KpqC 2라운드 후보 SMAUG-T의 키종속 연산에 대한 단순전력분석

유성환*, 한재승**, 한동국***

*,**,***국민대학교 (학부생, 대학원생, 교수)

Simple Power Analysis of key dependent operation in KpqC round 2 candidate SMAUG-T

Sunghwan Yoo*, Jaeseung Han**, Dong-Guk Han***

*,**,***Kookmin University(Undergraduate, Graduate student, Professor)

요 약

최근 양자 컴퓨터 기술의 발전으로 양자 내성 암호 (Post-quantum cryptography, PQC)에 대한 관심도가 높아지고 있으며 국내에서는 2022년도부터 국내 양자내성암호 국가공모전(Korean Post-Quantum Cryptography Competition, KpqC)을 시작하였다. 2024년 5월 기준으로 8종의 알고리즘이 2라운드에 선정되었고 평가 진행 중이다. 본 논문은 KpqC 2라운드 후보에 선정된 알고리즘 중 격자 기반 PKE/KEM 알고리즘인 SMAUG-T에 대한 신규 취약점 두 개를 보이고 비밀키 정보 누출 가능성을 설명한다. 복호화 연산 중 다항식 곱셈 연산이 상수시간을 만족하지 않음을 실험을 통해 보인다. 또한, 곱셈 연산에서 피연산자의 데이터 특성에 따라 소비전력의 차분이 식별됨을 보인다. 두 취약점을 활용하여 1, 0, -1로 구성된 비밀키의 1의 개수와 -1의 개수를 복구할 수 있음을 증명하고, 나아가 해당 취약점에 대한 대응기법 연구의 필요성을 제시한다.

I. 서론

최근 양자 컴퓨터 기술의 급격한 발전으로 RSA와 ECC 등 현재 사용되고 있는 암호 알고리즘이 위협을 받고 있다. 이에 양자 컴퓨터 출현 이후에도 안전한 암호 통신이 가능한 알고리즘인 양자 내성 암호 (Post-Quantum Cryptograpy, PQC)에 대한 연구가 활발히 진행되고 있다. 미국 국립표준기술연구소(NIST)에서는 2016년에 PQC 알고리즘 공모를 시작했으며 2024년 5월 기준 PKE(Public Key Encryption)/KEM(Key Encapsulation Mechanism) 알고리즘에서 CRYSTALS-Kyber가 표준으로 선정되었다. 국내에서도 2022년부터 KpqC 공모전을 시작하였고 현재 2라운드가 진행 중이며, PKE/KEM 알고리즘에서 4종, 전자서명 알고리즘에서 4종이 선정되어 평가되고 있다.

PQC가 고전 컴퓨터에서 공격 가능한 부채널 공격에 대한 저항성을 가지고 있는가는 다른 문제이기 때문에 NIST에서 표준으로 선정된 Kyber 알고리즘의 경우 현재까지 부채널 공격의 저항성을 만족하는지에 대한 연구가 활발히 진행되고 있다[1]. 따라서 국내에서 진행되는 KpqC 2라운드 알고리즘에도 부채널 공격에 대한 저항성 연구는 활발히 이루어져야 한다.

본 논문은 KpqC 2라운드 PKE/KEM 알고리즘 중 SMAUG-T 알고리즘[2]에서 비밀키를 사용한 다항식 곱셈 연산이 상수시간 연산을 만족하지 않음을 보이고, 비밀키 값에 따라 달라지는 연산에 대한 단순전력분석(Simple Power Analysis, SPA)가 가능함으로 보인다. 본 공격 기법을 통해 연산 과정에서 발생하는 소비전력 단일 파형만으로 비밀키의 일부 정보를 찾아낼 수 있음을 보인다.

[Conribution]

1. **(새로운 공격 제안)** 비밀키를 이용한 다항식 곱셈 연산이 상수시간을 만족하지 않음을

보인다. 또한, 피연산자의 값이 1과 -1에 따라 연산하는 소비전력의 차분이 식별됨을 보인다. 두 취약점을 활용하여 SPA분석을 통한 공격 기법을 제안한다.

2. **(공격 결과)** ARM 기반 STM32F3 환경에서 복호화 연산이 동작하는 동안 필요한 소비전력을 측정했다. 1에서 제안한 기법을 활용해 SPA 분석을 했고 비밀키의 다항식 별 1과 -1의 총 개수를 유출할 수 있을 뿐만 아니라 1의 개수와 -1의 개수까지 복원됨을 증명하였다.

II. 배경지식

2.1. SMAUG-T 알고리즘

SMAUG-T는 격자(Lattice) 기반 키 교환 알고리즘으로 KpqC 1라운드 후보 중 SMAUG와 Tiger를 결합하여 만든 알고리즘이다. 격자 기반에서도 모듈(module) 격자에서 정의되는 MLWE와 MLWR 문제를 기반하고 있으며 키 생성에 있어 sparse ternary 개념을 함께 사용하고 있다. SMAUG-T는 보안 강도에 따라 128, 192, 256로 구분되어 있으며 [표 1]은 보안 강도별 설정되는 파라미터를 설명한다.

[표 1] 파라미터 의미

기호	의미	128	192	256
k	모듈 개수	2	3	5
q	라운딩 전 계수 범위	1024	2048	2048
p'	라운딩 후 계수 범위	32	256	64

키 생성의 경우 MLWE 문제를 기반으로 키를 생성하며, 암·복호화의 경우 MLWR 문제를 기반으로 동작한다. [표 2]는 복호화 알고리즘의 의사 코드로 입력값인 비밀키 s와 파라미터 t, p, p'의 라운딩 연산을 통해 메시지 m 값을 복구한다.

2.2 sparse ternary 특징

sparse ternary는 비밀키 생성에서 사용되며, 1, 0 ,-1 값으로만 다항식 계수로 사용하고 1과 -1의 개수가 주어진 헤밍웨이트 만큼만 할당하여 희소성을 부여하는 기술이다.

[표 2] SMAUG-T 복호화 알고리즘

Input : pk = $(seed_A, b)$, m, $seed_r$
Output : ct = (c_1, c_2)
1. $A = \textbf{expandA}(seed_A)$
2. if $seed_r$ is not given then $seed_r \xleftarrow{\$} \{0,1\}^{256}$
3. r $\leftarrow HWT_{h_r}(seed_r)$
4. $c_1 = \lfloor p/q \cdot A \cdot r \rceil$
5. $c_2 = \lfloor p'/q \cdot \langle b, r \rangle + \frac{p'}{t} \cdot m \rceil$
6. **return** ct = (c_1, c_2)

비밀키 생성에 있어 n개 다항식의 총 해밍웨이트 정보는 공개된 정보이지만, 다항식 하나의 해밍웨이트 정보는 랜덤하게 생성되어 비밀 정보로 유지된다. 이러한 특징들은 비밀키 저장 방식과 다항식 곱셈 연산에 변화를 주어 다른 격자 기반 암호와 차별점을 가지게 한다. 본 절에서는 sparse ternary 개념을 이용한 비밀키 저장 방식과 다항식 곱셈 연산에 관해 설명한다.

1) 비밀키 저장

sparse ternary 특징으로 인해 비밀키의 다항식 계수는 1, 0, -1만 존재한다. 이에 대해 비밀키 저장 방식에서는 계수가 1과 -1인 항의 차수만을 저장하며 계수가 1인 항의 차수를 먼저 저장한 후 -1인 항의 차수를 추가로 저장한다.

2) 다항식 곱셈

비밀키의 다항식 계수가 1, 0, -1만이 존재하기에 곱셈 연산은 덧셈과 뺄셈만으로 구현된다. [표 3]은 Reference Code에서 정의된 다항식 곱셈 함수인 poly_mult_add 함수이다. [표 3]의 입력값 중 a는 비밀키와 곱셈 연산을 수행할 다항식이며 b는 256차 다항식의 비밀키 정보가 저장된 배열을 의미한다. neg_start 변수는 비밀키에서 계수가 -1인 항의 차수가 처음 저장된 배열의 인덱스 값을 의미한다. 따라서 neg_start 이전의 저장된 차수 항은 계수가 1이기에 곱셈 연산이 덧셈으로 대체되고, line 2-5의 구현 부분이 해당 연산을 나타낸다. neg_start 이후에 저장된 차수 항의 계수는 -1이므로 뺄셈 연산으로 대체된다. line 6-9의 구

현 부분이 뺄셈 연산을 나타낸다.

[표 3] poly_mult_add 함수

Input : a, b, neg_start
Output : c
1. $c = 0$
2. for i from 0 to neg_start - 1 do
3. degree = $b[i]$
4. for j from neg_start to $n - 1$ do
5. c[degree + j] = c[degree + j] + $a[j]$
6. for i from neg_start to **len(b)** - 1 do
7 degree = $b[i]$
8 for j from 0 to $n - 1$ do
9 c[degree + j] = c[degree + j] - $a[j]$
10 for j from 0 to $n - 1$ do
11 $c[j] = c[j] - c[n + j]$
12 return c

III. SMAUG-T 다항식 곱셈 SPA

3.1 공격 논리

2.2절에서 설명한 sparse ternary 특징으로 인해 SMAUG-T는 비밀키 생성 시 1과 -1의 개수 합이 파라미터로 명시된 해밍웨이트 만큼의 부여된다. SMAUG-T128 기준, 비밀키는 256차 다항식 2개로 구성되어 있고, 512개의 계수 중 140개의 계수가 1과 -1로 구성된다. 키 생성에서 140이라는 해밍웨이트 정보는 공개된 정보이지만, 각 256차 다항식에서 구성되는 해밍웨이트 정보는 랜덤하게 형성 비밀로 유지된다.

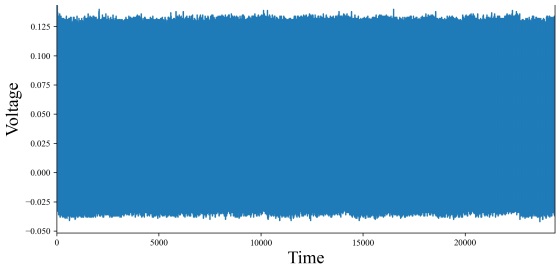
하지만 [표 3]의 다항식 곱셈 연산은 비밀키 배열인 b의 길이에 따라 연산 시간이 변하고 b의 길이는 키 생성마다 랜덤하게 형성되기에 poly_mult_add 함수는 상수시간 연산을 만족하지 않는다. 또한, 해당 함수를 구성하는 덧셈 연산과 뺄셈 연산은 정해진 상수시간을 만족하기에 곱셈 연산 전체의 파형과 덧셈 혹은 뺄셈 연산 1회 전력 파형을 수집하면 SPA 분석을 통해 랜덤하게 형성되는 다항식 별 해밍웨이트 정보를 알아낼 수 있다.

추가적으로 곱셈 연산 시 다항식의 계수에 따라 수행되는 함수가 다른 지점을 이용하면 다항식 별 해밍웨이트 정보뿐만 아니라 계수가 1인 항의 차수 개수와 -1인 항의 차수 개수를 알아낼 수 있다.

3.2 공격 결과

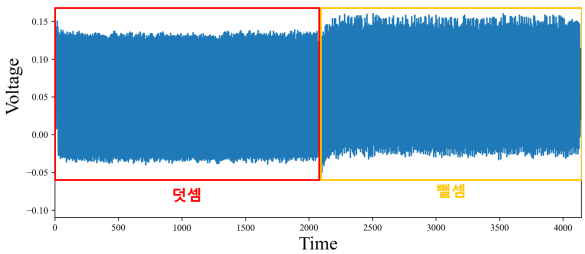
본 논문의 실험 환경은 전력 파형 수집에 있어 ChipWhisperer-Lite 보드와 CW308UFO STM32F303 MCU를 사용했다. 공격 대상 코드는 KpqC 2라운드 참조구현 C 코드 중 [표 3]에서 소개된 poly_mult_add 함수를 사용했다.

선택된 암호문과 키 한 쌍 중 비밀키 256차 다항식 곱셈 연산에 대해 덧셈과 뺄셈 함수 전 구간을 수집한 결과 수집 포인트는 100,964포인트가 수집되었다. [그림 1]은 수집한 파형 일부분을 나타낸다.



[그림 1] poly_mult_add 함수 파형 일부분

[그림 2]는 덧셈과 뺄셈 함수가 1회씩 동작하는 동안 수집한 전력 파형이다. 파형 수집 결과 4,144포인트가 수집되었고 SPA 결과 덧셈과 뺄셈 연산당 약 2,050포인트 정도 사용됨을 알 수 있다. 따라서 전체 100,964포인트를 2,050포인트로 나눈 값이 덧셈과 뺄셈 연산의 총횟수이자 랜덤하게 생성되는 해밍웨이트 값으로 유추된다.

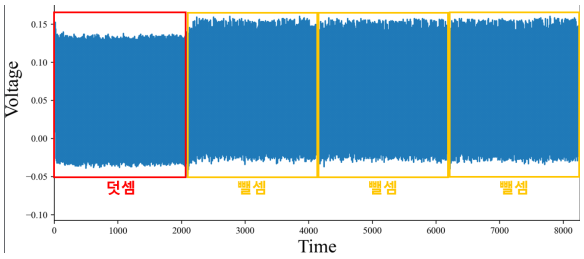


[그림 2] 덧셈과 뺄셈 1회 수행 시 파형

실제 실험에서 선택한 비밀키의 다항식 중 첫 번째 다항식의 해밍웨이트를 확인한 결과 49로 정확히 일치함을 확인했다.

더 나아가 계수가 1인 항의 차수 개수와 -1인 항의 차수 개수가 복구되는지 확인하기 위

해 다항식의 해밍웨이트를 4로 설정하고 neg_start 값을 1로 설정하여 파형을 수집하였다. 해당 값으로 설정함에 따라 곱셈 연산 시 덧셈 연산이 1회, 뺄셈 연산이 3회 수행된다. 곱셈 연산을 수행하는 동안 수집된 파형을 SPA 기법으로 분석한 결과 2100포인트를 기준으로 두 파형의 전압 차이가 발생하는 특이점을 확인했다. 또한, 덧셈 혹은 뺄셈 함수의 1회 연산 시 약 2050포인트가 사용됨을 3.2에서 확인했기에 [그림 3]과 같이 4개의 구간으로 파형이 분리됨을 확인했다. [그림 3]에서 분리된 파형을 통해 소비전력 차이가 작은 1개의 파형은 덧셈 연산 1회를 의미하며, 소비전력 차이가 큰 3개의 파형은 뺄셈 연산 3회를 의미하는 것을 파악했다.



[그림 3] 파형 SPA 결과

해당 정보는 앞서 조정된 비밀키 정보와 같음으로 공격자는 복호화 연산에서 수집된 파형의 SPA 분석을 통해 비밀키 다항식의 1의 개수와 -1의 개수를 도출할 수 있다.

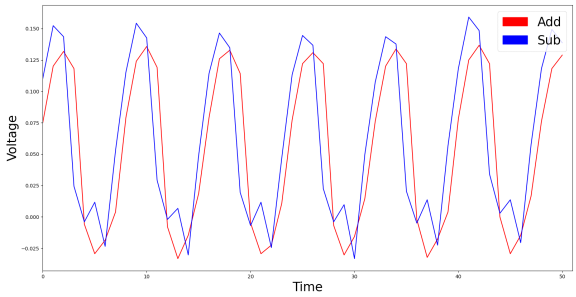
SPA 분석에 있어 더 명확한 수치를 기반으로 분석하고자 파형의 극대, 극솟값을 통해 소비전력 차이의 평균을 계산한 결과 [표 4]와 같이 덧셈 연산에서는 약 0.162, 뺄셈 연산에서는 약 0.094 정도의 소비전력 차이가 발생함을 확인했다. [그림 3]에서 파형 소비전력 차이가 뺄셈 연산이 더 큰 값으로 확인되지만, 평균 소비전력 차이가 낮은 이유는 [그림 4]와 같이 덧셈과 달리 뺄셈은 파형 사이에 소비전력 차이가 미세한 파형이 수집되기에 평균값이 작아진다. 수치화된 소비전력 차이를 기준으로 파형을 구분한 결과 [표 5]과 같이 성공적으로 SPA를 수행하여 정보를 도출하는 것을 확인했다.

[표 4] 덧셈과 뺄셈 연산에서의 소비전력 차

연산	소비전력 차이
덧셈	0.16324745
뺄셈	0.09461675

[표 5] 수집 파형 SPA 분석 결과

파형 번호	전력 차이	연산 구분	연산 횟수
1	0.16274414	덧셈	1
2	0.09422652	뺄셈	3
3	0.09363014	뺄셈	
4	0.09471872	뺄셈	



[그림 4] 덧셈과 뺄셈의 소비전력 차이

IV. 결론

본 논문은 KpqC 2라운드 PKE/KEM 알고리즘 중 SMAUG-T 복호화 연산이 상수시간을 만족하지 않음과 피연산자의 데이터 특성에 따라 소비전력의 차분 공격이 가능함을 분석하고 ARM 기반 STM32F3에서 실험을 진행하였다. 그 결과 다항식 1의 개수와 -1의 개수 정보까지 복구됨을 확인하였다. 향후 본 논문에서 제기된 취약점의 대응 방안 연구가 필요하다.

[사사] "본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터사업의 연구결과로 수행되었음" (IITP-2024-RS-2022-00164800)

[참고문헌]

[1] Jeonghwan Lee, et al. "A key recovery side-channel attack on KpqC 1 round candidate SMAUG" 정보보호학회 2023

[2] Jung Hee Cheon, et al. "SMAUG-T: the Key Exchange Algorithm based on Module-LWE and Module-LWR" KPQC 2024