情報保護學會論文誌
第19卷 第2號, 2009. 4

NTRU 암호에 대한 전력 분석 공격 및 대응 방법*

송 정 은,[1†] 한 동 국,[2] 이 문 규,[1‡] 최 두 호[3]
[1]인하대학교 컴퓨터정보공학부, [2]국민대학교 수학과, [3]한국전자통신연구원

Power analysis attacks against NTRU and their countermeasures*

Jeong Eun Song,[1†] Dong-Guk Han,[2] Mun-Kyu Lee,[1‡] Dooho Choi[3]
[1]School of Computer and Information Engineering, Inha University,
[2]Department of Mathematics, Kookmin University,
[3]Electronics and Telecommunications Research Institute

요 약

NTRU는 1990년대 Hoffstein 등에 의해 제안된 격자(Lattice) 기반 공개키 암호체계로서 기존의 공개키 암호와 비교하여 동일한 안전성을 제공하면서 암호화 및 복호화 속도가 빠르며 양자 연산 알고리즘을 이용한 공격에도 강하다는 이점이 있어 많은 주목을 받고 있다. 본 논문에서는 단순 전력 분석 공격과 통계적 특성을 이용한 전력 분석 공격인 상관계수 전력 분석 공격에 대한 NTRU의 안전성을 분석하고, NesC로 구현한 NTRU의 연산을 Telos 모트(mote)에서 수행시켜 측정한 전력 소모 데이터에 상관계수 전력 분석 공격을 적용하여 개인키 정보를 복원하는 실험 결과를 보인다. 또한 이러한 전력 분석 공격을 방지하기 위한 대응 방법을 제시한다. 먼저, 단순 전력 분석 공격을 방지하기 위해 연산 결과를 저장할 배열을 0이 아닌 수로 초기화 시키는 방법을 제안하고, 통계적 특성을 이용한 전력 분석 공격을 방지하기 위해 연산 순서를 변경하거나 컨볼루션(convolution) 연산에 사용되는 피연산자들에게 무작위성(randomness)을 부여하여 같은 입력에 대해서 랜덤한 전력 소모를 보이도록 하는 방법을 제안한다.

ABSTRACT

The NTRU cryptosystem proposed by Hoffstein et al. in 1990s is a public key cryptosystem based on hard lattice problems. NTRU has many advantages compared to other public key cryptosystems such as RSA and elliptic curve cryptosystems. For example, it guarantees high speed encryption and decryption with the same level of security, and there is no known quantum computing algorithm for speeding up attacks against NTRU. In this paper, we analyze the security of NTRU against the simple power analysis (SPA) attack and the statistical power analysis (STPA) attack such as the correlation power analysis (CPA) attack. First, we implement NTRU operations using NesC on a Telos mote, and we show how to apply CPA to recover a private key from collected power traces. We also suggest countermeasures against these attacks. In order to prevent SPA, we propose to use a nonzero value to initialize the array which will store the result of a convolution operation. On the other hand, in order to prevent STPA, we propose two techniques to randomize power traces related to the same input. The first one is random ordering of the computation sequences in a convolution operation and the other is data randomization in convolution operation.

Keywords: NTRU, Side Channel Attack, Power Analysis Attack, Countermeasure

접수일(2008년 7월 7일), 수정일(2008년 11월 17일), 게재확정일(2008년 12월 18일)
* 본 연구는 지식경제부 및 정보통신연구진흥원의 IT핵심기술개발사업의 일환으로 수행하였음. (2008-F-045-02, 장애인 및 고령자를 위한 Digital Guardian 기술개발)
† 주저자, aboutucs@gmail.com
‡ 교신저자, mklee@inha.ac.kr

I. 서 론

NTRU는 1990년대 Jeffrey Hoffstein 등이 제안한 격자(Lattice) 문제를 기반으로 하는 공개키 암호 체계로 기본 연산은 다항식 환(polynomial rings) 상에서 이루어진다[1]. 현재 IEEE에서 P1363.1[2]로 격자 문제를 기반으로 하는 공개키 암호 표준으로 고려되고 있는 NTRU는 기존 공개키 암호 RSA, ECC(Elliptic Curve Cryptography) 등에 비교하여 동일한 안전성을 제공하면서 암호화 및 복호화 속도가 빠르다는 이점을 갖는다. 또한 RSA나 DSA, Diffie-Hellman과 같은 이산 로그 기반의 암호 체계나 ECDSA, ECMQV와 같은 타원 곡선 이산 로그 시스템과 같은 경우 양자 연산 알고리즘(quantum computation algorithm)을 이용하여 공격 속도를 향상시킬 수 있는 반면 NTRU에 대해서는 공격 속도를 향상시킬 수 있는 알려진 방법이 존재하지 않는다[3].

NTRU에 대한 지금까지의 연구는 다음과 같다. Hoffstein과 Silverman은 특별한 형태의 다항식을 이용하여 NTRU에서 사용하는 계산 양을 줄일 수 있음을 보였고[4], Bailey등은 한정된 자원을 갖는 장치에서 효율적으로 NTRU를 구현하는 방법을 제시하였다[5]. Gaubats, Kaps와 Sunar는 3000개 이하의 게이트(gate)를 사용해 NTRU를 하드웨어로 구현함으로써 센서 노드와 같은 초소형 장치에서도 공개키 암호를 사용할 수 있음을 보였다[6]. 이처럼 NTRU에 대한 소프트웨어 구현이나 하드웨어 구현을 중심으로 한 연구는 활발히 이루어지고 있으나 구현 후 부채널 공격에 대한 안전성을 분석하는 연구는 Atici 등에 의해 RFID Security 2008에 소개된 연구가 유일하다[7]. 그러므로 본 논문에서는 소프트웨어로 NTRU를 구현한 후 복호화 연산 시 부채널 공격을 통해 개인키를 복원하는 방법에 대해 기술하고 부채널 공격에 대한 대응 방법을 몇 가지 제안한다. NTRU에 대한 단순 전력 분석 공격은 시스템 내부에서 덧셈 연산 $0+x$와 $x+y$ $(x, y \neq 0)$를 수행 할 때 서로 다른 전력 소모 패턴을 보인다는 점을 이용하는 것으로 연산 결과를 저장할 배열을 0이 아닌 수로 초기화 시키는 방법으로 공격을 방지할 수 있다. 차분 전력 분석 공격이나 상관계수 전력 분석 공격은 덧셈 연산 시 피연산자들의 값에 따라 미묘하게 변화하는 전력 소모 패턴을 통계적인 분석을 통해 공격하는 것으로, 연산 순서를 무작위로 변경하거나 컨볼루션(convolution) 연산에 사용되는 피연산자들에게 무작위성(randomness)을 부여하여 연산 시 같은 입력에 대해서도 전력 소모는 랜덤하게 나타나도록 함으로써 공격을 방지할 수 있다. 실제로 Telos 모듈에서 구현된 NTRU에 대한 상관계수 전력 분석 공격 실험을 통해 개인키를 복원할 수 있음을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 NTRU 복호화 연산에 사용되는 컨볼루션을 이해하기 위해 필요한 배경지식인 다항식 환(polynomial ring)과 컨볼루션에 대해 설명하고, NTRU 암호 체계와 부채널 공격(side channel attack)에 대해 간략하게 설명한다. 3장에서는 NTRU에 대한 단순 전력 분석 공격 방법과 대응 방안에 대해 기술하며, 4장에서는 NTRU에 대한 상관계수 전력 분석 공격 방법을 제시하고, 실제 Telos 모듈상에서의 공격 실험 방법과 결과를 제시한다. 5장에서는 NTRU에 대한 상관계수 전력 분석 공격을 예방하기 위한 대응 방법을 제안하고, 마지막으로 6장에서는 결론을 맺고 향후 연구 방향에 대해 논의하도록 한다.

II. 배경지식

2.1 다항식 환(polynomial ring)과 컨볼루션(convolution)

여정수들의 집합을 Z라 할 때, 정수 계수를 갖는 모든 다항식들의 집합을 다항식 환이라 하고 $Z[X]$로 표시한다. $R = Z[X]/(X^N-1)$로 정의 되는 상환(quotient ring) R은 $Z[X]$에 속하는 다항식을 다항식 X^N-1로 나누었을 때 생길 수 있는 모든 가능한 나머지 다항식들의 집합을 의미한다. R의 원소 a는 다음 식 (1)과 같이 다항식으로 표현할 수 있다.

$$a(X) = \sum_{i=0}^{N-1} a_i X^i \tag{1}$$

a, b가 R의 원소일 때, 다항식 a와 b의 컨볼루션 곱 $c(c(X) = a(X) * b(X))$는 다음 식 (2)와 같이 표현할 수 있다.

$$\begin{aligned} c_k &= \sum_{i=0}^{k} a_i b_{k-i} + \sum_{i=k+1}^{N-1} a_i b_{N+k-i} \\ &= \sum_{i+j \equiv k \,(\mathrm{mod}\, N)} a_i b_j \\ &(\because X^N \equiv 1 \bmod (X^N-1)) \end{aligned} \tag{2}$$

각각의 다항식 a와 b가 임의의 정수 계수를 갖는 경우, 위의 식 (2)의 연산 시 N^2개의 정수 곱셈이 필요하다.

NTRU의 기본 연산들은 다항식 환 상에서 이루어지며 이 때 사용되는 다항식의 계수를 결정하는 방식은 여러 가지가 있는데, 본 논문에서는 두 다항식 중 하나의 다항식 계수가 0 또는 1인 이진 다항식인 경우에 대해서만 고려하는 IEEE P1363.1[2]의 방식을 따른다. [그림 1]은 NTRU에서 사용되는 기본 다항식 컨볼루션 알고리즘을 나타낸다. 알고리즘의 입력에서 $c(X)$는 '공개키' 혹은 '암호문'에 해당하는 정수 계수를 갖는 $N-1$차 일반 다항식을 의미하고, $a(X)$는 '개인키' 혹은 '임의로 선택된 다항식'에 해당하는 0 또는 1의 계수를 갖는 $N-1$차 이진 다항식을 의미하며, $a(X)$의 1의 위치는 배열 b에 저장되어 있다. 알고리즘의 출력에서 배열 t는 $a(X)*c(X)$의 연산 결과를 저장할 배열이다. 제1행, 5행, 9행의 for 문에서 범위를 제한하는 N과 제4행의 for문에서 범위를 제한하는 d, 제10행에서 모듈러 연산의 피연산자로 사용되는 q는 모두 공개된 매개 변수이다.

2.2 NTRU 공개키 암호 체계

NTRU는 공개키 암호로서 연산은 환을 기반으로 이루어진다. NTRU 암호체계는 다양한 버전이 존재하는데, IEEE P1363.1[2]에서 채택하고 있는 Hoffstein, Bailey등에 의해 제안된 개량 버전을 중심으로 설명하도록 한다.

- NTRU는 (N, d, p, q) 네가지 공개 매개 변수를 갖는다. $(\gcd(p, q)=1,\ p \ll q)$
- 다항식의 계수들은 $\bmod p$나 $\bmod q$에 의해 감산되는데, 이는 p나 q로 나눈 나머지로 계산됨을 의미한다.
- 다항식 f의 $\bmod q$ 상의 역원은 $f^{-1} \bmod q$로 표기하고 $f*f^{-1} \equiv 1 \bmod q$를 만족시킨다.

NTRU의 공개 매개 변수는 다양하게 존재하는데 공개 매개 변수의 선택에 따라 보안 수준을 결정할 수 있다. 본 논문의 후반부에 제시될 실험에서는 IEEE P1363.1[2] 표준의 초안에서 제안하는 매개 변수 집합을 사용하였다.

2.2.1 키 생성

N개의 이진 계수 중 d개가 1이고 나머지 $N-d$개는

입력: b는 이진 다항식 $a(X)$의 계수들 중 1의 위치를 나타내는 배열. $c(X)$는 일반 다항식; 다항식 $a(X)$, $c(X)$는 N개의 계수를 갖는다.
출력: $a(X)*c(X)$ 연산 결과를 저장한 배열 t

```
1: for 0 ≤ j < 2N do
2:   t_j ← 0
3: end for
4: for 0 ≤ j < d do
5:   for 0 ≤ k < N do
6:     t_{k+b[j]} ← t_{k+b[j]} + c_k
7:   end for
8: end for
9: for 0 ≤ j < N do
10:  t_j ← (t_j + t_{j+N}) mod q
11: end for
```

[그림 1] NTRU의 기본 다항식 컨볼루션 알고리즘

0인 임의의 다항식 F, g $(F, g \in R)$를 이용하여 개인키와 공개키를 생성한다. 개인키 f는 $f = 1 + pF \bmod q$로 계산되는 다항식이고 공개키 h는 $h = pf^{-1} * g \bmod q$로 계산되는 다항식이다.

2.2.2 암호화

메시지 m $(m \in R)$과 임의로 선택한 작은 계수를 갖는 다항식 r $(r \in R)$을 이용하여 암호문 e를 계산한다. 암호문 e는 $e = r*h + m \bmod q$로 계산된다.

2.2.3 복호화 및 유효성

암호문 e를 복호화 하는 중간 과정으로 $a = e*f \bmod q$를 계산한다. 이 때, 다항식 a의 계수들의 범위가 $A \le a_i < A+q$를 만족하도록 선택한다. A는 나머지 매개 변수에 따라 간단한 공식으로 구할 수 있는 고정된 값이다. 계산된 다항식 a를 $\bmod p$ 연산하면 평문 메시지 m을 복원할 수 있다$(m = a \bmod p)$.

복호화를 위해 계산된 다항식 a는 다음 식 (3)과 같은 식을 만족한다.

$$\begin{aligned} a &\equiv e*f \bmod q \\ &\equiv (r*h+m)*f \bmod q \ (\because e \equiv r*h+m) \\ &\equiv pr*g + m*f \bmod q \\ &\quad (\because h*f \equiv pg*f^{-1}*f \equiv pg) \end{aligned} \tag{3}$$

최종 다항식 $pr*g + m*f \bmod q$에서 매개 변수를 적당히 선택하면 계수들을 q보다 작은 길이의 범위 내에 놓이도록 조정할 수 있다. 따라서 다음 식 (4)와 같이

a에 대한 등식이 성립하게 된다.

$$a = pr*g + m*f = pr*g + m*(1+pF) \quad (4)$$

즉, mod q에 대해서가 아니라 정확하게 등식이 성립하므로 $m = a \bmod p$가 되어 메시지를 복원할 수 있다.

2.3 부채널 공격(Side Channel Attack)

부채널 공격은 알고리즘 상의 기술적 취약성을 이용해 공격하는 것이 아니라 암호 시스템을 물리적으로 구현한 장치로부터 부수적으로 얻어지는 정보를 이용하여 비밀 정보를 알아내는 공격이다. 대표적인 부채널 공격으로는 시간 공격(timing attack)[8]과 전력 분석 공격(power analysis attack)[9]이 있다. 시간 공격은 암호 장치에서 연산을 수행하는데 걸린 시간을 측정하여 측정한 시간을 기반으로 비밀 정보를 알아내는 공격 방법이다. 전력 분석 공격은 1998년 Kocher등에 의해 소개된 방법으로 암호 장치에서 연산을 수행할 때 소모되는 전력을 측정하여 관찰함으로 비밀 정보를 알아내는 공격 방법이다[9]. 전력 분석 공격 방법으로는 한번의 알고리즘 수행 시 소모하는 전력 소비를 측정하여 연산에 따라 다르게 나타나는 전력 소모 패턴을 관찰함으로 비밀 정보를 얻어내는 단순 전력 분석(simple power analysis, SPA) 공격과 다양한 입력 데이터를 이용하여 알고리즘을 수행시켜 얻은 전력 소비 데이터와 비밀 정보와의 상관관계를 통계적인 분석을 통해 비밀 정보를 얻어내는 차분 전력 분석(differential power analysis, DPA) 공격, 상관계수 전력 분석(correlation power analysis, CPA)[10] 공격이 있다. 차분 전력 분석 공격이나 상관계수 전력 분석 공격을 위해서는 다양한 입력에 대한 알고리즘 수행 시 소비되는 전력을 표본화(sampling)하여 수집하는 데이터 수집 단계와 수집한 데이터를 통계적인 방법을 이용하여 분석하는 데이터 분석 단계를 수행하게 된다. 차분 전력 분석 공격에서는 상관관계를 분석하기 위해 수집한 전력 데이터를 분류하여 각각 평균을 구한 후 차분 전력을 구하여 비밀 정보를 알아내고, 상관계수 전력 분석 공격에서는 상관관계를 분석하기 위해 수집한 전력 데이터와 알고리즘 내 연산 중간 결과 값의 상관계수를 구하여 비밀 정보를 알아낸다. 일반적으로 거론되는 차분 전력 분석은 1차 차분 전력 분석을 의미하며, 알고리즘 수행 시 서로 다른 n개의 연산의 중간 값들에 상응하는 전력 소비 데이터를 이용하여 차분 전력 분석 공격을 할 수 있는데 이를 n차 차분 전력 분석(nth order DPA) 공격이라 한다[11]. n차 차분 전력 분석 공격을 이용할 경우 1차 보다 강력한 공격이 가능하나 처리해야 할 데이터 양이 급격하게 증가한다. 2차 차분 전력 분석 공격의 경우 알고리즘 상에서 서로 다른 두 개의 연산의 중간 값들에 대해 전력 소비를 측정하고 연산에 대응되는 정확한 전력 소비를 알 수 없으므로 길이가 ℓ인 전력 데이터의 각각의 값의 차의 절대값을 취해 얻은 길이 $\ell(\ell-1)/2$의 전력 데이터를 이용해 1차 차분 전력 분석 방법으로 공격한다[12]. 상관계수 전력 분석 또한 n차로 확장이 가능하다.

전력 분석을 이용해 공격할 경우 전력을 분석하는 모델은 크게 해밍 웨이트(hamming weight) 모델과 해밍 디스턴스(hamming distance) 모델로 나눌 수 있다. 해밍 웨이트 모델은 관찰 대상 레지스터가 저장하고 있는 데이터의 해밍 웨이트에 따라 전력을 분석하고, 해밍 디스턴스 모델은 관찰 대상 레지스터의 이전 상태와 이후 상태의 데이터의 해밍 디스턴스, 즉 두 상태의 데이터의 XOR 값에 대한 해밍 웨이트에 따라 전력을 분석한다.

III. NTRU에 대한 단순 전력 분석 공격 및 대응 방법

NTRU에 대한 부채널 공격은 암호문의 복호화 시 수행되는 연산 $e*f \bmod q$에서 이루어진다. 실제 소프트웨어로 구현된 NTRU의 복호화 연산에서는 두 개의 일반 다항식의 곱셈인 $e*f$ 형태의 연산을 사용하지 않는다. 그 이유는 일반 다항식과 일반 다항식 곱셈의 경우 연산 시간이 오래 걸리기 때문이다. 따라서 일반적으로는 암호문을 복호화하기 위해 계산된 형태의 개인키 f를 저장하고 있는 것이 아니라 효율적인 컨볼루션 연산을 위해 $f = 1+pF$이므로 F를 저장하여 복호화를 수행하게 되어 실제로 복호화를 위해 수행되는 연산은 $e*f \bmod q = e + pe*F \bmod q$와 같다. 그러므로 본 논문에서는 복호화 시 사용되는 개인키가 F 형태로 저장되어 있다고 가정하고 $e*F$ 컨볼루션 연산이 수행되는 시점에서 부채널 공격을 실시한다. 복호화 시 수행되는 컨볼루션 연산은 [그림 1]과 같은 알고리즘에 의해 이루어진다. 복호화 시 입력이 되는 배열 b는 이진 다항식으로 이루어진 개인키 생성을 위한 다항식 F의 1의 위치를 저장한다. 공격자는 b의 값을 알

아냄으로 개인키 정보를 복원해낼 수 있다. 또 다른 입력인 $c(X)$는 암호문을 저장하고 있는 배열로 공격자가 임의로 값을 설정할 수 있다.

이 장에서는 기본적인 컨볼루션 알고리즘을 이용해 구현한 NTRU가 단순 전력 분석 공격에 취약함을 보이고, 이를 방지할 수 있는 대응 방법을 제시한다.

3.1 공격 방법

NTRU의 컨볼루션은 [그림 1]의 제6행과 같이 반복적인 덧셈 연산으로 이루어진다. NTRU에 대한 단순 전력 분석 공격은 덧셈 연산 시 시스템 내부에서 $0+x$와 $x+y$ $(x, y \neq 0)$ 각각의 경우에 대해 다른 전력 소모 패턴을 보인다는 점을 이용하는 것이다. 따라서 공격자는 복호화 시 수행되는 컨볼루션 연산의 입력인 일반 다항식 $c(X)$를 0이 아닌 값으로 구성하여 컨볼루션 연산을 수행시켜 개인키 정보를 알아낼 수 있다.

[그림 1]의 제1-3행에서 컨볼루션 결과를 저장하는 배열 t를 0으로 초기화하므로 제4행에서 $j=0$일 경우 제5-7행의 N개 연산은 $0+c_k(c_k \neq 0)$가 된다. 제4행에서 $j=1$이 되면 $b[1]-b[0]=b1$라 할 경우 제5-7행에서 k의 범위가 $0 \leq k < N-b1$ 일 경우에는 $c_{k+b1}+c_k(c_{k+b1}, c_k \neq 0)$ 연산을 하고, k의 범위가 $N-b1 \leq k < N$ 일 경우에는 $0+c_k(c_k \neq 0)$ 연산을 한다. 따라서 공격자는 연산에 따른 전력 소모 패턴의 차이를 이용하여 $0+c_k(c_k \neq 0)$가 어느 부분에서 발생하는지 분석하여 $b1$의 값을 구할 수 있다. 같은 방법으로 제4행의 $j=J(1 \leq J < d)$ 일 경우 $b[J]-b[J-1]$ 값을 구할 수 있다. 즉, 배열 b의 인접한 원소들 간의 상대적인 차이 값을 모두 구할 수 있으며, 전수 조사를 통해 $b[0]$의 값을 찾아내면 배열 b의 모든 값을 알 수 있으므로 개인키를 복원할 수 있다.

예를 들어, $N=8, b=[0, 3, 4, 6, 7]$인 경우에 대해 생각해 보자. 0이 아닌 x, y에 대하여 $0+x$ 연산에 대한 전력 소모 패턴을 ZN, $x+y$ 연산에 대한 전력 소모 패턴을 NN이라 할 때, 시간에 따른 컨볼루션 연산의 전체적인 전력 소모 패턴을 다음과 같이 나타낼 수 있다.

	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$
$j=0$:	ZN	ZN	ZN	ZN	ZN	ZN	ZN	ZN
$j=1$:	NN	NN	NN	NN	NN	ZN	ZN	ZN
$j=2$:	NN	NN	NN	NN	NN	NN	NN	ZN
$j=3$:	NN	NN	NN	NN	NN	NN	ZN	ZN
$j=4$:	NN	NN	NN	NN	NN	NN	NN	ZN

위의 전력 소모 패턴을 분석해 보면, j가 $1 \sim 4$ 값을 갖는 동안에는 전력 소모 패턴의 후반부가 ZN으로 나타남을 볼 수 있는데, 이는 배열 b의 인접한 원소들 간의 상대적인 차이를 의미한다. 즉, $j=1$일 경우 ZN 형태의 전력 소모 패턴은 마지막 부분에서 세 번 관찰되는데 이는 $b[0]$와 $b[1]$의 상대적인 차이가 3임을 나타내는 것이고, $j=2$일 경우 ZN 형태의 전력 소모 패턴은 한 번 관찰 되는데 이는 $b[1]$과 $b[2]$의 상대적인 차이가 1임을 나타낸다. 마찬가지로 $j=3, j=4$인 각각의 경우에 대해 ZN 형태의 전력 소모 패턴이 관찰되는 수를 조사하면 $b[2]$와 $b[3]$의 상대적인 차이는 2, $b[3]$과 $b[4]$의 상대적인 차이는 1임을 알 수 있다. 결과적으로 공격자는 배열 b의 원소간의 상대적인 위치를 알고 있으므로 배열 b의 값을 추측할 수 있다. 즉, 처음 $b[0]$의 위치는 알 수 없으므로 α라고 가정하면 $b=[\alpha, \alpha+3, \alpha+4, \alpha+6, \alpha+7]$를 얻을 수 있다. 공격자는 전수 조사를 통해 α를 구할 수 있는데, 배열 b의 특성을 고려하면 가능한 α의 범위를 크게 줄일 수 있다. 배열 b는 이진 다항식으로 이루어진 개인키 생성을 위한 $N-1$차 다항식 F의 1의 위치를 저장하고 있으므로 배열 b의 원소는 $N-1$보다 작거나 같은 값을 갖는다. 그러므로 배열 b의 원소 중 가장 큰 값을 갖는 $\alpha+7$의 값은 $\alpha+7 \leq 7$의 범위를 가지므로 $\alpha \leq 0$이 된다. 따라서 위 예제의 경우 $\alpha=0$임을 바로 알 수 있으므로, 공격자는 $b=[0, 3, 4, 6, 7]$를 알아낼 수 있다.

3.2 대응 방법

단순 전력 분석 공격은 $0+x$와 $x+y$ $(x, y \neq 0)$ 각각의 연산 수행 시 소모되는 전력 패턴이 다르기 때문에 가능하므로 [그림 1]의 제5-7행의 연산에서 $0+x$ $(x \neq 0)$인 경우가 발생하지 않도록 함으로써 공격을 막을 수 있다. 추가적인 연산 없이 $0+x$ $(x \neq 0)$ 연산이 발생하는 것을 방지하기 위한 한 방법을 알고리즘으로 기술하면 [그림 2]와 같다. [그림 2]의 제1-3행과 같이 연산 결과를 저장하는 배열 t를 0으로 초기화하지 않고 q로 초기화 하여 [그림 2]의 제5-7행의 연산에서 항상 $x+y$ $(x, y \neq 0)$ 연산이 되도록 만들어 준다. 이때, 배열 t의 초기값을 [그림 2]의 제16행의 mod 연산의 피연산자인 q와 동일한 값으로 설정함으로써, 다항식 컨볼루션 연산의 최종적인 결과에 영향을 주지 않는다.

그러나 [그림 2]의 알고리즘은 경우에 따라서 공격이 가능할 수도 있는데, 예를 들어 공격자가 입력 다항

식 $c(x)$의 모든 원소들을 $2^{wordsize}-q$ 값을 갖도록 구성한 후 알고리즘을 수행시키면 단순 전력 분석 공격을 할 수 있다. 입력 다항식인 $c(x)$의 원소들이 $2^{wordsize}-q$의 값을 갖는 경우 [그림 2]의 제4-8행에서 $j=0$일 때 제6행의 연산 $t_{k+b[j]} \leftarrow t_{k+b[j]}+c_k$는 $2^{wordsize}$ 즉, 오버플로우가 발생하여 0이 된다. 그러므로 제4행의 j값이 $0<j<d$의 범위를 갖는 동안의 제6행의 연산은 이전 단계에서 오버플로우 발생으로 인해 0값을 저장하고 있는 배열 t의 원소들에 의해 $0+x$ $(x\neq 0)$이 되어 단순 전력 분석 공격이 가능해 진다. 따라서 공격자의 어떠한 입력 조작에 대해서도 단순 전력 분석 공격에 강한 컨볼루션 연산을 수행하기 위해서는 후에 5장에서 제시하는 상관계수 전력 분석 공격을 예방하는 대응 방법을 적용하는 것이 바람직하다.

입력: b는 이진 다항식 $a(X)$의 계수들 중 1의 위치를 나타내는 배열, $c(X)$는 일반 다항식; 다항식 $a(X)$, $c(X)$는 N개의 계수를 갖는다.
출력: $a(X)*c(X)$ 연산 결과를 저장한 배열 t

```
1: for 0 ≤ j < 2N do
2:   t_j ← q
3: end for
4: for 0 ≤ j < d do
5:   for 0 ≤ k < N do
6:     t_{k+b[j]} ← t_{k+b[j]} + c_k
7:   end for
8: end for
9: for 0 ≤ j < N do
10:  t_j ← (t_j + t_{j+N}) mod q
11: end for
```

[그림 2] 단순 전력 분석 공격에 대한 대응 방법

IV. NTRU에 대한 상관계수 전력 분석 공격

이 장에서는 단순 전력 분석 공격에 대한 대응 방법이 적용된 컨볼루션 알고리즘을 이용하여 구현한 NTRU가 상관계수 전력 분석 공격에는 취약함을 설명하고 실제로 실험을 통해 Telos 모듈에서 NTRU를 실행시켜 얻은 전력 신호를 측정하여 수집한 데이터를 MATLAB을 이용하여 분석한 상관계수 전력 분석 공격 결과를 제시한다.

4.1 공격 방법

NTRU 기본 컨볼루션 알고리즘에 단순 전력 분석 공격의 대응 방법을 적용하여 덧셈 연산에서 $0+x$ $(x\neq 0)$ 연산을 제거함으로 $0+x$와 $x+y$ $(x,y\neq 0)$ 각각의 경우에 따른 전력 소모 패턴이 구분되지 않도록 할 수는 있으나, 여전히 덧셈 연산 시 피연산자들의 값에 따라 미세한 전력 소모 패턴의 차이가 발생할 수 있다. 공격자는 이러한 미세한 전력 소모 패턴의 차이를 감지할 수 있으므로 NTRU의 복호화 연산 시 단순 전력 분석 공격에 안전한 [그림 2] 알고리즘을 이용하더라도 암호문 $c(X)$를 변화시켜가면서 전력 소모 패턴을 측정하여 상관계수 전력 분석 공격을 할 수 있다. 즉, 공격자가 해밍 디스턴스 모델로 공격한다고 가정할 경우, 공격자는 [그림 2]의 제6행에서 레지스터의 값이 $t_{k+b[j]}$에서 $t_{k+b[j]}+c_k$로 변하는 순간의 해밍 디스턴스를 측정하여 이 해밍 디스턴스와 제6행의 연산이 수행될 때의 전력 소모와의 상관관계를 구해 배열 b의 값을 알아낼 수 있다. 이 때, 암호문 $c(X)$를 변화시키면서 얻은 전력 데이터의 수가 많을수록 데이터 처리 시간이 길어지지만 더 정확한 결과를 얻을 수 있다.

해밍 디스턴스를 관찰하는 레지스터에 해당하는 배열의 인덱스는 [그림 2]의 제5행의 변수 k의 값과 개인키 정보를 담고 있는 배열 $b[j]$에 의해 결정된다. 상관계수 전력 분석 공격을 수행하기 위해서는 $k=0$인 경우에 대해서만 고려하는데 그 이유는 배열의 인덱스 값에 영향을 주면서 예측할 수 있는 $b[j]$의 범위를 최대화하기 위함이다. 예를 들어 $N=8$, $b=[0,3,4,6,7]$이라고 하자. [그림 2]의 제4-8행 연산은 [그림 3]과 같은 단계로 이루어질 것이다. $HW(x)$는 x의 해밍 웨이트를 나타내고 $b[1]-b[0]=b1$이라고 할 때, 공격자는 $b1$을 찾아내기 위해 모든 가능한 경우의 수에 대해 해밍 디스턴스를 구하기 위해 해밍 웨이트 $HW((c_{k+b1}+q)\oplus(c_{k+b1}+c_k+q))$를 계산할 것이다. 다항식 $c(x)$는 $N-1$차 즉, 7차 다항식 이므로 배열 c의 인덱스 범위는 7을 넘지 않는다. 따라서 $0\leq k+b1\leq 7$이며 $b1$은 0이 아니므로 $1\leq b1\leq 7-k$ 범위의 값을 조사하게 된다. 이때, 만약 $k=5$인 경우에 대해서 고려한다면 $b1$의 값의 범위는$1\leq b1\leq 2$ 이므로 원래 값인 3을 찾을 수 없게 된다. 따라서 실제로 발생할 수 있는 모든 가능성 있는 $b1$에 대한 해밍 웨이트를 계산하기 위해서는 $k=0$인 경우를 선택해야 한다.

본 논문에서는 상관계수 전력 분석 공격으로 접근할 경우 해밍 디스턴스와 전력 소모와의 상관관계를 구하기 위해 다음과 같은 피어슨 상관계수(Pearson correlation coefficient) 식을 이용한다.

$$r(<x_1,\ldots,x_S>,<y_1,\ldots,y_S>) = \frac{\sum_{s=1}^{S}(x_s-\bar{x})(y_s-\bar{y})}{\sqrt{\sum_{s=1}^{S}(x_s-\bar{x})^2}\sqrt{\sum_{s=1}^{S}(y_s-\bar{y})^2}} \quad (5)$$

공격자는 식 (5)를 이용하여 해밍 디스턴스와 전력 데이터와의 상관계수를 계산하는데, 상관계수가 1에 가까울수록 해밍 디스턴스와 전력 데이터와의 관계가 밀접함을 나타내고, 0에 가까울수록 관계가 멀어짐을 나타낸다.

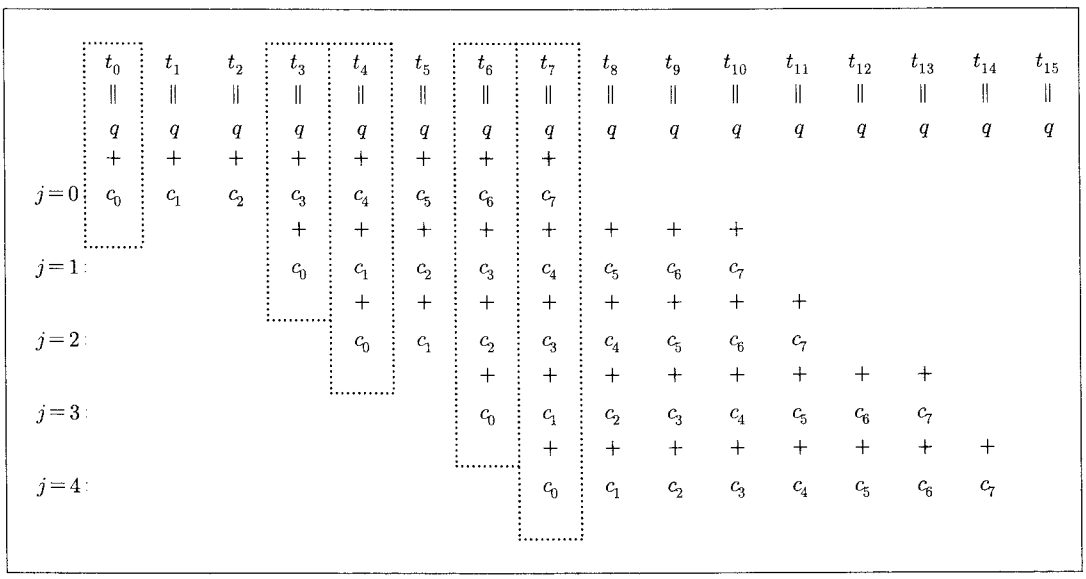
공격자는 먼저 $b[1]-b[0]$의 가능한 모든 값들에 대해 위 상관계수를 계산하여 이중 상관계수 값을 최대로 만드는 값을 $b[1]-b[0]$의 값으로 예측한다. 다음에는 그 값을 바탕으로 $b[2]-b[1]$ 값을 찾는다. 최종적으로 공격자는 $b[d-1]-b[d-2]$를 찾아내고 단순 전력 분석 공격과 같은 방법으로 전수조사를 통해 정확한 배열 b의 값을 찾을 수 있다. [그림 3]에서 볼 수 있듯이 NTRU의 컨볼루션 연산은 배열 b의 값에 따라 더해지는 위치가 결정된다. 만약 공격자가 중간에 배열 b의 값을 잘못 예측하게 되면 그 이후의 값을 예측하는데 연쇄적으로 영향을 미쳐 올바른 값을 구하기 어렵게 된다.

차분 전력 분석 공격으로 접근할 경우에는 해밍 디스턴스와 전력 소모와의 상관관계를 구하기 위해 전력 데이터를 분류한 후 각각 평균에 대한 차분 전력을 이용하면 되므로 상관계수 전력 분석 공격과 동일하게 적용될 수 있다.

4.2 공격 실험 및 결과

본 실험에서는 상관계수 전력 분석 공격을 이용하여 NTRU의 개인키 정보를 얻는다. 소프트웨어로 구현된 NTRU의 다항식 컨볼루션 알고리즘이 실제 모트(mote)에서 수행되는 동안 소비되는 전력 신호를 측정하여 수집한 전력 신호 데이터를 MATLAB을 이용하여 분석한다. 본 실험에서 사용한 모트로는 8MHz이고 10KB RAM, 48KB Flash를 포함하는 MSP430 프로세서를 갖는 Telos 모듈을 이용하였다. NTRU의 다항식 컨볼루션은 NesC로 구현하였고, NTRU의 공개 매개 변수 값으로는 $(N,d,p,q)=(251,48,2,197)$을 사용하였다. 서로 다른 $c(X)$를 이용하여 1000번의 다항식 컨볼루션을 수행하였고 각 $c(X)$에 따른 전력 소비 데이터를 수집하였다.

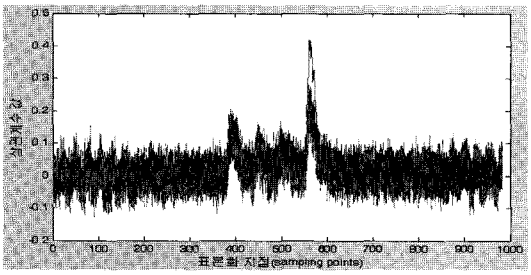
$b[1]-b[0]=b1$을 구하기 위해 i값에 따른 각각의 c_0+c_i+q 와 c_i+q $(1\le i\le 250)$의 해밍 디스턴스를 구하고 각 연산에 대응되는 전력 데이터와의 상관계수를 구한다. 이 때, 상관계수의 값이 최대값을 갖는 인덱스 i가 $b1$을 의미한다. $b1$을 이용해 $b[2]-b[1]=b2$를 구한다. i값에 따른 각각의 $c_0+c_i+c_{b1+i}+q$ 와 c_i+c_{b1+i} $(1\le i\le 250-b1)$의 해밍 디스턴스를 구하고 각 연산



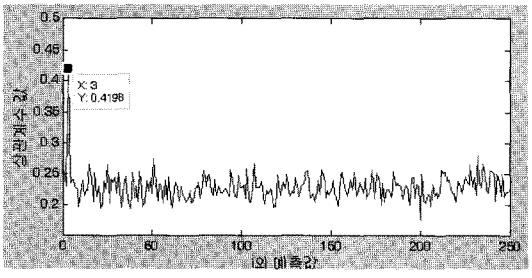
[그림 3] N = 8, b = {0, 3, 4, 6, 7}일 경우 컨볼루션 중간 결과

에 대응되는 전력 데이터와의 상관계수를 계산하면 이 때, 상관계수 값이 최대인 i가 $b2$를 의미한다. 반복적으로 이전에 찾은 배열 b 원소들 간의 상대 위치를 이용하여 해밍 디스턴스를 구할 식을 결정하고 전력 데이터와의 상관계수를 구하면 된다. 배열 b의 원소들 간의 모든 상대 위치를 찾았을 경우, 전수조사를 통해 $b[0]$의 위치를 찾아내면 정확한 배열 b의 값을 복원할 수 있다.

[그림 4]는 $1 \le i \le 250$ 범위를 갖는 모든 i값에 대한 c_0+c_i+q와 c_i+q의 해밍 디스턴스와 각 연산에 대응되는 전력 소비 데이터와의 상관계수 계산 결과를 보여준다. [그림 4]의 가로축은 전력 소비 데이터의 표본화 지점(sampling points), 즉 시간축을 의미하며 세로축은 상관계수 값을 의미한다. 그림에서 가로축의 값이 500과 600 사이일 때 극값이 나타나는 것을 관찰 할 수 있는데 이는 레지스터의 값 c_i+q에 c_0가 더해져 $(c_i+q)+c_0$로 바뀌는 연산이 일어나는 시점을 의미한다. 이와 같이 레지스터의 값이 c_i+q에서 $(c_i+q)+c_0$로 변하는 특정 지점에 대해 i값에 따른 상관계수 값을 그래프로 나타내면 [그림 5]와 같다. [그림 5]의 가로축은 예측하는 i값을 의미하고 세로축은 상관계수 값을 의미한다. [그림 5]에서 볼 수 있듯이 $i=3$일 때 상관계수는 최대값을 갖고 이는 실제 $b[1]-b[0]=3$임을 의미한다.



[그림 4] 전력 소비 데이터에 대한 모든 가능한 i의 값의 상관계수 그래프



[그림 5] i=3에서 극값을 갖는 상관계수 그래프 (∴b1 = 3)

V. NTRU에 대한 상관계수 전력 분석 공격에 대한 대응 방법

5.1 순서의 무작위화(order-randomization)

상관계수 전력 분석 공격을 막기 위한 첫 번째 방안으로 컨볼루션의 입력이 되는 배열 b의 순서를 임의대로 섞어 컨볼루션 연산의 순서가 무작위화 되도록 하는 방법이 있다. 배열 b의 순서를 무작위화함으로써 전력 데이터를 분류하기 위한 연산을 예측하기 어려워지기 때문이다. [그림 6]은 배열 b의 순서를 무작위화하는 연산을 포함하는 다항식 컨볼루션 알고리즘이다. 제4-7행은 순차적으로 변하는 인덱스 j와 임의로 선택된 인덱스 r에 해당하는 배열 b의 값을 서로 교환하는 단계를 보여준다. [그림 2]와 비교하였을 때 [그림 6]은 배열 b의 값을 섞는 제4-7행의 연산만큼 오버헤드가 발생한다.

5.2 데이터의 무작위화(data-randomization)

상관계수 전력 분석 공격을 막기 위한 두 번째 방안으로 피연산자가 되는 데이터를 무작위화 하는 방법이 있다. 즉, 컨볼루션 결과를 저장할 배열 t의 각 원소들을 0이 아닌 무작위수로 초기화 하는 것이다. 단, 배열 t의 각 원소들은 모두 같은 값으로 초기화 할 수도 있

입력: b는 이진 다항식 $a(X)$의 계수들 중 1의 위치를 나타내는 배열, $c(X)$는 일반 다항식; 다항식 $a(X)$, $c(X)$는 N개의 계수를 갖는다.
출력: $a(X)*c(X)$ 연산 결과를 저장한 배열 t

```
1: for 0 ≤ j < 2N do
2:   t_j ← q
3: end for
4: for 0 ≤ j < d do
5:   Generate a random r ( 0 ≤ r < d )
6:   Swap b[j] ↔ b[r]
7: end for
8: for 0 ≤ j < d do
9:   for 0 ≤ k < N do
10:     t_{k+b[j]} ← t_{k+b[j]} + c_k
11:  end for
12: end for
13: for 0 ≤ j < N do
14:   t_j ← (t_j + t_{j+N}) mod q
15: end for
```

[그림 6] 상관계수 전력 분석 공격에 대한 대응 방법 1 : 배열 b의 순서 무작위화

입력: b는 이진 다항식 $a(X)$의 계수들 중 1의 위치를 나타내는 배열. $c(X)$는 일반 다항식; 다항식 $a(X)$, $c(X)$는 N개의 계수를 갖는다.
출력: $a(X)*c(X)$ 연산 결과를 저장한 배열 t

```
1: for 0 ≤ j < 2N do
2:   Generate a random r_j
3: end for
4: for 0 ≤ j < 2N do
5:   t_j ← r_j
6: end for
7: for 0 ≤ j < d do
8:   for 0 ≤ k < N do
9:     t_{k+b[j]} ← t_{k+b[j]} + c_k
10:  end for
11: end for
12: for 0 ≤ j < 2N do
13:  t_j ← t_j − r_j
14: end for
15: for 0 ≤ j < N do
16:  t_j ← (t_j + t_{j+N}) mod q
17: end for
```

〔그림 7〕 상관계수 전력 분석 공격에 대한 대응 방법 2: 배열 t의 원소를 무작위수로 초기화

다. 이는 〔그림 2〕에서 같은 입력 $c(X)$에 대하여 제6행의 피연산자 값이 바뀌도록 함으로써 전력 소모 패턴 또한 매번 바뀌므로 배열 b의 값을 예측하기 어렵게 한다. 배열 t의 원소를 임의의 값으로 초기화 한 후 본래의 $a(X)*c(X)$ 연산 결과를 얻기 위해서는 컨볼루션 연산 후 배열 t의 원소를 초기화 했던 무작위수를 제거하여 연산 결과를 보정해주는 과정이 필요하다. 〔그림 7〕은 배열 t의 각 원소들을 무작위수로 초기화하는 단계를 포함하는 다항식 컨볼루션 알고리즘이다. 〔그림 7〕의 제1-3행은 배열 t의 원소의 개수만큼 임의의 수를 생성하는 단계를 나타내고, 제4-6행은 생성한 무작위수로 배열 t를 초기화 하는 단계를 나타낸다. 제12-14행은 연산 결과 보정을 위한 무작위수 제거 단계를 나타낸다. 〔그림 2〕와 비교하였을 때 〔그림 7〕은 무작위수를 생성하고 초기화 하는 제1-6행, 연산 결과를 보정하는 제12-16행의 연산만큼 오버헤드가 발생한다.

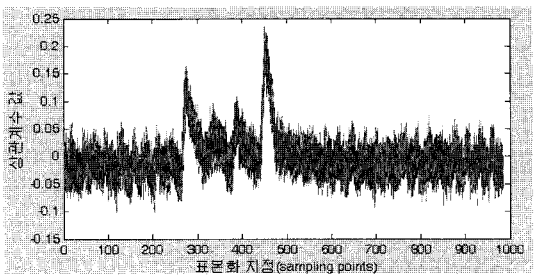
Remark 1. 데이터를 무작위화하기 위해 일반 다항식 $c(X)$에 무작위성(randomness)을 부여할 수도 있다. 임의의 수 r을 선택하여 입력 배열 c에 r을 더한 후 컨볼루션 연산을 수행하면 된다. 이 방법 역시 연산 결과를 보정하기 위해 임의의 수 r을 제거하는 연산이 필요하다. 대응 방법 2와 비슷하게 일반 다항식 $c(X)$에 무작위성을 부여하고, 연산을 보정하는 오버헤드가 발생한다.

Remark 2. 상관계수 전력 분석 공격에 보다 강력한 대응을 하기 위해 앞서 제시한 대응 방법 1과 대응 방법 2를 함께 적용할 수 있다. 피연산자를 무작위화하고 연산 순서를 임의로 섞으면 추가적인 오버헤드가 약간 커질 수 있으나 공격자가 예측할 수 있는 부분이 줄어들어 공격을 방지할 수 있다.

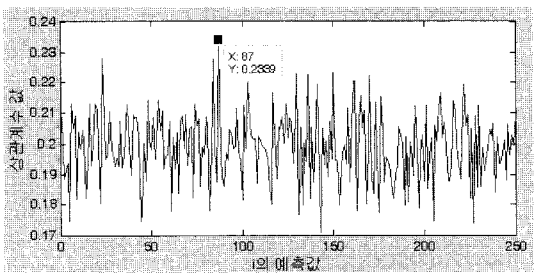
5.3 대응 방법을 적용한 컨볼루션에 대한 공격 실험 및 결과

본 절에서는 상관계수 전력 분석 공격을 예방하기 위한 대응 방법인 데이터의 무작위화의 유효성을 평가하기 위해, 데이터 무작위화가 적용된 컨볼루션 알고리즘 즉 〔그림 7〕에 대해 상관계수 전력 분석 공격을 시도한다. 실험 환경은 4.2절의 실험과 동일하게 소프트웨어로 구현된 NTRU의 다항식 컨볼루션 알고리즘을 실제 모트에서 수행시켜 얻은 전력 신호를 수집하여 MATLAB을 통해 분석하였으며, 앞의 실험에서와 같이 Telos 모듈에서 NesC로 구현하였다.

〔그림 8〕은 $1 \le i \le 250$ 범위를 갖는 모든 i값에 대한 $c_0 + c_i + q$와 $c_i + q$의 해밍 디스턴스와 각 연산에 대



〔그림 8〕 전력 소비 데이터에 대한 모든 가능한 i의 값의 상관계수 그래프



〔그림 9〕 극값이 없는 상관계수 그래프

응되는 전력 소비 데이터와의 상관계수 계산 결과를 보여준다. 가로축은 전력 소비 데이터의 표본화 지점을 의미하며, 세로축은 계산된 상관계수 값을 의미한다. [그림 4]의 경우 표본화 지점이 500~600사이일 경우 0.4에 가까운 상관계수 값을 갖는 i가 존재하지만 [그림 8]의 경우 대부분의 상관계수 값이 최대 0.25를 넘지 못함을 볼 수 있다. [그림 9]는 레지스터의 값이 c_i+q에서 $(c_i+q)+c_0$로 변하는 특정 지점에 대해 예측한 i값에 따른 상관계수 그래프를 나타낸다. 가로축은 i의 예측한 값을 의미하고, 세로축은 계산된 상관계수 값을 의미한다. [그림 5]의 경우 개인키의 1의 위치를 저장하고 있는 배열 b의 원소의 상대적인 차이 값과 값이 일치할 경우 극값을 보이지만 [그림 9]의 경우 데이터를 무작위화함으로써 주변 값들에 비해 상대적으로 높은 극값을 갖는 예측값이 없으며, 최대 극값을 갖는 예측값 87 또한 잘못된 결과이다. 따라서 본 논문에서 제안한 대응 방법 2가 상관계수 전력 분석에 안전함을 보였다. 뿐만 아니라 본 대응법은 모두 다른 랜덤값으로 초기값을 설정함으로 3.2절에서 언급한 특수한 초기값을 넣는 공격에도 안전함을 알 수 있다.

VI. 결론 및 토의

본 논문에서는 소프트웨어로 구현된 NTRU 공개키 암호체계가 전력 분석 공격에 취약함을 보였다. 구체적으로, NTRU에 대한 단순 전력 분석 공격과 상관계수 전력 분석 공격 방법을 이론적으로 제시하였고, 상관계수 전력 분석 공격을 이용해 NTRU 개인키의 복원이 가능함을 실험적으로 증명하였다. 또한 이들 공격을 예방할 수 있는 대응 방법도 제시하였다.

본 연구의 결과는 특히 센서네트워크 등 한정된 자원을 갖는 초소형 장치에 직접 적용될 수 있을 것으로 기대되는데, 기존의 RSA나 ECC에 비해 뛰어난 성능을 보이는 NTRU 암호체계를 보다 안전하게 구현할 수 있는 방안을 제시함으로써, 노드간 키 교환이나 라우팅 등 다양한 상황에서 보다 효율적이고 안전한 통신을 제공하는 것이 가능하게 된다.

그러나, NTRU의 기본 다항식 컨볼루션 알고리즘에 단순 전력 분석 및 상관계수 전력 분석 공격에 대한 대응 방법을 적용시킨 다항식 컨볼루션 알고리즘도 2차 이상의 상관계수 전력 분석 공격에 취약할 가능성이 있음을 고려할 필요가 있다. 그러므로 NTRU에 대한 2차 이상의 상관계수 전력 분석 공격을 실험적으로 증명하는 것과, 공격을 예방하기 위한 대응 방안을 모색하는 방향으로 추가적인 연구가 필요하다.

참 고 문 헌

[1] J. Hoffstein, J. Pipher, and J.H. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem," Algorithmic Number Theory(ANTSIII), LNCS 1423, pp. 267-288, 1998.

[2] W. Whyte et al., "Draft Standard for Public-Key Cryptographic Techniques Based on Hard Problems over Lattices," Oct. 2006.

[3] http://www.ntru.com/cryptolab/faqs.htm

[4] J. Hoffstein and J.H. Silverman, "Optimizations for NTRU," Proceedings of Public Key Cryptography and Computational Number Theory, pp. 1-12, Sep. 2000.

[5] D.V. Bailey, D. Coffin, A. Elbirt, J.H. Silverman, and A.D. Woodbury, "NTRU in constrained devices," CHES 2001, LNCS 2162, pp. 262-272, 2001.

[6] G. Gaubatz, J.P. Kaps, and B. Sunar, "Public key cryptography in sensor networks-Revisited," ESAS 2004, LNCS 3313, pp. 2-18, 2004.

[7] A. Atici, L. Batina, B. Gierlichs, and I. Verbauwhede, "Power analysis on NTRU implementations for RFIDs: First results," In Workshop on RFID Security 2008, pp. 128-139, July 2008.

[8] P.C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," Advances in Cryptology, CRYPTO '96, LNCS 1109, pp. 105-113, 1996.

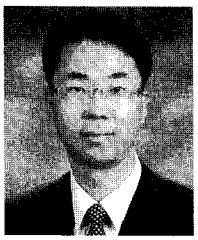
[9] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," Advances in Cryptology, CRYPTO '99, LNCS 1666, pp. 388-397, 1999.

[10] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a

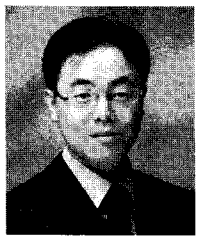
Leakage Model," CHES 2004, LNCS 3156, pp. 16-29, 2004.
[11] T.S. Messerges, "Using Second-Order Power Analysis to Attack DPA Resistant Software," CHES 2000, LNCS 1965, pp. 238-251, 2000.
[12] E. Oswald, S. Mangard, C. Herbst, and S. Tillich, "Practical Second-Order DPA Attacks for Masked Smart Card Implementations of Block Ciphers," CT-RSA 2006, LNCS 3860, pp. 192-207, 2006.

〈著者紹介〉

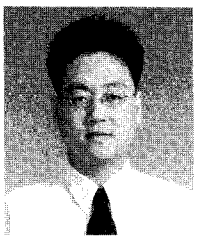
송 정 은 (Jeong Eun Song) 학생회원
2007년 2월: 인하대학교 컴퓨터공학과 졸업(학사)
2009년 2월: 인하대학교 정보공학과 졸업(공학석사)
2009년 3월~현재: 인하대학교 컴퓨터정보공학부 박사과정
〈관심분야〉 암호알고리즘, 부채널분석 등



한 동 국 (Dong-Guk Han) 정회원
1999년: 고려대학교 수학과 졸업(학사)
2002년: 고려대학교 수학과 석사 (이학석사)
2005년: 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 정보보호연구본부 선임연구원
2009년 3월~현재: 국민대학교 수학과 조교수
〈관심분야〉 공개키 암호시스템 안전성 분석 및 고속 구현, 부채널 분석, RFID/USN 정보보호 기술



이 문 규 (Mun-Kyu Lee) 정회원
1996년 2월: 서울대학교 컴퓨터공학과 졸업 (학사)
1998년 2월: 서울대학교 컴퓨터공학과 (공학석사)
2003년 8월: 서울대학교 전기컴퓨터공학부 (공학박사)
2003년 8월~2005년 2월: 한국전자통신연구원 선임연구원
2005년 3월~현재: 인하대학교 컴퓨터정보공학부 조교수
〈관심분야〉 암호알고리즘, 부채널분석 등



최 두 호 (Dooho Choi) 정회원
1994년: 성균관대학교 수학과 졸업(학사)
1996년: 한국과학기술원(KAIST) 수학과 석사(이학석사)
2002년: 한국과학기술원(KAIST) 수학과 박사(이학박사)
2002년 1월~현재: 한국전자통신연구원(ETRI) 정보보호연구본부 선임연구원./팀장
〈관심분야〉 RFID/USN 정보보호 기술, 페어링 기반 암호 이론, 암호시스템 안전성 증명, 비가환군 암호 이론