 / 605,345	11,743 / 40,312	12,417 / 616,155



(a) CPA attack result



(b) DDLA attack result

Fig. 12. CPA and DDLA attack on the 1st output of basemul operation with countermeasures

VII. 결 론

NIST가 양자 내성 암호의 표준화 확정 알고리즘을 발표함에 따라 해당 알고리즘들을 대상으로 여러 연구가 더욱 활발히 진행될 예정이다. 본 논문에서는 현재 PKE/KEM 분야에서 유일하게 표준화가 확정된 CRYSTALS-KYBER 알고리즘의 복호화 과정 중 계수별 곱셈 알고리즘에서 부채널 누출이 존재함을 비프로파일링 기반 전력 분석 공격을 통해 증명하였다. 기존의 비프로파일링 공격인 CPA 공격과 딥러닝을 기반으로 한 DDLA 공격에서 모두 NTT가 적용된 개인 키의 계수를 복구하는 데 성공하였다. 특히, DDLA 공격에서는 여러 라벨을 이용하여 공격 성능을 평가하였으며 라벨로 HW를 이용할 경우 평가 기법인 NMM 값이 13.0임을 확인하여 개인

키 계수의 복구가 가능함을 보였다.

또한, 암호문을 랜덤하게 두 개로 분할하는 방법과 계수별 곱셈 알고리즘 연산의 시작 지점을 랜덤화하는 방법을 통해 해당 연산의 부채널 누출을 막을 방어 기법을 제안하였으며 이를 검증하였다. 따라서 CRYSTALS-KYBER 알고리즘을 적용할 때에는 비프로파일링 기반 전력 분석 공격과 같은 부채널 공격을 방어하기 위해 본 논문에서 제시하는 방어 기법과 같은 대응책을 이용해 부채널 누출을 제거한 뒤 암호 장비에 구현하여야 할 것이다.

References

[1] P. Shor, "Polynomial time algorithms for discrete logarithms and factoring on a quantum computer," SIAM Journal on Computing, Vol. 26, Issue 5, pp. 1484-1509, 1997.

[2] D. Moody, G. Alagic, D.A. Cooper, Q. Dang, T. Dang, J.M. Kelsey, J. Lichtinger, Y.K. Liu, C.A. Miller, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone and D. Apon, "Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process," National Institute of Standards and Technology, July. 2022.

[3] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," CRYPTO'96, LNCS 1109, pp. 104-113, Aug. 1996.

[4] T.S. Messerges, "Using second-order power analysis to attack DPA resistant software," International Workshop on Cryptographic Hardware and Embedded Systems, pp. 238-251, Aug. 2000.

[5] L. Goubin, "A refined power-analysis attack on elliptic curve cryptosystems," International Workshop on Public Key Cryptography, pp. 199-211, Jan. 2003.

[6] S. Chari, J.R. Rao and P. Rohatgi, "Template attacks," CHES'02, pp. 13-28, Aug. 2002.

[7] P. Kocher, J. Jaffe and B. Jun, "Differential power analysis," Advances in Cryptology, CRYPTO' 99, LNCS 1666, pp. 388-397, Aug. 1999.

[8] E. Brier, C. Clavier and F. Olivier, "Correlation power analysis with a leakage model," CHES'04, pp. 16-29, Aug. 2004.

[9] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis," IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 107-131, Feb. 2019.

[10] D. Heinz, M.J. Kannwischer, G. Land, T. Pöppelmann, P. Schwabe and D. Sprenkels, "First-Order Masked Kyber on ARM Cortex-M4," National Institute of Standards and Technology, Jun. 2021.

[11] A. Karlov and N.L. de Guertechin, "Power analysis attack on Kyber," Cryptology ePrint Archive, Sep. 2021.

[12] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J.M. Schanck, P. Schwabe, G. Seiler and D. Stehlé, "CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM," IEEE European Symposium on Security and Privacy, pp. 353-367, Apr. 2018.

[13] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J.M. Schanck, P. Schwabe, G. Seiler and D. Stehlé, "CRYSTALS-KYBER Algorithm Specifications And Supporting Documentation (version 3.02)," NIST PQC Round 3, pp. 1-43, Aug. 2021.

[14] D. Bae, J. Hwang, H. Lee and J. Ha, "Non-profiling deep learning

side-channel attack with Hamming weight-based binary labels", CISC-W'20, 2020.
〔15〕 J. Han, B. Sim, H. Lim, J. Kim and D. Han, "Design of an Effective Deep Learning-Based Non-Profiling Side-Channel Analysis Model," Journal of the Korea Institute of Information Security & Cryptology, 30(6), pp. 1291-1300, Dec. 2020.
〔16〕 S. Kim, Y. Kim, H. Moon, S. An, T. Lee, J. Han and D. Han, "Correlation Power Analysis of NTT Multiplication of NIST PQC Round 3 Finalist Candidate KYBER," CISC-S'21, pp. 510-514, Jun. 2021.
〔17〕 Y. Won, D. Han, D. Jap, S. Bhasin and J. Park, "Non-Profiled Side-Channel Attack Based on Deep Learning Using Picture Trace," IEEE Access 9, pp. 22480-22492, Feb. 2021.
〔18〕 D. Bae and J. Ha, "Performance Metric for Differential Deep Learning Analysis," Journal of Internet Services and Information Security (JISIS), Vol. 11, No.2, pp.22-33, May. 2021.

〈저자소개〉

장 세 창 (Sechang Jang) 학생회원
2022년 2월: 호서대학교 정보보호학과 학사
2022년 3월~현재: 호서대학교 정보보호학과 석사과정
〈관심분야〉 부채널 공격, 정보보호, 인공지능 보안

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
2009년 1월~현재: 한국산학기술학회 이사
2013년 1월~현재: 한국정보보호학회 상임부회장
〈관심분야〉 암호학, 부채널 공격, 네트워크 보안, 정보보호