Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.3, Jun. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.3.551

NTRU 서명 시스템 구현에 대한 오류 주입 공격 및 대응 방안 연구

장 호 철,† 오 수 현, 하 재 철‡
호서대학교

A Study on Attack against NTRU Signature Implementation and Its Countermeasure

Hocheol Jang,† Soohyun Oh, Jaecheol Ha‡
Hoseo University

요 약

최근 양자 컴퓨팅을 활용한 연산 기술이 발달함에 따라 기존 암호 시스템들에 대한 안전성이 위협받고 있다. 이에 따라 양자 컴퓨터를 이용한 분석 공격에도 견딜 수 있는 새로운 포스트 양자 암호시스템(post-quantum cryptosystem)에 대한 연구가 활발하다. 그럼에도 불구하고 NTRU와 같은 격자 기반의 포스트 양자 암호시스템도 구현상에서 발생하는 취약점을 이용하는 오류 주입 공격에 의해 비밀 키가 노출될 수 있음이 밝혀졌다. 본 논문에서는 NTRU 서명 시스템에 대한 기존의 오류 주입 공격 대응 기법을 분석하고 효율성과 안전성이 개선된 새로운 대응 기법을 제안한다. 제안된 대응 기법에 대해 시뮬레이션을 수행한 결과, 오류 주입 검출율이 우수하며 구현이 효율적임을 확인하였다.

ABSTRACT

As the computational technology using quantum computing has been developed, several threats on cryptographic systems are recently increasing. Therefore, many researches on post-quantum cryptosystems which can withstand the analysis attacks using quantum computers are actively underway. Nevertheless, the lattice-based NTRU system, one of the post-quantum cryptosystems, is pointed out that it may be vulnerable to the fault injection attack which uses the weakness of implementation of NTRU. In this paper, we investigate the fault injection attacks and their previous countermeasures on the NTRU signature system and propose a secure and efficient countermeasure to defeat it. As a simulation result, the proposed countermeasure has high fault detection ratio and low implementation costs.

Keywords: Post-quantum cryptosystem, NTRU signature, Convolutional multiplication, Fault injection attack and countermeasure

I. 서 론

현재 널리 사용되고 있는 암호 시스템의 안전성은 소인수 분해 문제나, 이산 대수 문제와 같은 수학적 난제에 기반하고 있다. 따라서 향후 컴퓨팅 능력이 향상되어 큰 합성수에 대해 소인수 분해가 이루어지거나 이산 대수를 찾을 수 있게 된다면 기존의 암호 알고리듬들은 그 안전성이 보장될 수 없다. 최근 주목을 받고 있는 양자 컴퓨팅 기술이 개발되고 일반에게 보급될 경우에는 이 기술이 기존 암호 시스템을

Received(03. 09. 2018), Modified(04. 09. 2018), Accepted(04. 10. 2018)
† 주저자, achagun@gmail.com
‡ 교신저자,jcha@hoseo.edu(Corresponding author)

위협하는 요소가 될 것으로 전망하고 있다. 특히, 1995년 Shor는 소인수분해 문제와 이산로그 문제를 양자 컴퓨터로 해결하기 위한 알고리즘을 제시한 바 있다[1].

이에 따라 양자 컴퓨팅 기술이 도입되더라도 안전성을 보장받을 수 있는 포스트 양자 암호 시스템(post-quantum cryptosystem)에 대한 연구가 활발하다. 그렇지만 포스트 양자 암호가 이론적으로 안전할지라도 실제 디바이스에 구현 시 발생하는 취약점을 고려하지 않을 경우 부채널 분석이나 오류 주입 공격에 의해 비밀 키가 노출될 가능성이 있다. 이와 같이 암호 시스템 구현 시 발생하는 결함을 악용하여 비밀 정보를 얻어내는 공격을 통칭하여 구현 공격(implementation attack)이라고 한다[2-6]. 구현 공격은 수동적인 공격과 능동적인 공격으로 구분되는데 일반적으로 수동적인 공격을 부채널 분석이라고 한다. 능동적인 공격으로는 암호용 디바이스의 동작 과정에서 오류를 주입한 후 비밀 키를 분석하는 오류 주입 공격이 대표적이다.

오류 주입 공격은 암호 알고리즘의 수행 중에 공격자가 고의로 오류를 주입하여 오류 출력을 얻고, 오류 출력에서 나타나는 정보를 기반으로 비밀 키를 찾아내는 공격 기법이다[5-6]. 실제 오류 주입 공격은 AES 및 RSA와 같은 기존의 블록 암호 혹은 공개 키 암호 시스템에 모두 적용 가능하며 이에 관한 대응책도 많이 연구되고 있다.

본 논문에서는 포스트 양자 암호 중 격자 기반의(lattice-based) NTRU 암호 시스템[7-10]에서의 오류 주입 공격 기법[11-13]을 알아보고 이에 대한 대응 방안을 제안한다. 제안하는 대응 방안은 기존 대응 방안보다 연산량과 안전성 측면에서 더 높은 효율성을 가진다. 제안하는 대응 방안은 다항식 계수를 더하는 것과 같은 간단한 연산을 통해 오류를 검출하기 때문에 추가되는 연산량이나 하드웨어 구현 비용을 줄일 수 있다는 장점이 있다. 논문에서는 소프트웨어 방식으로 NTRU 서명 시스템을 구현한 후 제안된 대응 기법에 대한 시뮬레이션 결과, 오류 검출율이 우수하며 구현이 효율적임을 확인하였다.

II. NTRU 서명 시스템

격자 기반의 서명 시스템인 NTRUSign와 NTRUVerify는 디지털 메시지에 대한 서명 및 검증을 위한 시스템이다[8, 9]. NTRU 서명 시스템은 일반적으로 키 생성 알고리즘, 서명 및 검증 알고리즘으로 구성되는데 Fig. 1은 키 생성 알고리즘을 나타낸 것이다. 구체적인 내용은 참고 문헌에 자세히 기술되어 있다[9, 13].

키 생성 알고리듬에서는 소수 N과 다항식 환(polynomial ring)을 $R=\mathbb{Z}[x]/(x^N-1)$ 과 같이 정의한다. 또한, 메시지나 및 서명문의 계수는 모듈러스 q에 의해 처리되는 다항식 환 $R_q=(\mathbb{Z}/q\mathbb{Z}[x])/(x^N-1)$ 을 이용한다. 그리고 두

1. INPUT : Prime integer(N), q, N, $B \geq 0$, (d_f, d_g for binary polynomials), (d for trinary polynomials), and the NTRU lattice type ="standard", or "transpose".
2. Generate B private lattice bases and one public lattice basis: Set $i=B$
   While do $i \geq 0$ do :
   a. Randomly choose binary polynomials f, $g \in R$ with (d_f, d_g) ones. For the parameter sets defined in [14, 15], choose trinary polynomials f, g with $(d+1)$ +1s and d -1s. f needs to be invertible in R_q in case of using NTRU lattice type = "standard" and g needs to be invertible in R_q in case of using NTRU lattice type = "transpose".
   b. Find small polynomials $F, G \in R$ such that $f \star G - F \star g = q$
   c. If NTRU lattice type = "standard", set $f_i = f$ and set $f_i' = F$. If NTRU lattice type = "transpose", set $f_i = f$ and $f_i' = g$.
   Set $h_i = f_i^{-1} \star f_i' \bmod q$.
   Set $i = i-1$.
3. PUBLIC OUTPUT: The input parameters and the public key $h = h_0$.
4. PRIVATE OUTPUT: The set of polynomial $\{f_i, f_i', h_i\}$ for $i = 0 \ldots B$

Fig. 1. Key generation algorithm

다항식 $a(x), b(x) \in R$의 콘볼루션 곱셈(convolution multiplication)인 ★ 연산은 다음과 같이 정의된다.

$$a(x) \star b(x) = c(x) = \sum_{k=0}^{N-1} c_k x^k$$

여기서 $c_k = \sum_{i+j=k \bmod N} a_i b_j \ (0 \le k \le N-1)$이다.

NTRU 서명 시스템에서 양의 정수 d_1과 d_2에 대해 $\tau(d_1,d_2)$는 다음을 만족하는 삼진 다항식의 집합을 의미한다.

$$a(x) \in R: \begin{cases} a(x)\text{의 계수가 1인 계수가 } d_1\text{개 존재} \\ a(x)\text{의 계수가 -1인 계수가 } d_2\text{개 존재} \\ \text{나머지 다른 계수는 모두 0} \end{cases}$$

본 논문에서는 키 생성시 참고 문헌 [14]와 [15]에서 정의된 것과 같이 $(d+1)$개 1의 계수와 d개 -1의 계수를 갖는 삼진 다항식 f와 g를 사용한다. 또한, 다항식 f가 가역성(invertible)을 갖게 선택함으로써 "standard"형 NTRU 서명 시스템을 사용하는 것으로 가정한다.

추가적으로 $\lfloor a \rceil$는 a와 가장 가까운 정수 값으로 반올림하는 라운드(round) 연산이고, 다항식 $a(x)$의 Centered norm 연산 $\| a(x) \|^2$의 정의는 다음과 같다.

$$\| a(x) \|^2 = \sum_{i=0}^{N-1} (a_i - \mu_a)^2 = \sum_{i=0}^{N-1} a_i^2 - \frac{1}{N} (\sum_{i=0}^{N-1} a_i)^2$$

여기서 $\mu_a = (1/N) \sum_{i=0}^{N-1} a_i$로 $a(x)$의 계수의 평균이다. 또한, 다항식 환 R상에서 주어진 두 개의 다항식 a_0와 a_1에 대한 Centered norm은 $\| (a_0, a_1) \|^2 = \| a_0 \|^2 + \| a_1 \|^2$를 만족하는 양의 실수를 의미한다.

그림에서 보는 바와 같이 NTRU 서명 시스템의 키 생성 과정에서는 정수 N, q, $\mathbb{N}$, $B \ge 0$, NTRU type을 입력으로 한다. 여기서 N은 다항식의 차수가 $N-1$차 까지임을 나타내고, $\mathbb{N}$는 Norm-bound 값이다. 서명 시스템의 type에 따라 생성된 h_0를 사용자의 공개 키로 설정하고, 사용자의 비밀 키 값으로 $i = 0,...,B$에 대한 $\{f_i, f_i', h_i\}$을 생성한다.

다음 Fig. 2는 "standard"형 NTRU 서명 알고리듬을 나타낸 것으로 여기에서 D는 서명하고자 하는 전자 문서를 나타낸다. 문서 D와 비밀 키 집합 $\{f_i, f_i', h_i\}$를 입력으로 하여 정의된 B값에 따라 반복 연산을 수행한다. 이 서명 과정에서 주 연산은 콘볼루션 곱셈이며 알고리즘을 종료하면 서명문 (D, r, s)를 출력하게 된다. 그림의 단계 4와 단계 5에서 사용한 $\lfloor \frac{a \star b}{q} \rceil$ 연산은 두 다항식의 콘볼루

1. INPUT: A digital document D and the private key set $\{f_i, f_i', h_i\}$ for $i = 0...B$
2. Set $r = 0$.
3. Set $s = 0$, $i = B$. Encode r as a bit string. Set $m_0 = H(D \| r)$, where "$\|$" denotes the concatenation. Set $m = m_0$.
4. Perturb the point using the private lattices:
   While $i \ge 1$:
   a. $A = \lfloor \frac{-f_0' \star m}{q} \rceil$, $B = \lfloor \frac{f_0 \star m}{q} \rceil$, $s_i = A \star f_i + B \star f_i' \bmod q$
   b. Set $m = s_i \star h_i - h_{i-1} \bmod q$.
   c. Set $s = s + s_i$. Set $i = i - 1$.
5. Sign the perturbed point using the lattice public key:
   $A = \lfloor \frac{-f_0' \star m}{q} \rceil$, $B = \lfloor \frac{f_0 \star m}{q} \rceil$,
   $s_0 = A \star f_0 + B \star f_0' \bmod q$,
   $s = s + s_0$
6. Check the signature:
   a. Set $b = \| (s, s \star h - m_0 \bmod q) \|$.
   b. if $b \ge \mathbb{N}$. Set $r = r + 1$ and go to step 3.
7. OUTPUT: The triplet (D, r, s).

Fig. 2. NTRUSign algorithm

션 곱셈을 수행한 각 계수를 q로 나눈 후 반올림한 계수를 갖는 다항식을 의미한다.

다음 Fig. 3은 NTRU 서명 검증을 수행하는 NTRUVerify 알고리즘이다. 서명 알고리즘의 출력인 서명문 (D, r, s)과 공개 키 h를 입력으로 연산을 수행하는데, 검증 값 b가 유효한지 그렇지 않은지에 대한 여부를 검사하여 송신자 서명의 정당성을 판별한다.

1. INPUT: A signed document (D, r, s) and the public key h.
2. Encode r as a bit string. Set $m_0 = H(D \| r)$.
3. Set $b = \| (s, s \star h - m_0 \bmod q) \|$.
4. OUTPUT: "valid" if $b < \mathbb{N}$, "invalid" otherwise.

Fig. 3. NTRUVerify algorithm

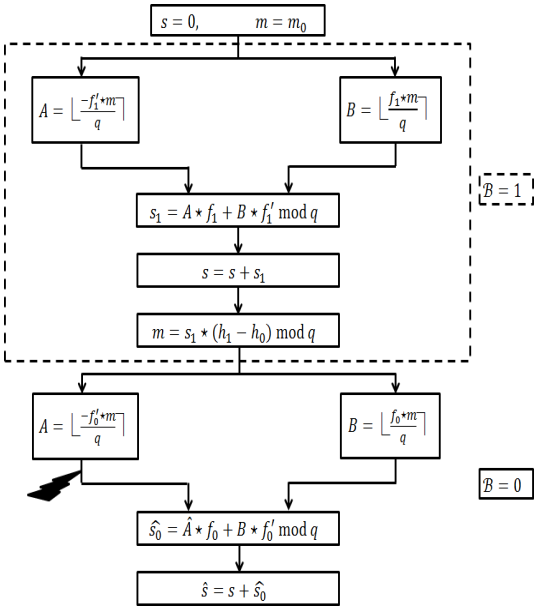

Fig. 4. Fault attack model on NTRUSign

III. NTRU 서명에 대한 오류 주입 공격

3.1 오류 주입 공격 모델

NTRUSign에 대한 기존 오류 주입 공격에서는 서명 값 s_0를 연산하기 전 A 또는 B에 오류를 주입하고 정상 서명과의 차분을 이용해 비밀 키를 찾아낸다[13]. Fig. 4는 NTRUSign에 대한 오류 주입 공격 모델을 나타낸 것이다.

해당 모델은 NTRUSign의 type이 "standard"일 경우에 해당하는 모델이다. 서명 type이 "transpose"일 경우 공개 키와 f_0값 만으로 g_0를 계산할 수가 없어 A에 대한 오류 주입 뿐 아니라 B에 대한 오류 주입을 추가적으로 수행해야 완전한 비밀 키를 찾을 수 있다.

NTRUSign의 "standard" 모델을 가정하고 중간 값 A에 오류가 주입되었을 때 출력되는 오류 서명 $\hat{s} = s + \hat{s_0}$로 표현할 수 있다. 단, 여기서 $\hat{s_0} = \hat{A} \star f_0 + B \star f_0' \bmod q$이며 $\hat{A} = A + \epsilon \bmod q$이다. 공격자는 서명 연산을 두 번 수행하면서 얻은 오류 서명 $\hat{s}$과 정상 서명 s'의 차분 $\triangle s = \hat{s} - s'$를 계산하는데 $\hat{s} - s' = (s + \hat{s_0}) - (s + s_0) = \hat{s_0} - s_0$이 되고, 이는 $\triangle s = \epsilon \star f_0 \bmod q$임을 알 수 있다.

따라서 공격자는 가능한 모든 값의 오류를 주입하여 $f_0 = \epsilon^{-1} \star \triangle s \bmod q$를 계산하여 비밀 키 f_0를 획득한다. 이 경우 주의할 점은 오류 다항식 ϵ이 $\bmod q$상에서의 역 $\epsilon^{-1} \bmod q$을 가지고 있는 경우에만 공격이 가능하므로 오류를 반복 주입하여 모든 가능한 ϵ에 대한 비밀 키를 찾아내야 한다.

NTRUSign 서명 시스템에 오류 주입 공격의 성공 확률을 분석해 보면 오류 다항식 ϵ이 역원을 가질 확률은 $(1-1/p)(1-1/p^n)^{(N-1)/n}$이다. 여기서 $q = p^l$이며 n은 $p^n = 1 \bmod N$이 되는 가장 작은 정수를 의미한다[15]. 즉, R_q에 있는 오류 다항식 ϵ이 역원을 가질 확률은 약 $(1-1/p)$ 정도이다.

3.2 NTRUSign에 대한 오류 주입 공격 대응 기술

NTRUSign에서의 오류 주입 공격에 대한 대응 방안으로 순환 이동 메시지(cyclically shifted message) 재연산 방법이 제시된 바 있다[13]. 다음 Fig. 5는 오류 주입 공격에 대응할 수 있는 순환 이동 메시지 방법을 도식화한 것이다.

이 대응책에서는 원 메시지를 순환 이동 후 서명을 한 번 더 수행하고 그 결과를 다시 역 순환시킴으로써 오류 주입 여부를 판단한다. 결국, 이 방법은 오류를 100% 검출 가능하다는 장점이 있지만 서명

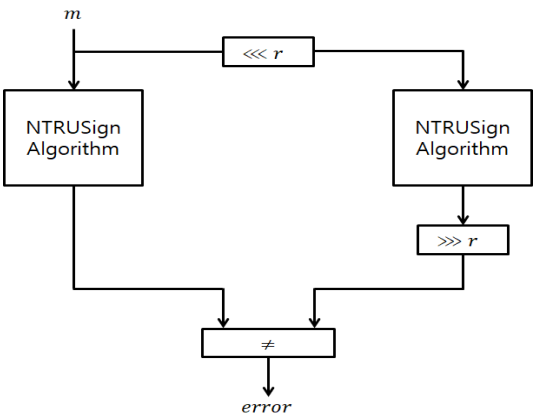

Fig. 5. Fault detection by recomputing with cyclically shifted messages

과 동일한 연산을 한 번 더 수행하므로 연산 오버헤드가 크다는 단점이 있다

순환 이동 메시지 방법은 오류를 검출하는 방법인 것에 비해 오류 감염(error infection) 개념을 이용한 대응책을 제안하기도 하였다[13]. 오류 감염 기반의 대응책에서는 서명 알고리듬 전체를 검사하는 것이 오버헤드가 크므로 중간 값 A와 B만 검사하는 재연산(recomputation) 방법을 사용하였다.

구체적인 과정을 보면 서명 과정에서 A와 B를 계산 한 후 다음 수식을 추가하여 계산한다.

$$AA = \lfloor \frac{-f_0' \star m}{q} - A \rceil$$

$$BB = \lfloor \frac{f_0 \star m}{q} - B \rceil$$

위 식을 연산한 후 $z_1 = \sum_{i=0}^{N-1} |AA_i|$, $z_2 = \sum_{i=0}^{N-1} |BB_i|$를 연산한다. 그리고 이 값을 이용하여 최종 서명 값 $s = q^{\left\lceil \frac{z_1+z_2}{z_1+z_2+1} \right\rceil}(s+s_0) \bmod q$을 생성하게 된다. 만약, 서명 s에 오류가 없다면 z_1과 z_2는 모두 0이 되고 정상 서명이 출력된다. 그러나 A, B, AA 혹은 BB 연산 시 오류가 발생하게 되면 최종적으로 $s=0$이 출력된다.

그러나 상기한 오류 감염 기반 대응책은 서명 전체에 대한 오류 주입 여부를 추출하는 것이 아니라 서명의 중간 값인 $A = \lfloor \frac{-f_0' \star m}{q} \rceil$ 와 $B = \lfloor \frac{f_0 \star m}{q} \rceil$ 를 계산하는 과정에서 발생한 오류만 검출이 가능하다. 그 이유는 A, B, AA 그리고 BB 계산이 끝난 후 수행하는 $s_0 = A \star f_0 + B \star f_0' \bmod q$를 계산하기 위해 읽어오는 A나 B에 대한 오류 주입은 검출할 수 없기 때문이다.

결국, s_0값을 계산하는 과정에서 A나 B에 주입된 오류는 z_1이나 z_2를 만드는 것과 상관이 없으므로 오류가 감염되는 효과를 얻을 수 없어 서명 전체 구간에 걸쳐 오류 주입 공격을 방어할 수 없다는 결정적인 보안 취약점을 가지고 있다.

IV. 제안하는 대응 방안

4.1 NTRUSign에 오류 주입 공격 대응 방안 제안

NTRUSign 알고리듬에 대한 오류 주입공격 대응책을 설계하기 위해 다음과 같은 보조 정리 1과 2를 이용한다.

- Lemma 1

임의의 두 다항식 $A(x)$, $B(x)$에 대해 $C(x) = A(x) \star B(x) \bmod q$ 일 때 $\sum_{i=0}^{N-1} C_i = \sum_{i=0}^{N-1} A_i \times \sum_{i=0}^{N-1} B_i \bmod q$ 와 같다.

(증명) $C(x) = A(x) \star B(x) \bmod q$연산을 행렬(matrix) 형태로 표현하면 다음과 같다.

$$\begin{pmatrix} C_0 \\ C_1 \\ . \\ . \\ . \\ C_{N-1} \end{pmatrix} = \begin{pmatrix} A_0 & A_{N-1} & \cdots & A_1 \\ A_1 & A_0 & \cdots & A_2 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ A_{N-1} & A_{n-2} & \cdots & A_0 \end{pmatrix} \begin{pmatrix} B_0 \\ B_1 \\ . \\ . \\ . \\ B_{N-1} \end{pmatrix}$$

$$= \begin{pmatrix} (B_0A_0 + B_1A_{N-1} + \cdots + B_{N-1}A_1) \bmod q \\ (B_0A_1 + B_1A_0 + \cdots + B_{N-1}A_2) \bmod q \\ . \\ . \\ . \\ (B_0A_{N-1} + B_1A_{N-2} + \cdots + B_{N-1}A_0) \bmod q \end{pmatrix}$$

따라서 다음과 같은 등식이 성립한다.

$$(\sum_{i=0}^{N-1} C_i) \bmod q = (\sum_{i=0}^{N-1} A_i \times \sum_{i=0}^{N-1} B_i) \bmod q \quad \square$$

• Lemma 2

다항식 $A(x)$와 $\tau(d_f, d_f-1)$을 만족하는 삼진 다항식 $f(x)$에 대해 $C(x)=A(x)\star f(x) \bmod q$ 일 때 $\sum_{i=0}^{N-1} C_i = \sum_{i=0}^{N-1} A_i \bmod q$ 와 같다.

(증명) 보조 정리 2를 만족하는 콘볼루션 곱셈에서 $\sum_{i=0}^{N-1} f_i = 1$ 이 되는 특수한 경우이므로 다음과 같은 등식이 성립한다.

$$\sum_{i=0}^{N-1} C_i = (\sum_{i=0}^{N-1} A_i \times \sum_{i=0}^{N-1} f_i) = (\sum_{i=0}^{N-1} A_i) \bmod q \quad \square$$

다음 Fig. 6은 NTRUSign에서의 오류 주입 공격을 방어하기 위해 제안하는 대응 알고리듬을 나타낸 것으로 Fig. 2의 단계 5를 기술한 것이다. 제안하는 대응 방안에서는 기본적으로 두 다항식 간의 콘볼루션 곱셈의 결과 다항식 계수의 합은 원래 두 다항식 계수의 합을 곱한 값이 같다는 보조 정리를 이용하고 있다.

5. Sign the perturbed point using the lattice public key:

5-1) $A = \lfloor \frac{-f_0' \star m}{q} \rceil$

5-2) $B = \lfloor \frac{f_0 \star m}{q} \rceil$

5-3) $s_0 = A \star f_0 + B \star f_0' \bmod q$

5-4) $z_1 = \sum_{i=0}^{N-1} s_{0_i} \bmod q$

5-5) $C = \lfloor \frac{-f_0' \star m}{q} \rceil$

5-6) $D = \lfloor \frac{f_0 \star m}{q} \rceil$

5-7) $z_2 = \sum_{i=0}^{N-1} C_i + (\sum_{i=0}^{N-1} D_i \times \sum_{i=0}^{N-1} f'_{0_i}) \bmod q$

5-8) if$(z_1 = z_2)$ then $s = s + s_0$
else Return(error)

Fig. 6. Proposed countermeasure for secure NTRUSign

그림에서 제안한 오류 주입 공격 대응 방안에서 f_0가 삼진 다항식이고 계수의 합이 1이므로 보조 정리 2에 의해 $\sum_{i=0}^{N-1}(A\star f_0)_i = \sum_{i=0}^{N-1} A_i$ 가 성립한다는 점을 활용한다. 즉, $A\star f_0$와 C의 계수의 합이 같으므로 $\sum_{i=0}^{N-1} A_i = (\sum_{i=0}^{N-1} C_i) \bmod q$가 성립하게 된다.

마찬가지로, B와 f'_0 각각의 계수의 합을 곱한 결과와 $B\star f'_0$를 수행한 결과의 계수의 합은 보조 정리 1에 의해 다음과 같다.

$$\sum_{i=0}^{N-1}(B\star f'_0)_i = \sum_{i=0}^{N-1} B_i \times \sum_{i=0}^{N-1} f'_{0_i}$$

따라서 검사 기법인 $\sum_{i=0}^{N-1} D_i \times \sum_{i=0}^{N-1} f'_0$와 알고리즘에서의 연산 $\sum_{i=0}^{N-1} B_i \times \sum_{i=0}^{N-1} f'_{0_i}$의 값이 같게 된다. 결국, 다항식 s_0의 계수의 전체 합은 다음과 같다.

$$z_1 = \sum_{i=0}^{N-1} s_{0_i} = \sum_{i=0}^{N-1} A_i + \sum_{i=0}^{N-1} (B\star f_0') \bmod q$$

그리고 다항식 C의 계수의 합과 D의 계수의 합과 관계식은 다음과 같이 정리될 수 있다.

$$z_2 = \sum_{i=0}^{N-1} C_i + (\sum_{i=0}^{N-1} D_i \times \sum_{i=0}^{N-1} f'_{0_i}) \bmod q$$

그러므로 정상적인 연산이 이루어지면 z_1은 z_2와 같은 값이 되며, 보조 정리에 의해 그림과 같은 대응 방안을 이용하여 서명 시 발생할 수 있는 오류를 검출할 수 있다.

본 논문에서는 계수를 더한 두 값의 비교를 통해 오류 여부를 확인하는 오류 검출(error detection) 기법을 사용하였지만 오류 감염 기법으로 쉽게 바꿀 수 있기 때문에[16] 표현이 용이한 오류 검출 형식으로 표현하였다. 오류 감염 기법을 적용하기 위해서는 다음과 같은 연산 기법을 사용할 수 있다.

$$s = q^{\lceil (z_1+z_2)/(z_1+z_2+1) \rceil} (s+s_0) \bmod q.$$

제안하는 대응 방안은 NTRUSign에서의 기존 오류 주입 대응 방안인 순환 이동 메시지 재연산 기법에 비해 연산 구간이 적어 연산량 측면에서 효율적이다. 또한, 기존 오류 감염 기반 방법으로 검출할 수 없었던 구간 즉, $s_0 = A \star f_0 + B \star f_0' \bmod q$ 연산 시 발생하는 오류도 쉽게 검출할 수 있다는 장점이 있다. 그러나 제안 방식은 다항식 A의 계수가 여러 개 변경되었음에도 같은 합의 결과, 즉 동일한 z_1이 나오는 경우에는 오류를 검출할 수 없다.

4.2 대응 방안에 대한 비교 분석

NTRUSign에서의 기존의 두 가지 대응 방안과 제안하는 대응 방안을 효율성과 안전도 측면에서 비교한다. 연산량을 비교함에 있어 M은 다항식 간의 콘볼루션 곱셈 수를 의미하고 다항식의 계수를 더하는 연산 수는 S로 표시하였다. Table 1에서는 원래 NTRUSign 서명에 필요한 연산이 $4M$임을 가정할 때 대응책을 구현하기 위해 추가되는 연산량을 표시한 것이다.

순환 이동 메시지 재연산 기법은 $4M$의 연산이 추가되므로 연산량이 거의 100% 증가하는 기법이며 오류 감염 기반 대응책은 $2M+2S$정도의 연산량 추가가 필요하므로 약 50%이상의 계산 비용이 추가된다. 제안 방식은 $2M+4S$정도로 오류 감염 기반 방법에 비교해서는 2번 정도의 다항식 계수를 더하는 연산만 추가된다.

그러나 다항식의 계수를 더하는 연산은 콘볼루션 곱셈 연산에 비하면 거의 무시할 정도로 적은 연산이다. 그러나 상기한 바와 같이 오류 감염 기반 방법은 s_0 연산시 발생하는 오류 주입을 검사할 수 없다는 결정적인 문제점을 안고 있다. 따라서 제안하는 대응 방안은 서명 전체 구간에 대한 검사를 수행 가능하고 연산 오버헤드가 적어 시스템을 효율적으로 구현할 수 있다.

Table 1. Comparison of countermeasures for fault-resistant NTRUSign

Countermeasures	Additive computational cost	Remarks
Fault detection by recomputing with cyclically shifted messages[13]	$4M$	High cost
Fault infection method[13]	$2M+2S$	Not Secure
Proposed method	$2M+4S$	

V. 오류 주입 공격 대응책 시뮬레이션

본 장에서는 오류 주입 공격 및 대응 방안의 검증을 위한 시뮬레이션을 수행하고 그에 대한 결과를 설명한다. 시뮬레이션 환경은 다음과 같다.

- 운영체제 : Windows 10 Pro
- 서명 시스템 구현 언어 : Java
- 사용 도구 : Eclipse

NTRUSign에 대한 오류 주입 공격 시뮬레이션은 다음과 같이 진행하였다. NTRUSign에 사용된 파라미터는 $N=157$, $q=256$, $B=0$, NTRU type은 standard이며 $N=200$으로 설정하였다. Fig. 7은 다음과 같은 오류 주입 공격 시뮬레이션을 통해 얻은 결과를 나타낸 것이다. 오류 주입 공격은 Fig. 2의 단계 5에서 A 혹은 B 값을 계산하거나 저장할 때 오류를 주입하면 성공할 수 있는데, 시뮬레이션에서는 A값을 저장하는 레지스터에 오류를 주

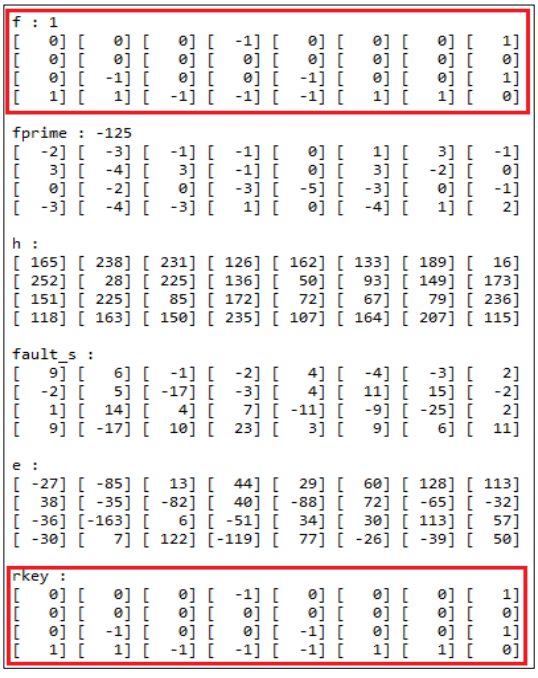

Fig. 7. Simulation of fault attack on NTRUSign

입하는 것을 가정하였다.

〔 오류 주입 공격 단계 〕

① 서명 알고리즘을 수행하여 정상 서명 획득

② 오류 주입 공격 시 사용하기 위한 f^{-1} 계산

③ 서명 알고리즘을 한 번 더 수행하며 A에 오류를 주입한 서명 획득

④ $f_0 = \epsilon^{-1} \star \triangle s \bmod q$ 이므로, ϵ이 $\bmod q$상에서 역을 가질 때 까지 오류를 반복 주입

⑤ 오류 서명과 정상 서명의 차분에서 f_0를 획득 가능

그림에서의 다항식은 모두 하위 32개의 다항식 계수만을 표현하였으며, fault_s가 오류 서명, e가 오류 다항식 ϵ이며 rkey가 복구된 키 f_0이다. 시뮬레이션 결과를 통해 상기한 오류 주입 공격을 기법을 이용하면 비밀 키가 충분히 복구될 수 있음을 알 수 있다.

제안하는 대응 방안을 적용하여 오류 주입 공격을 탐지 가능한지 시뮬레이션 한 결과는 Fig. 8과 같다. 그림에는 서명이 정상적으로 이루어지게 되면 $\sum_{i=0}^{N-1} C_i + (\sum_{i=0}^{N-1} D_i \times \sum_{i=0}^{N-1} f'_0) \bmod q$ 값과 서명 s의 계수의 합이 같다는 것을 확인할 수 있었다. 또한, 시뮬레이션에서 오류를 주입하게 되면 다항식의 계수의 합이 정상 서명 시의 결과와 다르기 때문에 오류가 탐지됨을 확인할 수 있다.

오류 주입 공격은 다항식에 주입된 계수의 수에 따라 공격에 성공할 확률이 달라진다. 만약, A라는 다항식 중 오류가 주입된 계수가 하나인 경우($t=1$)는 오류가 주입된 계수가 두 개인($t=2$)경우와 비교하여 ϵ의 역수가 존재하더라도 실제 비밀 키를 찾는 공격 성공 확률이 달라지기 때문이다. 실험에서는 다항식에 주입되는 계수의 수에 따라 공격이 성공할 확률이 어느 정도인지를 측정하였다.

Fig. 9는 다항식에 오류 주입 계수가 하나일 때($t=1$)를 가정하고 공격이 성공할 가능성과 제안 대응 방식을 적용했을 경우의 대응 가능성을 시뮬레이션한 그림이다.

그림에서 Total count는 오류 주입 공격을 시도한 횟수를 나타낸다. Detection count는 오류를 주입했을 경우 ϵ의 역수가 존재하고 오류 주입 공격이 성공할 수 있는 경우의 수를 의미한다. 또한, Attack count는 제안하는 대응 기법을 적용했을 때 오류 주입 공격 성공 횟수이다. 그림에서 보는 바

```
f :
[   0] [   0] [   0] [  -1] [   0] [   0] [   0] [   1]
[   0] [   0] [   0] [   0] [   0] [   0] [   0] [   0]
[   0] [  -1] [   0] [   0] [  -1] [   0] [   0] [   1]
[   1] [   1] [  -1] [  -1] [  -1] [   1] [   1] [   0]

fPrime :
[  -2] [  -3] [  -1] [  -1] [   0] [   1] [   3] [  -1]
[   3] [  -4] [   3] [  -1] [   0] [   3] [  -2] [   0]
[   0] [  -2] [   0] [  -3] [  -5] [  -3] [   0] [  -1]
[  -3] [  -4] [  -3] [   1] [   0] [  -4] [   1] [   2]

Sum of Coefficient (A(x) + (B(x) * f'(x)): 196
Sum of Coefficient (C(x) + (D(x) * f'(x)) : 196
s(0) mod q : 196
s : 196
[  10] [   6] [  -1] [  -2] [   4] [  -4] [  -3] [   2]
[  -2] [   5] [ -17] [  -3] [   4] [  11] [  15] [  -2]
[   1] [  14] [   4] [   7] [ -11] [  -9] [ -25] [   2]
[   9] [ -17] [  10] [  23] [   3] [   9] [   6] [  11]

Sum of Coefficient (fault_s) : 193

Fault is Detected!
```

Fig. 8. Simulation of proposed countermeasure

```
f : 1
[   0] [   0] [   0] [  -1] [   0] [   0] [   0] [   1]
[   0] [   0] [   0] [   0] [   0] [   0] [   0] [   0]
[   0] [  -1] [   0] [   0] [  -1] [   0] [   0] [   1]
[   1] [   1] [  -1] [  -1] [  -1] [   1] [   1] [   0]

fprime : -125
[  -2] [  -3] [  -1] [  -1] [   0] [   1] [   3] [  -1]
[   3] [  -4] [   3] [  -1] [   0] [   3] [  -2] [   0]
[   0] [  -2] [   0] [  -3] [  -5] [  -3] [   0] [  -1]
[  -3] [  -4] [  -3] [   1] [   0] [  -4] [   1] [   2]

h :
[ 165] [ 238] [ 231] [ 126] [ 162] [ 133] [ 189] [  16]
[ 252] [  28] [ 225] [ 136] [  50] [  93] [ 149] [ 173]
[ 151] [ 225] [  85] [ 172] [  72] [  67] [  79] [ 236]
[ 118] [ 163] [ 150] [ 235] [ 107] [ 164] [ 207] [ 115]

A + (B * fp): 196
s : 196
[  10] [   6] [  -1] [  -2] [   4] [  -4] [  -3] [   2]
[  -2] [   5] [ -17] [  -3] [   4] [  11] [  15] [  -2]
[   1] [  14] [   4] [   7] [ -11] [  -9] [ -25] [   2]
[   9] [ -17] [  10] [  23] [   3] [   9] [   6] [  11]

Total count : 10000
detection count : 3209
attack count : 0
```

Fig. 9. Experimental result on possibility of successful fault attack ($t=1$)

와 같이 제안 대응책을 적용하면 오류 주입 공격을 100% 검출할 수 있음을 알 수 있다.

시뮬레이션 결과, 오류 주입 공격 대응책이 적용되지 않은 상태에서 다항식에 오류가 주입된 개수가 하나인 경우에는 10,000개의 오류 중에서 3,209번만 비밀 키 추출이 가능하다. 그렇지만 대응 기법에서는 모든 주입된 오류가 검출되어 공격이 성공할 수 없음을 보이고 있다.

Table 2는 NTRUSign에서의 다항식 계수의 오류 개수와 이에 따른 ϵ의 역수가 존재하는 횟수 그리고 공격이 성공할 수 있는 조건의 존재 비율을 비교한 것이다. 표에서 보는 것처럼 오류가 주입되는 다항식의 계수의 수가 적을수록 비밀 키를 찾는 성공 확률이 높음을 알 수 있다. $t=1$인 경우는 약 32.1%, $t=2$인 경우에는 약 8.7%의 성공 확률을 보였지만 계수가 3개가 변경되는 경우에는 성공 확률이 1.3%대로 낮아짐을 알 수 있다. 이 결과는 오류 주입 시에도 특정한 계수에 하나의 오류를 주입할 수 있을 정도의 정밀한 오류 주입 기법이 성공 확률이 높다는 것을 의미한다. 그러나 대응 기법을 적용한 경우에는 오류 주입 계수의 수와 관계없이 100% 오류를 탐지할 수 있었다.

Table 2. Possibility of successful fault injection attack according to the number of faulty coefficients
($N=157, q=256, B=0, standard$, $\mathbb{N}=200$)

t	The number of trial	ϵ			Possibility of successful fault injection
		Non-invertible	Invertible		
			Fail	Pass	
1	10000	5001	1790	3209	0.3209
2	10000	4901	4230	869	0.0869
3	10000	5001	4868	131	0.0131

VI. 결 론

양자 컴퓨팅 기술을 암호 해독에 활용할 경우, 기존 암호시스템에 대한 안전성을 보장할 수 없기 때문에 포스트 양자 암호에 대한 관심이 증가하고 있다. 그러나 포스트 양자 암호가 이론적으로 안전할지라도 부채널 공격이나 오류 주입 공격과 같은 구현 공격에 취약할 가능성이 있기 때문에 이에 대한 대응 방안이 필요하다.

본 논문에서는 격자 기반 포스트 양자 암호인 NTRU 서명 시스템에 대한 오류 주입 공격과 기존 대응 방안들을 살펴보고 새로운 대응 방안을 제안했다. NTRUSign에서는 다항식의 계수에 오류를 주입하는 방법으로 비밀 키를 공격할 수 있음이 밝혀진 상태이다. 본 논문에서는 이러한 오류 주입 공격에 대응할 수 있는 방안을 제시하고 실제로 이 대응 방법이 구현의 효율성면에서 타 방식보다 효과적임을 증명하였다. 또한, 컴퓨터 시뮬레이션을 통해 오류 주입 공격의 위험성과 대응 기법이 오류 검출 능력이 우수함을 확인하였다.

References

[1] P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," Proceedings of the 35th Annual Symposium on Foundations of Computer Science, pp. 124-134, 1994.

[2] P. Kocher, "Timing Attacks on Implementation of Diffie-Hellman, RSA, DSS, and Other Systems," CRYPTO'96, LNCS 1109, pp. 104-113, 1996.

[3] J. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," CHES'99, LNCS 1717, pp. 292-302, 1999.

[4] T. Messerges, E. Dabbis, and R. Sloan, "Power analysis attacks of modular exponentiation in smartcard," CHES'99, LNCS 1717, pp. 144-157, 1999.

[5] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," CRYPTO'97. LNCS 1294, pp. 513 - 525, 1997.

[6] D. Boneh, R. DeMillo, and R. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults," EUROCRYPT'97, LNCS 1233, pp.

37-51, 1997.

〔7〕 J. Hoffstein, J. Pipher, and J. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem," ANTS'98, LNCS 1423, pp. 267-288, 1998.

〔8〕 J. Hoffstein, J. Pipher, and J. Silverman, "NSS: An NTRU lattice-based signature scheme," EUROCRYPT'01, LNCS 2045, pp. 211 - 228, 2001.

〔9〕 J. Hoffstein, N. Graham, J. Pipher, J. Silverman, and W. Whyte, "NTRUSign: Digital signatures using the NTRU lattice," CT-RSA'03, LNCS 2612, pp. 122-140, 2003.

〔10〕 IEEE P1363.1: "Public-Key Cryptographic Techniques Based on Hard Problems over Lattices," version D12, 2008.

〔11〕 M. Taha and T. Eisenbarth. "Implementation Attacks on Post-Quantum Cryptographic Schemes," IACR Cryptology ePrint Archive 2015/1083, 2015.

〔12〕 A. Kamal and A. Youssef., "Strengthening hardware implementations of NTRUEncrypt against fault analysis attacks," Journal of Cryptographic Engineering, pp. 227 - 240, 2013.

〔13〕 A. A. Kamal and A. M. Youssef. "Fault analysis of the NTRUSign digital signature scheme," Cryptography and Communications, pp. 131 - 144, 2012.

〔14〕 J. Hoffstein, N. Howgrave-Graham, J. Pipher, and W. Whyte, "Practical lattice-based cryptography: NTRUEncrypt and NTRUSign," The LLL Algorithm : Survey and Applications Information Security and Cryptography, pp. 349-390, Springer, 2010.

〔15〕 J. Hoffstein, N. Howgrave-Graham, J. Pipher, and W. Whyte, "Performance improvements and a baseline parameter generation algorithm for NTRUSign," In Proc. of Workshop on Mathematical Problems and Techniques in Cryptology, pp. 99-126, Barcelone, Spain, 2005.

〔16〕 P. Rauzy and S. Guillry. "Countermeasure against high-order fault-injection attacks on CRT-RSA,"

〈저자소개〉

장 호 철 (Hocheol Jang) 정회원
2016년 2월: 호서대학교 정보보호학전공 학사
2018년 2월: 호서대학교 정보보호학과 석사
〈관심분야〉 정보보호, 암호학

오 수 현 (Soohyun Oh) 종신회원
1998년 2월: 성균관대학교 정보공학과 학사
2000년 2월: 성균관대학교 전기전자 및 컴퓨터공학부 석사
2003년 8월: 성균관대학교 전기전자 및 컴퓨터공학부 박사
2004년 3월~현재: 호서대학교 컴퓨터정보공학부 교수
2012년 1월~현재: 한국정보보호학회 이사
〈관심분야〉 암호학, 네트워크 보안, IoT 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 부교수
2007년 3월~현재: 호서대학교 컴퓨터정보공학부 교수
2013년 1월~현재: 한국정보보호학회 상임부회장
2009년 1월~현재: 한국산학기술학회 이사
〈관심분야〉 정보보호, 네트워크 보안, 부채널 공격