Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.5, Oct. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.5.1089

NTRUEncrypt에 대한 단일 파형 기반 전력 분석*

안 수 정,† 김 수 리, 진 성 현, 김 한 빛, 김 희 석,‡ 홍 석 희
고려대학교

Single Trace Side Channel Analysis on NTRUEncrypt Implementation*

Soojung An,† Suhri Kim, Sunghyun Jin, HanBit Kim, HeeSeok Kim,‡ Seokhie Hong
Korea University

요 약

양자 컴퓨터의 개발이 가시화됨에 따라 RSA, Elliptic Curve Cryptosytem과 같은 암호 시스템을 대체할 수 있는 포스트 양자 암호에 대한 연구가 활발히 진행되고 있다. 하지만 이러한 포스트 양자 암호조차도 실 환경에서 구동될 때 발생할 수 있는 부채널 분석에 대한 취약점을 반드시 고려해야만 한다. 본 논문에서는 NIST 포스트 양자 암호 표준 공모에 제출된 NTRU 구현 소스인 NTRUEncrypt에 대한 새로운 부채널 분석 기법을 제안한다. 기존에 제안되었던 부채널 분석 방법은 많은 수의 파형을 이용하였지만 제안하는 분석 기술은 단일 파형을 이용한 분석 방법으로 공개키 암호 시스템에 실질적으로 적용이 가능하다. 또한 본 논문에서는 제안하는 분석 기술에 안전성을 제공할 수 있는 새로운 대응 기술을 제안한다. 이 대응 기법은 과거 NTRU 부채널 분석 기법에도 안전할 뿐만 아니라 기존의 NIST에 제출된 코드보다 더 효율적 구현이 가능한 알고리즘을 제안한다.

ABSTRACT

As the development of quantum computers becomes visible, the researches on post-quantum cryptography to alternate the present cryptography system have actively pursued. To substitute RSA and Elliptic Curve Cryptosystem, post-quantum cryptography must also consider side channel resistance in implementation. In this paper, we propose a side channel analysis on NTRU, based on the implementation made public in the NIST standardization. Unlike the previous analysis which exploits a thousands of traces, the proposed attack can recover the private key using a single power consumption trace. Our attack not only reduces the complexity of the attack but also gives more possibility to analyze a practical public key cryptosystem. Furthermore, we suggested the countermeasure against our attacks. Our countermeasure is much more efficient than existing implementation.

Keywords: Side Channel Analysis, Single Trace Analysis, Post Quantum Cryptography, NTRUEncrypt

I. 서 론

현재 가장 널리 사용되고 있는 공개키 암호 시스템인 RSA와 타원 곡선 암호 (Elliptic Curve Cryptography, ECC)는 양자 컴퓨터가 개발될 경우 소인수 분해 문제와 이산대수 문제를 다항 시간 안에 푸는 쇼어 알고리즘에 취약한 것으로 알려져 있다 [1]. 최근, 양자 컴퓨터 개발이 가시화됨에 따라 미국의 표준 기술 연구소 (National Institute of Standards and Technology, NIST)는 양자 컴퓨터로도 풀리지 않을 포스트 양자 암호 (Post-quantum cryptography) 표준을 공모하

Received(08. 28. 2018), Modified(09. 19. 2018), Accepted(09. 28. 2018)
* 이 성과는 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구 재단의 지원을 받아 수행된 연구임(No. NRF-2017R1C1B2004583).
† 주저자, soojung02@korea.ac.kr
‡ 교신저자, 80khs@korea.ac.kr(Corresponding author)

고 있다.

포스트 양자 암호는 격자 기반 암호(Lattice-based cryptography), 다변수 기반 암호(Multivariate-based cryptography), 해쉬 기반 암호(Hash-based cryptography), 코드 기반 암호(Code-based cryptography), 그리고 아이소제니 암호(Isogeny-based cryptography)로 분류되며, 이 중 격자 기반 암호는 다른 포스트 양자 암호에 비해 속도가 빠르고 합리적인 키 사이즈를 가지기 때문에 가장 활발히 연구되고 있다. 실제로 NIST 표준 공모에 제출된 전체 82개 암호 알고리즘 중 28개 알고리즘이 격자 기반 암호이다 [3].

1995년 Hoffstein 등이 제안한 NTRU는 격자 기반 문제 중 Shortest Vector Problem (SVP)에 기반한 공개키 암호 알고리즘으로 오랫동안 안전성이 증명되어왔으며 ECC, RSA보다 암복호화 속도가 빠르고 코드의 구현이 가볍기 때문에 작은 기기에도 효율적으로 사용될 수 있어 NIST 표준으로 선정될 수 있는 유망한 후보 중 하나이다 [2].

하지만 이론적으로 안전한 것으로 알려진 암호 알고리즘조차도 암호 연산 중, 예기치 못한 전력, 전자파 등의 부가 정보 누출에 의해 비밀 정보가 드러나는 부채널 분석에 대한 안전성을 반드시 고려하여야만 한다. 부채널 분석에 대한 안전성을 고려하지 않을 경우 비밀 정보가 쉽게 드러날 수 있음이 많은 연구 결과로 증명되었으며 NIST 표준 공모에서도 부채널 분석에 대한 안전성을 요구하고 있다 [15].

본 논문에서는 NIST에 제출된 NTRU 구현 코드(NTRUEncrypt) [5]에 대한 단일 파형 기반 부채널 분석 방법을 제안한다. 이전 NTRU를 분석한 논문의 경우에는 많은 수의 파형을 이용하여 개인키를 복구했다. 하지만 공개키의 경우 세션 키를 교환하는 데에 주로 사용되어 부채널 분석에 활용할 수 있는 파형의 개수는 대다수 경우 적은 수로 한정되어 있다. 따라서 실질적으로 적용이 불가능한 기존 공격 기술에 비해 한 개의 파형만을 활용하는 본 논문의 분석 기술은 효과적으로 적용이 가능하다.

또한 본 논문의 대응기술은 NIST에 제출된 구현보다 효율적이면서 기존의 분석 기법 [10]에도 안전한 알고리즘을 제안한다. 과거 분석들은 NIST에 제출된 구현 방법과 다른 알고리즘을 분석한 것이지만 동일한 공격이 적용가능하다. 기존의 공격에도 안전하며 새로운 공격에도 안전하며 연산량이 적은 대응기법을 제안하고 있다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 분석에 필요한 배경 지식을 소개한다. NTRU 알고리즘과 구현에 대하여 설명하고, 이전 NTRU를 분석한 논문들에 대하여 간단하게 소개한다. 다음으로 3장에서는 NTRUEncrypt를 분석하고 실험 결과를 소개하고 있다. 다음으로 4장에서는 분석에 대한 대응 방법을 제안하고 있으며, 5장은 결론을 낸다.

II. 배경 지식

2.1 NTRU 알고리즘

NTRU는 SVP에 기반한 공개키 암호 알고리즘으로 양자 컴퓨터에서도 다항 시간 안에 풀리지 않는 것으로 알려져 있다. 암호화 및 복호화 과정은 환 $R=Z[X]/(x^N-1)$ 위에서 연산한다. R의 원소는 계수가 정수인 $(N-1)$차 다항식이며 원소 $f\in R$는 $f=\sum_{i=0}^{N-1} f_i x^i$로 표현한다. 환 R 위에서 곱셈 연산 $(\cdot)$은 수식 (1)과 같이 표현한다.

$$\begin{aligned} f\cdot g &= h, \\ h_k &= \sum_{i=0}^{k} f_i g_{k-i} + \sum_{i=k+1}^{N-1} f_i g_{N+k-i} \\ &= \sum_{i+j\equiv k(\mathrm{mod}\,N)} f_i g_j \end{aligned} \tag{1}$$

NTRU의 알고리즘에 사용되는 파라미터 값들은 다음과 같다.

- N : 환 R 위의 다항식의 차수
- a : 양의 정수
- p : "작은" 모듈러 값
- q : "큰" 모듈러 값
- L_a : $\{f\in R: f$의 계수가 $a+1$개의 1, a개의 -1, 나머지 계수는 0으로 이루어진 다항식의 집합 $\}$
- B_a : $\{g\in R: g$의 계수가 a개의 1, a개의 -1, 나머지 계수는 0인 다항식 집합

p, q는 $\gcd(p,q)=1$, $p<<q$를 만족하는 정수이다 [2]. 위의 계수는 표준값으로 공개되어있다.

2.1.1 키 생성

개인키 f는 집합 L_f에서 선택된 삼진 다항식이며, 공개키 h는 $h=pf_q^{-1}\cdot g(\mathrm{mod}q)$, $g\in B_g$이다. 여기서 삼진 다항식이란 다항식의 계수가 -1, 0, 1의 세 가지 값으로만 이루어진 다항식을 의미한다.

f_q^{-1}는 f의 $Z_q[X]/(x^N-1)$에서의 역원을 의미한다. 복호화 과정에서 f_p^{-1}가 사용되는데, 마찬가지로 f_p^{-1} 또한 f의 $Z_p[X]/(x^N-1)$에서의 역원이다.

2.1.2 암호화

평문 $m\in B_m$을 암호화 하기 위해서는 수식 (2)와 같은 연산을 한다.

$$e=r\cdot h+m(\mathrm{mod}q) \tag{2}$$

r는 $r\in B_r$을 만족하는 임의의 다항식이며 h는 공개키이다. 다항식 곱셈은 환 R 위에서 연산하며, 모듈러 q 연산은 다항식의 계수를 모듈러 q를 하는 것을 의미한다. 이 모듈러 연산의 표현은 복호화에서도 동일하게 표현된다.

2.1.3 복호화

암호문 e는 수식 (3)과 (4)와 같은 단계를 통해 복호화된다.

$$a=f\cdot e(\mathrm{mod}q) \tag{3}$$

$$m=a\cdot f_p^{-1}(\mathrm{mod}p) \tag{4}$$

위의 식은 수식 (5)와 (6)의 과정을 통해 옳게 복호화됨을 확인할 수 있다.

$$\begin{aligned} a&=f\cdot e(\mathrm{mod}q)\\ &=r\cdot h\cdot f+m\cdot f\\ &=pr\cdot g+m\cdot f(\mathrm{mod}q) \end{aligned} \tag{5}$$

$$\begin{aligned} a\cdot f_p^{-1}&=(pr\cdot g+m\cdot f)\cdot f_p^{-1}(\mathrm{mod}p)\\ &=m\cdot f\cdot f_p^{-1}(\mathrm{mod}p)\\ &=m(\mathrm{mod}p) \end{aligned} \tag{6}$$

기존의 알고리즘대로는 (5)와 (6)의 과정을 통해 복호화를 하지만, 개인키 f를 특정 조건에 맞게 설정함으로 복호화를 빠르게 할 수 있다 [4]. 예를 들어 f를 $pF+1$, $F\in L_f$가 되도록 선택하면 f_p^{-1}는 1이 될 것이다. 그러므로 복호화 과정 중 수식 (4)에 해당하는 단계를 생략할 수 있다. 본 논문의 분석 대상인 NIST에 제출된 NTRU 구현에서도 이와 같은 방법을 사용하고 있다.

2.2 NTRU에 대한 부채널 분석 이전 연구

최초의 부채널 분석은 1996년 Kocher 등의 시간 정보를 이용한 분석이다 [6]. 시간 정보를 이용한 분석 방법에 이어서 전력 파형을 이용한 단순 전력 분석(Simple Power Analysis, SPA), 차분 전력 분석(Differential Power Analysis, DPA), 그리고 상관 전력 분석(Correlation Power Analysis, CPA)이 등장했다 [7, 8]. 위의 분석 기법들을 이용하면 암호의 키를 실질적인 시간 안에 복구할 수 있어 유의미하다. 이와 같은 분석에 대응하기 위해 마스킹과 하이딩 기법들이 연구되고 있다 [9]. 마스킹이란 비밀 정보를 난수값으로 가려 연산하는 방법을 의미하고, 하이딩이란 소비 전력이 데이터에 의존하는 성질을 가리는 것을 의미한다.

NTRU를 부채널 분석한 최초의 논문은 2007년 논문으로, 시간 정보를 이용하여 분석한다 [11]. 이후 2009년 Song 등이 CPA를 적용했다 [10]. [10]에서는 기본적인 SPA 방법을 제안하였고, 1,000개의 파형을 이용해 CPA로 분석됨을 증명하였다. 더불어 대응 방법으로 기본적인 마스킹과 하이딩 방법 중 셔플링 기법을 제안하고 있다. 이를 이어 [12]에서는 [10]의 대응 방법을 적용한 알고리즘을 10,000개의 파형을 이용해 이차 상관 전력 분석(Second Order Correlation Power Analysis)을 통해 분석한다. 마찬가지로 [10]의 대응 방법을 적용한 알고리즘을 2013년 Zheng 등이 일차 충돌 공격 방법 (First-order Collision Attack)으로 5,000개의 파형을 이용해 분석한다 [13].

NTRU의 구현은 과거에 특허가 되어 있어 비공개 자료였으나 2017년에 공개되었다. NTRU를 분석한 이전 논문의 경우 참고한 구현 방법과 현재 공개되어 있는 것은 차이가 있다. 본 논문에서는 공개된 공식 구현을 분석하고 있으며 또한 하나의 파형을 이용하여 개인키를 복구한다.

III. NTRU 구현에 대한 단일 파형 부채널 분석

3.1 NTRU 구현

본 논문에서는 공개키 암호 시스템의 비밀 정보인 개인키 복구를 목표로 한다. 그러므로 본 장에서는 개인키가 사용되는 복호화 과정의 알고리즘을 소개한다. 여기서 설명하는 복호화 알고리즘은 여러 NTRUEncrypt 구현 중에서 NIST에 제출된 알고리즘을 소개한다 [5].

먼저 NTRUEncrypt는 다항식을 표현할 때 계수를 차수 순서대로 배열에 저장하고 있다. 예를 들어, 다항식 $F(x)=x^3+x-1$의 경우 코드에서 배열 F에 $F=\{-1,1,0,1\}$와 같이 저장한다. 그리고 본격적인 다항식 곱셈 연산에 앞서 삼진 다항식 형태로 저장된 F를 $f=pF+1$로 전환하는 과정이 있다. 그러고 나면 계산된 f를 이용해 수식 (3)에 해당하는 암호문과 개인키 다항식 곱셈을 한다. 다항식 곱셈은 기본적인 곱셈 방법인 Grade School Multiplication으로 연산한다. 곱셈 과정 후 마지막으로 $\mathrm{mod}(x^N-1)$을 처리해주면 메시지를 얻을 수 있다. 이 과정은 Algorithm 1과 같다.

Algorithm 1. Decryption in NTRUEncrypt

Input: Trinary polynomial $F \in L_f$, ciphertext $e \in R$
Output: $m = f \cdot e \pmod q$

1. for $0 \le i < N$ do
2. $\quad f_i \leftarrow F_i \times p$
3. end for
4. $f_0 \leftarrow f_0 + 1$
5. for $0 \le j < N$ do
6. $\quad t_j \leftarrow e_0 \times f_j$
7. end for
8. for $1 \le i < N$ do
9. $\quad t_{i+N-1} \leftarrow 0$
10. $\quad\quad$ for $0 \le j < N$ do
11. $\quad\quad\quad t_{i+j} \leftarrow t_{i+j} + e_i \times f_j$
12. $\quad\quad$ end for
13. end for
14. $t_{2N-1} \leftarrow 0$
15. for $0 \le i < N$
16. $\quad m_i \leftarrow (t_i + t_{i+N}) \bmod q$
17. end for
18. return m

3.2 제안 방법

제안하는 공격 방법은 Algorithm 1의 단계 1에서 3에 해당하는 연산과 5부터 13까지의 연산 과정의 소비 전력을 이용하여 삼진 다항식 F를 복구한다. F가 복구되고 나면 개인키 f는 $pF+1$ 연산을 통해 알아낸다. 먼저 1부터 3까지에 해당하는 F를 f로 바꾸는 연산 중에 개인키 다항식의 계수 중 -1의 상대적인 위치를 알아낼 수 있다. F는 삼진 다항식이므로 동일한 값인 p와 연산되는 값은 -1, 0, 1 중 한 개의 값이다. 대부분의 프로세서는 음수 표현을 위해서 2의 보수 방법을 사용하기 때문에 -1의 해밍 웨이트(Hamming weight)는 매우 클 것이다. 그러므로 -1 값이 연산되는 부분에서는 소비 전력이 가장 높게 뜨는 것을 확인할 수 있을 것이다. 이때 소비 전력이 높게 뜨는 파형의 위치를 하이픽(high peak)이라고 한다.

다음으로 5부터 13에 해당하는 암호문(e)와 개인키(f)의 연산 과정에서는 계수 0의 상대적인 위치를 알아낼 수 있을 것이다. 암호문의 계수와 0이 연산된 결과는 다른 연산 값보다 소비 전력 값이 낮을 것이다. 이렇게 소비 전력이 낮게 나타나는 부분은 로우픽(low peak)이라고 한다. 그러므로 다른 로우픽의 위치를 분석하여 0의 상대적인 위치를 알아내고, 앞서 알아낸 -1의 정보를 합치면 삼진 다항식 F를 복구해 낼 수 있다. 그러면 최종적으로 $pF+1$ 연산을 통해 개인키 f를 복구할 수 있다.

3.3 실험 결과

Fig. 1은 NTRUEncrypt 코드를 Atmega128 SCARF 보드 [14, 16]에 올려, Lecroy HDO6104A 오실로스코프를 이용해 250M sampling rate로 수집한 파형이다. 이때 사용한 파라미터 값은 차수 $N=49$, $p=3$, $q=2048$ 그리고 사용한 개인키 값은 $f=\{1,1,1,1,1,0,1,0,-1,-1,1,1,0,0,1,1,-1,-1,1,0,0,0,-1,0,-1,-1,1,-1,1,1,0,0,0,-1,0,0,-1,0,1,0,1,-1,0,0,-1,-1,1,1\}$을 사용한다. p와 q 값은 표준에서 사용하는 값이며 N은 실험 환경에 따라 표준의 차수보다 작은 값을 사용한다. 아래 그림의 세로축은 소비 전력 값을 의미하며 가로축은 시간을 의미한다.

Fig. 3은 Algorithm 1의 단계 1부터 3에 해당하는 파형을 확대한 것이다. 앞서 분석한 것처럼 하

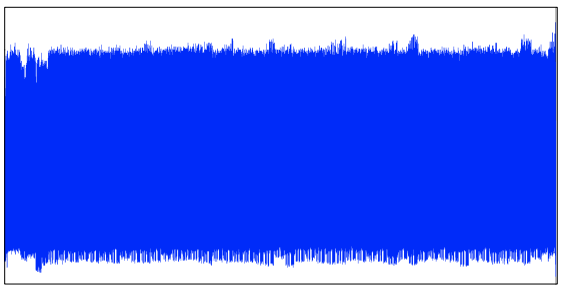


Fig. 1. Decryption of NTRUEncrypt

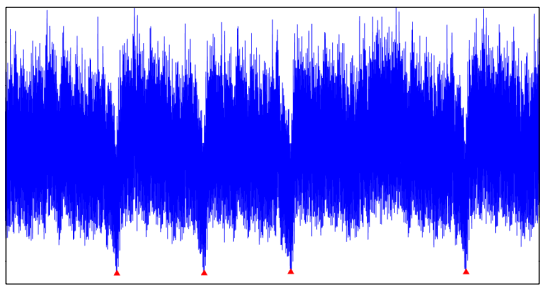


Fig. 2. Enlarged Low Peaks

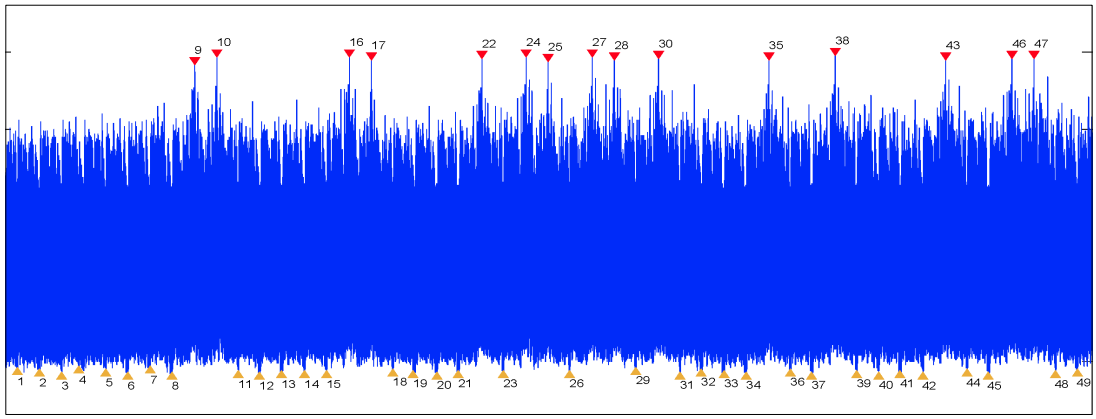


Fig. 3. Result of SPA against $f = pF + 1$

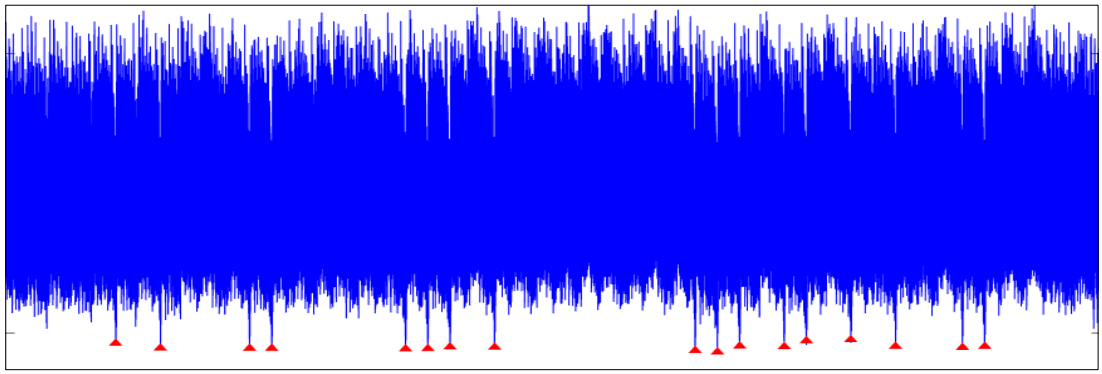


Fig. 4. The Average of 10 Power Consumption Traces of Polynomial Multiplication using Grade School Multiplication Method

이픽이 나타나는 값은 -1과 p의 연산 부분임을 알 수 있다. 또한 같은 그림에서 로우픽을 관찰할 수 있는데 이 값은 계수가 0과 1일 때 연관된 결과이다. 그러므로 Fig. 3의 결과를 통해 -1과 0, 1의 순서를 결정할 수 있다.

이어서 다음 과정을 분석해 본다. 암호문 계수 하나당 개인키 연산이 N번씩 반복된다. N번의 연산마다 개인키 0의 연산은 동일한 순서로 나타나기 때문에 전체 파형(Fig. 1)을 보면 로우픽이 규칙적으로 나타난다. 개인키 0의 위치를 복구하기 위해서는 먼저 전체 파형에서 암호문 하나의 항과 개인키의 연산에 해당하는 파형을 찾아내야 한다. 이 과정은 단순

전력 분석을 통해 계수 하나의 연산을 찾을 수 있으며 방법은 다음과 같다. 먼저 다항식 연산은 $pF+1$ 연산 이후일 것이므로 앞서 분석한 파형 이후를 시작점으로 생각한다. 그리고 곱셈 과정에 해당하는 연산은 총 N^2번 있을 것이므로, 규칙성을 보이는 파형 전체를 N^2으로 나눠 암호문 계수 하나의 연산에 해당하는 파형을 찾아갈 수 있다. 이런 파형 하나를 분석하여 로우픽을 분석할 수도 있지만 한 개의 파형에서 다항식 곱셈 파형을 여러 개를 겹쳐 평균을 내면 로우픽을 더 확실히 구분할 수 있다. Fig. 4는 위의 분석을 통해 알아낸 여러 세트의 평균 파형이다. 앞서서 분석한 -1과 0, 1의 상대적인 위치를 분석한 결과를 활용하여 0의 상대적인 위치를 알 수 있다. 또한 Fig. 2는 4개의 로우픽과 로우픽이 뜨지 않은 파형을 확대한 그림으로 로우픽의 파형의 구분이 됨을 확인할 수 있다. 최종적으로 -1과 0, 1의 순서를 합치고 나면 개인키를 복구해 낼 수 있다.

IV. 대응 기법

본 논문에서 제안하는 대응 기법은 세 개의 난수로 초기화된 테이블을 사용하여 연산한다. 기존 구현이 개인키 $f(=pF+1)$를 복구한 뒤에 $f \cdot e$ 연산하는 방식이 기존이었다면, 제안하고 있는 대응 기법은 F를 이용하여 $p \times F \cdot e$ 계산 후 마지막에 e를 더하여 메시지를 복구한다. 이를 수식으로 표현하면 다음과 같다.

$$m = f \cdot e = (pF+1) \cdot e = pe \cdot F + e \qquad (7)$$

$p \times e$를 먼저 계산하여 저장한 뒤, 삼진 다항식 F의 값 -1, 0, 1에 따라 연산 결과를 세 개의 테이블에 저장한다. 그러고 나서 1의 테이블에서 -1의 결과를 빼는 과정으로 연산을 마무리 할 수 있다. 값을 더하며 누적하는 과정에서 덧셈을 시작할 파라미터 값을 $(i+j) \bmod N$을 이용하여 $\bmod(x^N-1)$까지 동시에 처리함으로 연산을 더 효율적으로 할 수 있다. 또한 F_i 값을 불러올 때 -1과 0과 1이 로드됨에 따라 발생할 수 있는 부채널적 취약점을 보완하기 위해 복호화 과정에 사용하는 F 값을 인코딩하여 보관한다. 기기의 물리적 특성을 분석하여 -1, 0, 1을 인코딩하기 위한 값을 선택한다. 본 예시에서는 다음과 같은 규칙으로 개인키를 인코딩 했다.

$$enc(-1)=1,\ enc(0)=2,\ enc(1)=4$$

즉 제안하는 알고리즘에서 삼진 다항식은 $\{1,2,4\}$로 구성된다. 이를 알고리즘으로 표현하면 Algorithm 2와 같다. 마지막 13단계에서는 다항식 계수 중 1의 연산 결과와 -1의 연산 결과를 빼주는 과정과 암호문 e를 더하는 연산과 $\bmod q$를 동시에 처리할 수 있다. $\bmod q$ 연산의 경우 표준에서 q 값을 2의 지수승으로 정하여, $(q-1)$ 값을 AND 연산을 통해 계산해줄 수 있다.

Algorithm 2에서는 F에 따라 서로 다른 테이블을 호출하고 있지만 동일한 연산을 하고 있으므로 -1, 0, 1 값에 따른 소비 전력의 차이가 나타나지 않을 것이라고 생각할 수 있다. 또한 세 개의 테이블 값을 단계 3~4와 같이 동일한 난수로 초기화했을 때 [10]의 SPA와 CPA에도 안전한 설계가 될 수 있다. 0과 0이 아닌 값이 더해지는 연산$(0+nz)$와 0이 아닌 두 값의 연산$(nz+nz)$에 해당하는 파형의 차이를 통한 SPA를 막을 수 있고, 더해지는 데이터 값들이 무작위화되면서 중간값을 추측할 수 없기 때문에 CPA에 안전하다. 동일한 값으로 초기화를 해주게 되면, 13단계에서 다른 연산들과 함께 난수값이 사라지므로 따로 연산을 처리할 필요가 없다.

이를 증명하기 위해 분석에서 사용했던 보드와 환경을 동일하게 하고, Algorithm 2를 적용하여 파

Algorithm 2. Countermeasure of NTRUEncrypt

Input: ciphertext polynomial $e \in R$, and F', an encoding of $F \in L_f$
Output: $m = f \cdot e \pmod q$

1. for $0 \leq i < N$ do
2. $PE_i \leftarrow p \times e_i$
3. $T_i[1] \leftarrow r$ ▷ r is a random
4. $T_i[2] \leftarrow r$
5. $T_i[4] \leftarrow r$
6. end for
7. for $0 \leq i < N$ do
8. for $0 \leq j < N$ do
9. $T_{i+j \bmod N}[F_i'] \leftarrow T_{i+j \bmod N}[F_i'] + PE_j$
10. end for
11. end for
12. for $0 \leq i < N$
13. $m_i \leftarrow (((T_i[4] - T_i[1]) + e_i) \bmod q$
14. end for
15. return m

형을 수집하였다. 그 결과, Fig. 5는 전체파형으로 로우픽이 없음을 확인할 수 있다. 또한, Fig. 6 과 Fig. 7은 전체 파형을 확대한 파형으로 모든 경우에 동일한 파형이 나타남을 확인할 수 있다.

위와 같이 적용한 대응 기법의 경우 기존 연산과 초기화, 덧셈, 곱셈의 횟수를 N차일 때 비교하면 Table 1과 같다. 대응 기법의 뺄셈의 경우 덧셈으로 포함하였으며 더할 배열의 위치를 계산하는 연산 $(i+j \bmod N)$는 세지 않았다.

또한 알고리즘의 입력과 출력을 제외하고 연산 과정 중에 사용되는 메모리의 사용량을 비교하면 Table 2와 같다. NTRUEncrypt 코드는 기본적으로 16비트를 워드 단위로 사용하고 있으므로, Table 2의 값은 워드 개수에 2 Byte를 곱한 결과

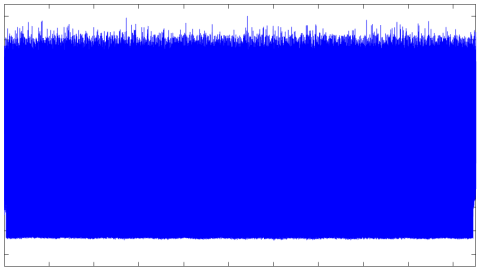


Fig. 5. Full Trace of Countermeasure

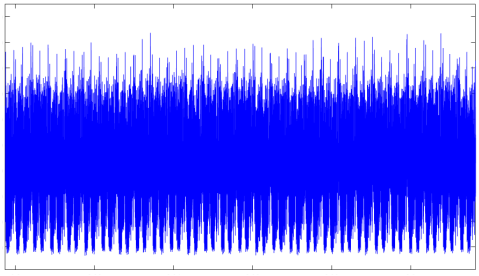


Fig. 6. Enlarged Trace of Countermeasure

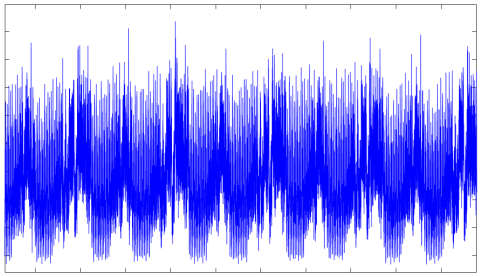


Fig. 7. More Enlarged Trace of Countermeasure

Table 1. Comparison of operation

	Unprotected	Protected
Initial	N	$3N$
Add/Sub	N^2	N^2+2N
Mul	N^2+N	N

를 의미한다. Algorithm 2에서 개인키를 1, 2, 4로 인코딩하기 위하여 5개의 N개짜리 배열을 만들게 된다. 또한 상수 p와 암호문 e를 연산하여 저장할 배열이 N개 사용되므로 Algroithm 2에서는 $6N$개의 16비트 배열을 사용한다. 반면 Algorithm 1은 개인키를 연산하여 저장할 배열 N개와 연산의 중간 값을 저장할 배열 $2N$개로 연산이 가능하다.

초기화와 덧셈, 뺄셈, 그리고 곱셈의 연산량을 비교했을 때 총 횟수가 줄었으며, 가장 무거운 연산인 곱셈의 횟수가 크게 줄어들었다. 메모리를 비교하면 제안하는 대응 기법에서 배열을 3배 사용하고 있지만, 감소한 연산의 횟수를 감안하였을 때 제안하는 알고리즘이 더 효율적이라고 말할 수 있다. 결론적으로 Algorithm 2의 방법은 Algorithm 1보다 효율적이면서 부채널적으로 안전한 알고리즘이라고 할 수 있다.

Table 2. Comparison of memory

	Unprotected	Protected
RAM (Byte)	$6N$	$12N$

V. 결 론

보안 인증이 필수적인 스마트 기기에 암호 모듈을 구현할 경우, 표준화된 코드를 이용해야 한다. 스마트 기기는 대부분 부채널 공격자에게 쉽게 노출된 환경이다. 그러므로 NIST에서 현재 진행 중인 표준화 코드들이 부채널 분석에 안전해야 함은 자명하다. 본 논문에서는 NIST의 포스트 양자 암호 표준화 공모전에 제출된 후보 중 하나인 NTRU를 분석했다. NTRU의 경우 다른 암호들과는 다르게 개인키 정보로 -1, 0, 1만을 사용한다. 위의 세 가지 값은 소비 전력 특징이 크게 나타나기 때문에 부채널 분석에 대응해야 할 필요가 있다. 이 값들을 가리며 연산할 수 있는 방법은 본 논문에서 제안한 것 이외에도 다양한 방법이 있을 수 있다. 뿐만 아니라 공개키 암호

는 효율적인 연산을 위한 연구가 계속될 것이다. 이후에도 새로운 방식을 적용할 때 부채널 분석까지도 고려해야만 더 안전한 암호를 구현할 수 있겠다.

References

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

[2] Hoffstein, J., Pipher, J., & Silverman, J. H, "NTRU: A ring-based public key cryptosystem," In International Algorithmic Number Theory Symposium, Springer, Berlin, Heidelberg, pp. 267-288, June. 1998.

[3] Peikert, C, "A decade of lattice cryptography," Foundations and Trends® in Theoretical Computer Science, vol. 10, no, 4, pp. 283-424, Mar. 2016.

[4] Hoffstein, J and Silverman, J, " Optimizations for NTRU," Public-Key Cryptography and Computational Number Theory, de Gruyter, Warsaw, pp. 77-88, 2001.

[5] NIST, "NTRUEncrypt" http://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions, 2017.

[6] Kocher, P. C, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems In Annual," International Cryptology Conference, Springer, Berlin, Heidelberg, pp. 104-113, Aug. 1996.

[7] Kocher, P, Jaffe, J, and Jun, B, "Differential power analysis," In Annual International Cryptology Conference, Springer, Berlin, Heidelberg, pp. 388-397, Aug. 1999.

[8] Brier, E, Clavier, C, and Olivier, F, "Correlation power analysis with a leakage model," In International workshop on cryptographic hardware and embedded systems, Springer, Berlin, Heidelberg, pp. 16-29, Aug. 2004.

[9] Messerges, T. S, "Securing the AES finalists against power analysis attacks," In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, pp. 150-164, Apr. 2000.

[10] Song, J. E, Han, D. G, Lee, M. K, and Choi, D. H, "Power analysis attacks against NTRU and their countermeasures," Journal of the Korea Institute of Information Security and Cryptology, 19(2), pp. 11-21, Apr. 2009.

[11] Silverman, J. H., & Whyte, W, "Timing attacks on NTRUEncrypt via variation in the number of hash calls," In Cryptographers' Track at the RSA Conference, Springer, Berlin, Heidelberg, pp. 208-224, Feb. 2007.

[12] Lee, M. K, Song, J. E, Choi, D, and Han, D. G, "Countermeasures against power analysis attacks for the NTRU public key cryptosystem," IEICE transactions on fundamentals of electronics, communications and computer sciences, vol. 93, no. 1, pp. 153-163, 2010.

[13] Zheng, X, Wang, A, and Wei, W, "First-order collision attack on protected NTRU cryptosystem," Microprocessors and Microsystems, vol. 37, no. 6-7, pp. 601-609, 2013.

[14] DooHo Choi, YongJe Choi, JeaCheol Ryou, "Implementing Side Channel Analysis Evaluation Boards of KLA-SCARF system," Journal of the Korea Institute of Information Security & Cryptology, 24(1), pp. 229-240, Feb. 2014.

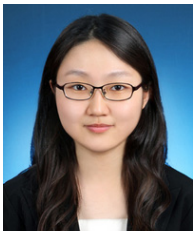
[15] Csrc.nist.gov, Post-Quantum Cryptogr

aphy, https://csrc.nist.gov/Projects/Post-Quantum-Cryptography, 2016.

[16] ATmel, ATmega128(L) Datasheet, http://www.atmel.com, 2006.

〈저 자 소 개〉

안 수 정 (Soojung An) 학생회원
2017년 2월: 고려대학교 환경생태공학과 학사
2017년 3월~현재: 고려대학교 정보보호학과 석사과정
〈관심분야〉 부채널 공격, 부채널 대응기법, 후양자암호



김 수 리 (Suhri Kim) 학생회원
2014년 2월: 고려대학교 수학과 학사
2016년 8월: 고려대학교 정보보호학과 석사
2016년 9월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 후양자암호

진 성 현 (Sunghyun Jin) 학생회원
2015년 2월: 서울시립대학교 수학과 학사
2017년 2월: 고려대학교 정보보호학과 석사
2017년 3월~현재: 고려대학교 정보보호학과 박사과정
〈관심분야〉 부채널 공격

김 한 빛 (Hanbit Kim) 학생회원
2014년 2월: 고려대학교 신소재공학과 학사
2016년 2월: 고려대학교 정보보호학과 석사
2016년 3월~현재: 고려대학교 정보보호학과 박사과정
〈관심분야〉 부채널 공격, 부채널 대응기법, 암호시스템 안전성 분석 및 고속구현

김 희 석 (HeeSeok Kim) 정회원
2006년: 연세대학교 수학과 학사
2008년: 고려대학교 정보보호대학원 석사
2011년: 고려대학교 정보보호대학원 박사
2011년 9월~2012년 12월: Bristor University 박사후 연구원
2013년~2016년 8월: 한국과학기술정보연구원(KISTI) 선임연구원
2015년~2016년 8월: 과학기술연합대학원대학교(UST) 조교수
2016년 9월~현재: 고려대학교 과학기술대학 사이버보안전공 조교수
〈관심분야〉 부채널 공격, 암호시스템 안전성 분석 및 고속구현, 암호칩 설계 기술, 보안관제, 네트워크 보안

홍 석 희 (Seokhie Hong) 종신회원
1995년: 고려대학교 수학과 학사
1997년: 고려대학교 수학과 석사
2001년: 고려대학교 수학과 박사
1999년 8월~2004년 2월: ㈜시큐리티 테크놀로지 선임연구원
2003년 3월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U. Leuven ESAT/SCD-COSIC 박사후 연구원
2005년 3월~2013년 8월: 고려대학교 정보보호대학원 부교수
2013년 9월~현재: 고려대학교 정보보호대학원 정교수
〈관심분야〉 대칭키 및 공개키 암호 알고리즘, 부채널 공격 및 대응기법, 디지털 포렌식