Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.34, NO.2, Apr. 2024 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2024.34.2.157

TiGER의 복호화 실패율 분석*

이 승 우,[1†] 김 종 현,[1] 박 종 환[2‡]
[1]고려대학교 (대학원생), [2]상명대학교 (교수)

Analysis on Decryption Failure Probability of TiGER*

Seungwoo Lee,[1†] Jonghyun Kim,[1] Jong Hwan Park[2‡]
[1]Korea University (Graduate student), [2]Sangmyung University (Professor)

요 약

LWE(learning with errors) 문제 기반의 공개키 암호는 기법 설계 및 파라미터 설정에 따라 복호화 실패율이 주어지는데, 높은 복호화 실패율은 실용성의 저하를 불러올뿐만 아니라 기법에 대한 공격으로 이어질 수 있음이 밝혀진 바 있다[1]. 따라서, KpqC 1차 라운드에 제안된 Ring-LWE 기반 KEM 기법인 TiGER[2]는 오류 보정 코드 (error correction code) Xef와 D2 인코딩 방법을 사용함으로써 복호화 실패율을 낮추고자 하였다. 그런데, Ring-LWE 문제에 기반한 암호화 기법 중 오류 보정 코드를 사용하는 기법의 경우 흔히 가정하는 각 비트 오류의 독립성이 성립하지 않음이 알려진 바 있다[3]. TiGER의 복호화 실패율 계산은 이를 고려하지 않은바, 본 논문에서는 오류 의존성을 고려하여 복호화 실패율을 다시 계산한다. 또한, TiGER(v2.0)의 비트 오류가 잘못 계산되었음을 발견하여 올바른 비트 오류 계산 식과 그에 따라 새로 계산한 복호화 실패율을 제시한다.

ABSTRACT

Probability of decryption failure of a public key cryptography based on LWE(learning with errors) is determined by its architecture and parameter settings. Since large decryption failure probability leads to attacks[1] on scheme as well as degradation of performance, TiGER[2], a Ring-LWE(R)-based KEM proposed for the first round of KpqC, tried to reduce the decryption failure probability by using error correction code Xef and D2 encoding method. However, D'Anvers et al. has shown that the commonly assumed independence of each bit error is not established since in the case of an encryption scheme based on Ring-LWE(R) using an error correction code, there is error dependency which is not negligible[3]. In this paper, since TiGER does not consider the error dependency, we calcualte the decryption failure probability of TiGER by considering the error dependency. In addition, we found that the bit error probability is incorrectly calculated in TiGER, so we present the correct calculation.

Keywords: Post-Quantum Cryptography, Ring-LWE(R), Decryption Failures, Error Correcting Codes

I. 서 론

기존의 인수분해 혹은 이산 대수 문제에 기반하는 공개키 암호가 양자 컴퓨팅에 취약함은 이미 알려져 있고, 따라서 양자 컴퓨터의 상용화가 다가옴에 따라 기존의 공개키 암호를 양자 컴퓨팅에도 안전한 양자 내성 암호로 대체하고자 하는 시도가 세계적으로 이어지고 있다. 제안된 기법 중 다수가 격자 기반 암호에 해당하는데, 격자 기반 암호의 중요한 성능 평가 요소 중 하나인 암호문의 크기는 크게 모듈러스의 크

Received(11. 24. 2023), Modified(01. 09. 2024), Accepted(02. 08. 2024)
* 본 연구는 고려대 암호기술 특화연구센터(UD210027XD)를 통한 방위사업청과 국방과학연구소의 연구비 지원으로 수행되었습니다.
† 주저자, kak5175@korea.ac.kr
‡ 교신저자, jhpark@smu.ac.kr(Corresponding author)

기와 암호문 압축의 정도 등에 따라 결정된다. 모듈러스를 작은 값으로 설정하거나 암호문을 많이 압축하여 암호문 크기를 줄일 수 있으나, 이 경우 복호화 실패율이 증가하게 된다. 복호화 실패율은 충분히 작지 않으면 키 복구 공격의 타겟이 될 수 있음이 알려져 있는 등[1] 안전성에 영향을 미칠 수 있으며, 정상적인 암호 통신에 문제를 발생시킬 수 있다. 이에 Kiltz 등은 복호화 실패율이 κ비트 안전성에 대하여 $2^{-\kappa}$보다 작거나 같아야 함을 제시한 바 있다[4]. 따라서, 둘 사이의 균형을 유지하는 것은 설계에 있어 중요한 요소이다.

한국 양자 내성 암호 공모전(KpqC) 1차 라운드에 제안된 기법 중 TiGER[2]는 격자 기반 KEM(Key Encapsulation Mechanism) 기법으로, Ring-LWE(R)에 기반하여 설계되었다. TiGER는 작은 모듈러스와 공개키 라운딩, 암호문 압축 등의 방식을 사용하여 설계되어 상기한 대로 암호문 크기가 작다는 장점을 갖지만 동시에 복호화 실패율이 충분히 작지 않다는 문제점이 있었고, 따라서 TiGER는 복호화 실패율을 줄이는 D2 인코딩 방식과 오류 보정 코드를 사용하여 해결하였다. 그런데, D'Anvers 등은 오류 보정 코드를 사용하는 경우 복호화 실패율 계산 시에 오류 의존성을 고려하지 않으면 복호화 실패율이 실제보다 낮게 계산될 수 있음을 보인 바 있고[3], 이승우 등[5]이 오류 의존성을 고려하여 TiGER의 복호화 실패율을 계산한 바 있다.

본 논문에서는 [5]의 방법을 베이즈 정리를 사용함으로써 오류 의존성을 반영하는 과정에서 근사를 줄여 보다 높은 정밀도를 갖는 식으로 TiGER의 복호화 실패율을 계산하는 방법과 계산 결과를 제시하고 오류 의존성을 고려하지 않은 계산과 비교한다. 또한, 본 연구진은 TiGER(v2.0)의 복호화 실패율이 D2 인코딩에 의한 영향이 고려되지 않은 채 잘못 계산되었음을 발견하였고, 올바르게 계산한 결과 복호화 실패율이 Kiltz 등이 제시한 요구 조건에 크게 못 미치는 것을 확인하였다. TiGER가 복호화 실패율을 요구 조건에 맞도록 조정하기 위해서는 전면적인 파라미터 수정이 필요할 것으로 보이며, 파라미터가 수정됨에 따라 암호문 크기가 증가할 것으로 예상하였다.

본 논문의 내용은 현재 TiGER(v3.0)에 반영되어, 복호화 실패율을 감소시키는 방향으로 파라미터가 수정되었으며 그에 따라 암호문의 크기가 증가하였다.

II. 배경지식

2.1 표기법

정수 집합 $\mathbb{Z}$에 대하여 $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$로 정의하고, 다항식 환 $R := \mathbb{Z}[X]/(X^n+1)$과 $R_q := \mathbb{Z}_q[X]/(X^n+1)$를 정의한다. 다항식 환의 원소는 소문자로, 벡터는 굵은 소문자로, 행렬은 굵은 대문자로 표기한다. 다항식 $a(x) \in R$에 대하여 x^i의 계수는 $(a)_i$로 표기하고, 다항식 $a(x) \in R$의 l_2-norm은 $\|a\|_2 = \sqrt{\sum_i (a)_i^2}$으로 정의한다. 각 성분이 다항식인 벡터 $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in R_q^n$의 l_2-norm은 $\|\mathbf{a}\|_2 = \sqrt{\sum_{i=0}^{n} \|a_i\|_2^2}$으로 정의한다. $a \leftarrow \chi(R_q)$는 $a \in R_q$의 각 계수가 χ의 분포에서 추출되었음을 의미하고, $\mathbf{a} \leftarrow \chi(R_q^k)$는 $\mathbf{a} \in R_q^k$의 각 성분의 각 계수가 χ에서 추출되었음을 의미한다. HWT_n^h는 $\{-1,0,1\}^n$에서 임의로 $n-h$개의 0과 h개의 1 또는 -1을 추출하는 분포이다. $\mathrm{Binom}(p,n,k) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}$으로 누적 이항 분포이다. $\lfloor \cdot \rceil : \mathbb{R} \to \mathbb{Z}$는 가장 가까운 정수를 반환하는 함수이다.

2.2 이산 확률 변수의 연산

이산 확률 변수 X와 $x \in X$ 대하여 확률질량함수를 다음과 같이 정의한다:

$$p_X(x) = \Pr[X = x].$$

또 다른 이산 확률 변수 Y에 대하여 이산 확률 변수 $X+Y$의 확률질량함수는 다음과 같이 합성곱(convolution)을 통해 구할 수 있음이 알려져 있다:

$$p_{X+Y}(x) = \sum_{y=-\infty}^{\infty} p_X(y) p_Y(x-y).$$

본 논문에서는 편의상 합성곱 연산을 $*$로 다음과 같이 표기한다:

$$p_X(x) * p_Y(x) := \sum_{y=-\infty}^{\infty} p_X(y) p_Y(x-y).$$

2개 이상의 확률 변수 $X_1, X_2, \ldots X_n$가 합성곱 연산되는 일반적인 경우, 다음과 같이 표기한다:

$$*_{i=1}^{n} p_{X_i}(x) := p_{X_1}(x) * p_{X_2}(x) * \cdots * p_{X_n}(x).$$

2.3 D2 인코딩

D2 인코딩은 Pöppelmann과 Güneysu가 제안한 인코딩 방식으로[6], Fig. 1.과 같이 메시지의 한 비트를 $\overline{m} \in R_q$의 두 계수에 인코딩한다. 디코딩은 Fig. 2.와 같이 오류가 포함되어 있는 두 계수를 합하여 최상위 비트를 출력하는 방식으로 이루어진다.

디코딩 과정에서 각 계수에 있는 오류가 함께 더해지므로, 최종적으로 한 비트의 메시지를 디코딩 할 때 오류는 두 계수에 들어있는 오류만큼 발생하지만, 오류의 임계값이 $q/4$에서 $q/2$로 증가하여 최종 복호화 실패율을 감소시키는 효과를 기대할 수 있다.

Algorithm D2_encoding($m = \{0,1\}^{n/2}$)

1. for i = 0 to $n/2-1$ do
2. for j = 0 to 1 do
3. $\overline{m}[2\mathrm{i}+\mathrm{j}] = m[i] \cdot \frac{q}{2}$
4. endfor
5. endfor
6. return $\overline{m}$

Fig. 1. D2 encoding algorithm

Algorithm D2_decoding($\widetilde{m} \in \mathbb{Z}_q^n$)

1. for i = 0 to $n/2-1$ do
2. if $\widetilde{m}[2i] + \widetilde{m}[2i+1] < q/2$ then
3. $m[i] = 0$
4. else
5. $m[i] = 1$
6. endif
7. endfor
8. return m

Fig. 2. D2 decoding algorithm

2.4 오류 의존성

[3]의 표기법을 따라 Fig. 3.에서 5.와 같이 일반적인 Ring-LWE/LWR 기반 암호화 기법을 쓸 수 있다. 본 논문에서는 p, k_1, k_2가 q를 나누는 수라고 가정한다.

이때 다음과 같이 라운딩 오류를 정의한다:

$$\mathbf{u_A} := \mathbf{A}\mathbf{s_A} + \mathbf{e_A} - \mathbf{b}_q,$$
$$\mathbf{u_B}' := \mathbf{A}\mathbf{s_B}' + \mathbf{e_B}' - \frac{q}{k_1}\mathbf{c_1},$$
$$u_B'' := \mathbf{b}_q \cdot \mathbf{s_B}' + e_B'' - \frac{k_2}{q}c_2.$$

Algorithm KeyGen()

1. $\mathbf{A} \leftarrow R_q^{l \times l}, \mathbf{s_A} \leftarrow \chi_{s_A}(R_q^l), \mathbf{e_A} \leftarrow \chi_{e_A}(R_q^l)$
2. $\mathbf{b} = \lfloor \frac{p}{q} \cdot (\mathbf{A}\mathbf{s_A} + \mathbf{e_A}) \rceil \in R_p^l$
3. **return** $(pk := (\mathbf{A}, \mathbf{b}), sk := \mathbf{s_A})$

Fig. 3. Key generation algorithm of a general Ring/Mod-LWE/LWR based encryption scheme

Algorithm Enc($pk = (\mathbf{A}, \mathbf{b}), m$)

1. $\mathbf{s_B}' \leftarrow \chi_{s_B'}(R_q^l), \mathbf{e_B}' \leftarrow \chi_{e'}(R_q^l), e_B'' \leftarrow \chi_{e'}(R_q)$
2. $\mathbf{b}_q \leftarrow \lfloor \frac{q}{p}\mathbf{b} \rceil$
3. $\mathbf{m_{ecc}} = \mathrm{ecc\_enc}(m)$
4. $\mathbf{c_1} = \lfloor \frac{k_1}{q}(\mathbf{A}\mathbf{s_B}' + \mathbf{e_B}') \rceil$
5. $c_2 = \lfloor \frac{k_2}{q}(\mathbf{b}_q \cdot \mathbf{s_B}' + \mathrm{e_B}'' + \frac{q}{2}m_{ecc}) \rceil$
6. **return** $(\mathbf{c_1}, c_2)$

Fig. 4. Encryption algorithm of a general Ring/Mod-LWE/LWR based encryption scheme

Algorithm Dec($sk = \mathbf{s_A}, \mathbf{c} = (\mathbf{c_1}, c_2)$)

1. $m_{ecc}' = \frac{q}{k_2}c_2 - (\frac{q}{k_1}\mathbf{c_1}) \cdot \mathbf{s_A}$
2. $m' = \mathrm{ecc\_dec}(m_{ecc}')$
3. **return** m'

Fig. 5. Decryption algorithm of a general Ring/Mod-LWE/LWR based encryption scheme

편의상

$$\mathbf{s}=\begin{pmatrix}-\mathbf{s_A}\\ \mathbf{e_A}+\mathbf{u_A}\end{pmatrix},\mathbf{c}=\begin{pmatrix}\mathbf{e_B}'+\mathbf{u_B}'\\ \mathbf{s_B}'\end{pmatrix},\ g=u_B''+e_B''$$

라고 정의하자. 복호화 실패율 DFP는 일반적으로 한 비트의 메시지를 한 다항식의 한 계수에 인코딩할 때 한 비트의 복호화 오류 확률 $p_b=\Pr[(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_i>q/4]$와 다항식의 차수 n에 대하여 각 비트에서 오류가 독립적으로 발생한다고 가정하고 $DFP=1-\mathrm{Binom}(p_b,n,f)$와 같이 계산한다. 이때, f는 오류 보정 코드가 보정할 수 있는 비트의 크기로, 오류 보정 코드를 사용하지 않는 경우 $f=0$이다.

그런데, D'Anvers 등[3]은 각 비트 간 복호화 오류 발생 확률이 실제로는 서로 의존적이고, 따라서 독립성 가정 하에서 계산한 복호화 실패율이 실제와 다를 수 있음을 지적하였다. 오류의 의존성은 다항식 환의 구조로부터 발생한다. Fig. 6.과 같이 각 원소를 균등하고 독립적으로 추출한 행렬 A와 벡터 v의 곱셈 결과는 서로 독립적이지만, 다항식 환의 두 원소의 곱은

$$\begin{aligned}xa(x)&=x(a_0+a_1x+a_2x^2+\cdots+a_{n-1}x^{n-1})\\&=\ a_0x+a_1x^2+a_2x^3+\cdots+a_{n-1}x^n\\&=-a_{n-1}+a_0x+a_1x^2+\cdots+a_{n-2}x^{n-1}\end{aligned}$$

와 같이 각 계수가 순환하는 방식으로 이루어지므로, $a,b\in R_q$의 각 계수를 균등하고 독립적으로 추출하여도 a의 원소의 순서만 순환하며 곱해지는 다항식의 곱셈 연산 구조로 인하여 곱셈 결과가 서로 의존적이다. 따라서, $\|a\|_2,\|b\|_2$가 크면 곱셈 결과의 norm인 $\|c\|_2$도 클 확률이 높아지는 것이다.

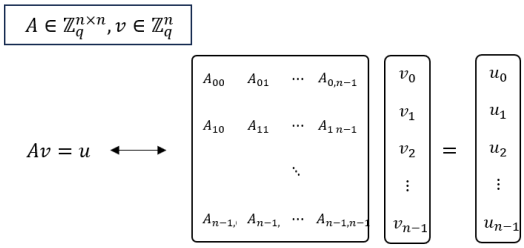

Fig. 6. Multiplication of matrix and vector over integer rings

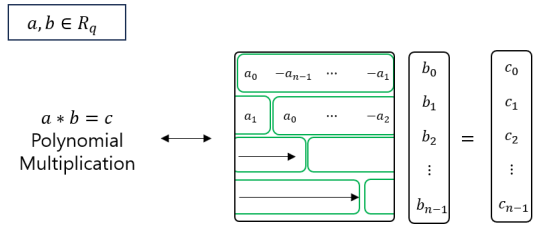

Fig. 7. Multiplication of polynomials in R_q

이러한 이유로, 오류 항 중 다항식 곱셈이 발생하는 항들인 $\mathbf{s},\mathbf{c}$의 norm이 크면 다항식 곱셈의 구조에 의해 다수의 비트에서 복호화 오류가 발생할 가능성이 높다. 즉, 한 비트의 오류 확률이 $\mathbf{s},\mathbf{c}$에 의존하는 것이다. 따라서 다음과 같이 비트 오류 확률 p_b를 $\|\mathbf{s}\|_2,\|\mathbf{c}\|_2$에 대한 조건부로 정의하여 이러한 의존성을 반영해야 한다:

$$p_b(\|\mathbf{s}\|_2,\|\mathbf{c}\|_2)=\Pr[|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_i|\geq q_t\mid\|\mathbf{s}\|_2,\|\mathbf{c}\|_2].$$

이에 따라 복호화 실패율도 다음과 같이 정의한다:

$$DFP=\sum_{\|\mathbf{s}\|_2,\|\mathbf{c}\|_2}(1-\mathrm{Binom}(p_b,n,f))\cdot\Pr[\|\mathbf{s}\|_2]\cdot\Pr[\|\mathbf{c}\|_2].$$

오류 보정 코드를 사용하지 않으면, 즉 $f=0$이면 복호화 실패 여부가 비트 오류의 개수와는 무관하게 비트 오류의 존재 여부로만 결정된다. 따라서, 다수의 비트 오류가 발생할 확률과 연관되어있는 위의 오류 의존성은 오류 보정 코드를 사용하지 않는 기법에서는 무시할 수 있다. 그러나, 1개 이상의 비트 오류를 허용하는 오류 보정 코드를 사용하는 기법에서는 오류 의존성을 고려하지 않으면 복호화 실패율을 실제보다 낮게 추정하게 될 수 있으므로, 이를 고려하여 복호화 실패율을 계산하는 것이 필요하다.

III. TiGER의 복호화 실패율

이번 장에서는 KpqC 1차 라운드에 진출한 격자 기반 KEM 기법인 TiGER(v2.0)의 복호화 실패율을 계산하는 방법과 계산 결과를 제시한다. 오류 독립성을 가정한 독립성 모델과 의존성을 고려한 의존성 모델에서 각각 복호화 실패율을 계산하고, 둘을 비교한다.

3.1 라운딩 오류의 분포 모델링

TiGER의 모듈러스는 모두 2의 지수 형태이므로, 본 논문에서는 비트 단위의 라운딩만 고려한다. d비트 만큼의 라운딩 오류의 분포가 양 끝 값을 제외한 값의 분포는 균등(uniform)하고, 양 끝 값의 확률은 나머지의 확률의 절반인 분포를 따른다고 가정한다. 이 분포는 실제 라운딩 오류의 분포와는 약간의 차이가 있으나, 그 차이가 크지 않으므로 계산의 편의를 위해 다음과 같이 모델링한다.

Table 1. Modeling of errors from rounding

u	-2^{d-1}	$-2^{d-1}+$	$\cdots$	$2^{d-1}-1$	2^{d-1}
Prob.	$\frac{1}{2^{d+1}}$	$\frac{1}{2^d}$	$\frac{1}{2^d}$	$\frac{1}{2^d}$	$\frac{1}{2^{d+1}}$

3.2 독립성 모델(independency model)

TiGER(v2.1)의 파라미터는 Table 2.와 같다. TiGER는 Ring-LWR에 기반하여 키를 생성하므로 $\mathbf{e_A}=0$이고, $\mathbf{s_A}, \mathbf{s_B}', \mathbf{e_B}', \mathbf{e_B}''$은 각각 $HWT_n^{h_{s_A}}$, $HWT_n^{h_{s_B'}}$, $HWT_n^{h_e}$, $HWT_n^{h_e}$에서 추출된다.

따라서

$$\mathbf{s}=\begin{pmatrix}-\mathbf{s_A}\\ \mathbf{u_A}\end{pmatrix}, \mathbf{c}=\begin{pmatrix}\mathbf{e_B}'+\mathbf{u_B}'\\ \mathbf{s_B}'\end{pmatrix}, g=u_B''+e_B''$$

라고 하자. TiGER에서는

$$p_b=1-\Pr[|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_i|<\frac{q}{2}]$$

와 같이 계산하였으나, D2 인코딩을 사용하므로 실제로는

Table 2. Parameter sets of TiGER

	l	n	q	p	k_1	k_2	h_{s_A}	$h_{s_B'}$	h_e	f
TiGER128	1	512	256	128	64	16	142	110	32	3
TiGER192	1	1024	256	128	64	4	132	132	32	5
TiGER256	1	1024	256	128	128	4	196	196	32	5

$$p_b=1-\Pr[|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_i|+|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_j|<\frac{q}{2}]$$

와 같이 계산해야 한다. 이때

$$1-\Pr[|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_i|+|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_j|<\frac{q}{2}]$$
$$=\Pr[|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_i|+|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_j|\geq\frac{q}{2}]$$

이고 삼각 부등식에 의해

$$|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_i|+|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_j|\geq|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_i+(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_j|$$

이므로, p_b는 다음의 부등식을 만족한다:

$$p_b\geq\Pr[|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_i+(\mathbf{s}^{\mathrm{T}}\mathbf{c}+g)_j|\geq\frac{q}{2}].$$

계산의 복잡성 때문에 본 논문에서는 p_b를 위의 하한으로 대체하여 고려한다. 이는 복호화 실패율이 실제보다 낮게 측정되는 결과로 이어져 TiGER의 안전성 수준을 실제보다 약간 높게, 즉 TiGER에 유리한 방향으로 평가하게 되지만, 본 논문에서는 그럼에도 TiGER의 복호화 실패율이 요구되는 수치를 크게 벗어남을 보인다.

$(-\mathbf{s_A}(\mathbf{e_B}'+\mathbf{u_B}'))_i$이 따르는 분포는 $\sum_{i=0}^{n-1}(-\mathbf{s_A})_i(\mathbf{e_B}'+\mathbf{u_B}')_i$의 분포인데, $\mathbf{s_A}$와 $(\mathbf{e_B}'+\mathbf{u_B}')_i$의 분포는 0을 기준으로 양수와 음수가 대칭이고 $\mathbf{s_A}$의 0이 아닌 원소는 -1혹은 1이므로 $(-\mathbf{s_A})_i(\mathbf{e_B}'+\mathbf{u_B}')_i$의 분포는 $\mathbf{s_A}$가 0이 아닐 때, $(\mathbf{e_B}'+\mathbf{u_B}')_i$의 분포와 동일하다. 또 $\mathbf{s_A}$의 0이 아닌 원소의 개수는 h_s개로 고정되어 있으므로, 결론적으로 $\sum_{i=0}^{n-1}(-\mathbf{s_A})_i(\mathbf{e_B}'+\mathbf{u_B}')_i$의 분포는 $\sum_{i=1}^{h_s}(\mathbf{e_B}'+\mathbf{u_B}')_i$의 분포와 동일하며 확률질량함수는

$$(*_{i=1}^{h_s}p_{e_B'}(x))*(*_{i=1}^{h_s}p_{u_B'}(x))$$

로 주어진다. 이때 e_B'은 -1과 1이 각각 $h_e/2n$의

확률을 갖고, 0은 $(n-h_e)/n$의 확률을 갖는 확률변수이고, u_B'은 Table 1.의 $d=q/p$인 경우의 확률변수이다. 마찬가지의 방식으로 $(\mathbf{u_A}\cdot\mathbf{s_B}')_i$의 분포는 $*_{i=1}^{h_s} p_{u_A}(x)$에 해당한다. 마지막으로, $\mathbf{g}_i=(\mathbf{e_B}''+\mathbf{u_B}'')_i$의 분포는 $p_{e_B''}(x)*p_{u_B''}(x)$에 해당한다. $|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+\mathbf{g})_i|$의 확률질량함수는

$$(*_{i=1}^{h_s} p_{e_B'}(x))*(*_{i=1}^{h_s} p_{u_B'}(x))*(*_{i=1}^{h_s} p_{u_A}(x))*p_{e_B''}(x)*p_{u_B''}$$

에 해당하는 확률 분포이므로, 최종적으로 $|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+\mathbf{g})_i|+|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+\mathbf{g})_j|$는 두 다항식의 지수 절댓값의 합이 $q/2$보다 크거나 같은 모든 경우의 수에 대하여 해당하는 두 다항식의 계수의 곱을 합하면 비트 오류 확률 p_b를 구할 수 있다. 복호화 실패율은 $DFP=1-\mathrm{Binom}(p_b,n,f)$와 같이 주어진다.

Table 3.은 TiGER에서 제시된 복호화 실패율과 본 논문에서 계산한 값을 비교한 표이다. 상기한 것처럼 TiGER 연구진이 D2 인코딩에 의해 오류 항이 두 번 더해지는 것을 고려하지 않는 오류를 범하였고, 본 계산에서는 고려하여 계산한바 복호화 실패율이 큰 차이를 보인다.

올바르게 계산된 복호화 실패율은 모든 파라미터에서 2^{-100}에도 크게 미치지 못하므로 복호화 실패를 이용한 비밀키 복구 공격[1] 등의 많은 공격에 노출되어 있다. 따라서 TiGER는 복호화 실패율을 적어도 2^{-128}이하 수준으로 낮추기 위해 모듈러스를 증가시키거나 암호문 압축의 정도를 줄이는 등의 파라미터 수정이 반드시 필요하다. 이와 같은 파라미터 수정이 이루어질 경우, TiGER는 암호문 크기가 현재보다 증가하는 등의 성능 하락이 있을 것으로 예상된다.

D2 인코딩에 의한 오류 항이 두 번 더해진 것 대비 하나의 항만 고려한 경우의 차이가 큰 차이를 보

Table 3. Decryption failure probability of TiGER

Parameter sets	DFP ($\log_2$)	
	TiGER spec.	Our results
TiGER128	-145.75	-63.01
TiGER192	-150.41	-38.64
TiGER256	-201.29	-54.40

Table 4. Bit error probability of TiGER

Parameter sets	p_b ($\log_2$)	
	1 error term	2 error terms
TiGER128	-44.98	-23.60
TiGER192	-33.36	-14.85
TiGER256	-41.90	-17.48

이는 것은 오류 보정 코드 때문으로 이해할 수 있다. 이는 두 경우 비트 오류 확률의 차이와 오류 보정 코드를 사용할 때 복호화 실패율을 계산하는 공식으로부터 다음과 같이 이해할 수 있다. 우선 오류 항이 두 개인 경우와 한 개인 경우 비트 오류 확률을 계산하여 비교하면 Table 4.과 같다.

오류 항이 한 개에서 두 개가 되면 오류 분포의 표준편차는 증가한다. 그러나, 표준편차가 상승하는 것에 비하여 오류가 발생하기 시작하는 임계값이 $q/4$에서 $q/2$로 증가함으로써 증가한 표준편차에 비해 임계값이 더 달아나게 되고 따라서 최종 비트 오류 확률(그리고 이어서 복호화 실패율도)은 오히려 감소하게 되는 것이다.

직관적 이해를 돕기 위하여 오류가 가우시안 분포를 따른다고 가정하고(실제로는 합성곱 연산을 통해 직접 확률 분포를 구한다), 간단히 근사하여 비트 오류 확률을 계산하면 다음과 같다. 두 개이어야 하는 오류 항을 한 개만 고려한 이 경우, TiGER128 파라미터에서 한 개의 오류 항만 고려한 오류 분포의 표준편차는 약 17.3인 반면 두 개의 오류 항을 고려한 경우 표준편차는 약 24.4이다. 확률변수의 덧셈은 합성곱의 형태로 연산되는데, 합성곱 연산의 특성상 오류 항이 두 개로 늘어나도 표준편차가 두 배가 되지는 않지만, 항이 늘어났으므로 표준편차가 증가한다. 오류의 임계값인 $q/2=128$은 표준편차의 순서대로 7.41배, 5.24배이다. 오류 분포에서 오류의 절대값이 임계값 $q/2=128$보다 큰, tail의 확률을 따지면 tail의 확률이 각각 $1-\mathrm{erf}(7.41/\sqrt{2})\approx-44.84$, $1-\mathrm{erf}(5.24/\sqrt{2})\approx-22.57$으로 Table 4.의 결과를 직관적으로 이해할 수 있다.

또한 오류 보정 코드를 사용하는 경우 복호화 실패율 공식

$$DFP=1-\mathrm{Binom}(p_b,n,f)=1-\sum_{i=0}^{f}\binom{n}{i}p_b^i(1-p_b)^{n-i}$$

로부터 복호화 실패율 *DFP*는 비트 오류 확률 p_b에 거듭제곱이 된 형태로 의존한다는 점에서, 비트 오류 확률이 증폭되고, 따라서 Table 3.와 같이 큰 격차가 벌어지는 것이다.

3.3 의존성 모델(dependency model)

비밀 키 $sk=\mathbf{s_A}$와 임시 키(ephemeral key) $\mathbf{s_B}'$, 오류 항 $\mathbf{e_A},\mathbf{e_B}',\mathbf{e_B}''$의 각 계수가 $\{-1,0,1\}$ 상의 고정 Hamming weight을 갖는 분포에서 추출되었고, 라운딩이 없어 $\mathbf{u_A},\mathbf{u_B}',\mathbf{u_B}''$가 모두 0인 경우 $\|\mathbf{s}\|_2,\|\mathbf{c}\|_2$가 항상 일정하므로 D'Anvers 등의 모델에서는 오류 의존성이 드러나지 않는다. 그러나, $\mathbf{u_A},\mathbf{u_B}',\mathbf{u_B}''$가 적어도 하나 0이 아닌 경우 오류 의존성의 영향을 받게 된다. TiGER는 이 경우에 해당하므로, 오류 의존성에 영향을 무시할 수 없다. 이 절에서는 라운딩 오류에서 기인한 오류 의존성을 고려하여 TiGER 복호화 실패율을 계산한다.

TiGER는 비밀 값과 LWE 오류를 $\{-1,0,1\}$ 상의 고정 Hamming weight 분포에서 추출하고, 오류 의존성은 R_q에서의 덧셈에서는 발생하지 않는다. 따라서 $-\mathbf{s_A},\mathbf{s_B}',\mathbf{e_B}',\mathbf{e_B}'',\mathbf{u_B}''$에 의한 오류 의존성은 무시할 수 있고, $\mathbf{u_A},\mathbf{u_B}'$의 의한 오류 의존성만을 고려한다. 즉, 구하고자 하는 비트 오류 확률은

$$p_b=\Pr[|(\mathbf{s}^{\mathrm{T}}\mathbf{c}+\mathbf{g})_i+(\mathbf{s}^{\mathrm{T}}\mathbf{c}+\mathbf{g})_j|\geq\frac{q}{2}\,|\,\|\mathbf{u_A}\|_2^2,\|\mathbf{u_B}'\|_2^2]$$

이고, 앞선 독립성 모델과의 유일한 차이는 $\mathbf{u_A},\mathbf{u_B}'$의 분포가 $\|\mathbf{u_A}\|_2^2,\|\mathbf{u_B}'\|_2^2$에 대한 조건부 확률 분포가 된다는 것이다.

$\mathbf{u}\in R_q$에 대하여 $\|\mathbf{u}\|_2^2$가 주어졌을 때 $(\mathbf{u})_i=a$일 확률 $\Pr[(\mathbf{u})_i=a\,|\,\|\mathbf{u}\|_2^2]$을 $(\mathbf{u})_i$가 될 수 있는 모든 값 a에 대하여 구함으로써 $\|\mathbf{u}\|_2^2$에 대한 $\mathbf{u}$의 조건부 확률 분포를 구할 수 있다. 베이즈 정리(Bayes' theorem)를 이용하여 $\Pr[(\mathbf{u})_i=a\,|\,\|\mathbf{u}\|_2^2]$를 다음과 같이 쓸 수 있다:

$$\begin{aligned}\Pr[(\mathbf{u})_i&=a\,|\,\|\mathbf{u}\|_2^2]\\&=\frac{\Pr[\,\|\mathbf{u}\|_2^2\,|\,(\mathbf{u})_i=a]\cdot\Pr[(\mathbf{u})_i=a]}{\Pr[\,\|\mathbf{u}\|_2^2\,]}.\end{aligned}$$

$(\mathbf{u})_i$의 확률 분포는 주어져 있고, $\|\mathbf{u}\|_2^2=\mathrm{u}_0^2+\mathrm{u}_1^2+\mathrm{u}_2^2+\cdots+u_{n-1}^2$이므로 $\|\mathbf{u}\|_2^2$의 확률질량함수는

$$*_{i=0}^{n-1}p_{(\mathbf{u})_i^2}(x)$$

와 같이 구할 수 있다. 또한, $(\mathbf{u})_i=a$일 때 $\|\mathbf{u}\|_2^2=\mathrm{u}_0^2+\mathrm{u}_1^2+\cdots+\mathrm{u}_{i-1}^2+\mathrm{a}^2+\cdots+\mathrm{u}_{n-1}^2$ 이므로, $\|\mathbf{u}\|_2^2\,|\,(\mathbf{u})_i=a$의 확률질량함수는

$$*_{i=0}^{n-1}p_{(\mathbf{u})_i^2}(x)+a^2$$

와 같이 구할 수 있다.

$(\mathbf{s}^{\mathrm{T}}\mathbf{c}+\mathbf{g})_i=(-\mathbf{s_A}(\mathbf{e_B}''+\mathbf{u_B}'')+\mathbf{u_A}\mathbf{s_B}')_i$의 분포는 다음과 같이 구한다. 3.2에서 서술한 바와 같이 $\sum_{i=0}^{n-1}(-\mathbf{s_A})_i(\mathbf{e_B}'+\mathbf{u_B}')_i$의 분포는 $\sum_{i=1}^{h_s}(\mathbf{e_B}'+\mathbf{u_B}')_i$의 분포와 동일하다. 이 값은 독립적으로 생성된 확률 분포들의 합이므로, 아래의 중심 극한 정리를 이용하여 분산이

$$h_s\cdot var((\mathbf{e_B}')_i)+h_s\cdot var((\mathbf{u_B}')_i)$$

이고 대칭성에 의해 평균은 0인 정규분포로 근사할 수 있다. 마찬가지로 $(\mathbf{u_A}\mathbf{s_B}')_i$도 분산이

$$h_r\cdot var((\mathbf{u_A})_i)$$

이고 평균이 0인 정규 분포로 근사할 수 있고, 두 가우시안 분포의 합은 분산이 각 분포의 분산의 합인 가우시안 분포이므로

$$(\mathbf{s}^{\mathrm{T}}\mathbf{c})_i\sim N(0,h_s\sigma_{\mathbf{u_B}'}^2+h_s\sigma_{\mathbf{e_B}'}^2+h_r\sigma_{\mathbf{u_A}}^2)$$

이다. D2 인코딩을 사용하였으므로, 최종적인 가우시안의 분산에 두 배를 해주어야 하고 $\mathbf{g}$의 분포 또한 두 번 더해준 값의 분포를 따르게 조정해주어야 한다:

$$\begin{aligned}(\mathbf{s}^{\mathrm{T}}\mathbf{c})_i+(\mathbf{s}^{\mathrm{T}}\mathbf{c})_j&\sim N(0,2(h_s\sigma_{\mathbf{u_B}'}^2+h_s\sigma_{\mathbf{e_B}'}^2+h_r\sigma_{\mathbf{u_A}}^2)),\\(\mathbf{g})_i+(\mathbf{g})_j&\sim(poly_{\mathbf{g}}(x))^2.\end{aligned}$$

오류 의존성을 고려한 비트 오류 확률 p_b는 다음과 같이 주어진다:

$$p_b = \sum_g \Pr[x > \frac{q}{2} - g \text{ or } x < -\frac{q}{2} - g \mid x \leftarrow N(0, 2(h_s\sigma^2_{\mathbf{u_B}'} + h_s\sigma^2_{\mathbf{e_B}'} + h_r\sigma^2_{\mathbf{u_A}}))]\Pr[g].$$

이로부터 복호화 실패율 DFP는 다음과 같이 구한다:

$$DFP = \sum_{\|\mathbf{u_A}\|_2, \|\mathbf{u_B}'\|_2} (1 - \mathrm{Binom}(p_b, n, f)) \cdot \Pr[\|\mathbf{u_A}\|_2] \cdot \Pr[\|\mathbf{u_B}'\|_2].$$

Table 5.는 3.2에서 오류 독립성을 가정하고 계산한 TiGER의 복호화 실패율과 위 식으로부터 오류 의존성을 가정하고 계산한 값들을 비교한 표이다. 오류 의존성이 무시할 수 없을 만큼 작용하여 오류 의존성을 고려할 때 복호화 실패율이 약 2^7 가량 상승하는 것을 확인할 수 있다.

Table 5. Compraison between independency model and dependency model on decryption failure probability of TiGER

Parameter sets	DFP with Xef($\log_2$)	
	Our results (Indep. model)	Our results (Dep. model)
TiGER128	-64.32	-57.05
TiGER192	-39.97	-32.77
TiGER256	-56.08	-48.59

IV. 복호화 실패율에 의한 TiGER 변경 사항

본 논문의 내용에 의하여 최신 버전의 TiGER(v3.0)는 다음 표와 같이 암호문 압축에 관여하는 파라미터를 수정함으로써 복호화 실패율을 낮추었다.

TiGER128의 경우 k_1이 64에서 128로 증가하여 암호문 c_1의 압축을 2비트에서 1비트로 줄였고, k_2는 16에서 8로 감소하여 암호문 c_2의 압축은 4비트에서 5비트로 증가하였다. 결과적으로 TiGER128의 암호문 크기는 그대로 유지되었다. 그러나 TiGER192와 TiGER256의 경우 c_1과 c_2의 압축을 각각 1비트씩 줄임으로써 복호화 실패율을 감소시키고자 했고, 이와 동시에 암호문 크기의 증가로도 이어졌다.

Table 6. Parameter sets of TiGER v3.0

	l	n	q	p	k_1	k_2	h_{s_A}	$h_{s_B'}$	h_e	f
TiGER128	1	512	256	128	128	8	104	104	32	3
TiGER192	1	1024	256	128	128	8	116	116	32	5
TiGER256	1	1024	256	128	256	8	184	184	32	5

파라미터 수정 전과 후의 암호문의 크기는 Table 7.와 같다. 공개키의 크기는 수정 전과 후가 동일하다.

추가로, TiGERv3.0의 파라미터별 복호화 실패율 검증 결과는 독립성 모델, 의존성 모델 순서로 Table 8., 9.과 같다.

독립성 모델에서 비트 오류 확률은 오차가 모두 2^{-1}미만으로, 오류 보정 코드가 적용되어 오차가 증폭된 복호화 실패율에서는 약간의 오차를 보이지만 충분히 검증되었다고 볼 수 있다.

의존성 모델에서는 복호화 실패율이 최대 약 2^{13} 가량 증가했다. 의존성을 고려할 때와 그렇지 않을 때가 여전히 유의미한 차이를 보이므로, 복호화 실패율을 계산하는 데에 있어 의존성을 고려할 필요성이 있다.

Table 7. Ciphertext size of TiGER v3.0 (Bytes)

Parameter sets	ct (v2.0)	ct (v3.0)
TiGER128	640	640
TiGER192	1,024	1,280
TiGER256	1,152	1,408

Table 8. Decryption failure probability of TiGERv3.0 in independency model

Par. sets	p_b ($\log_2$)		DFP ($\log_2$)	
	TiGER spec.	Our results	TiGER spec.	Our results
128	-41.06	-41.72	-132.86	-135.47
192	-39.60	-39.16	-187.10	-184.47
256	-35.62	-34.90	-163.21	-158.89

Table 9. Decryption failure probability of TiGERv3.0 in dependency model

Parameter sets	DFP ($\log_2$)	
	TiGER spec.	Our results
TiGER128	-132.86	-127.26
TiGER192	-187.10	-174.15
TiGER256	-163.21	-150.36

V. 결 론

본 논문에서는 Ring-LWE 기반 암호화 기법의 각 비트의 오류가 독립적이라고 가정하는 독립성 모델과, 각 비트의 오류가 의존적이라고 가정하는 의존성 모델 하에서 각각 TiGER의 복호화 실패율을 계산하고 둘을 비교하였다. 예상한 바와 같이, 의존성 모델에서 복호화 실패율이 독립성 모델로 구한 복호화 실패율에 비하여 상승하는 결과를 확인할 수 있었다. 또한, 기존 TiGER의 복호화 실패율 계산에 있던 오류를 정정하여 올바른 계산값을 구하였는데, 계산 결과 TiGER의 복호화 실패율이 기존의 분석보다 큰 폭으로 상승하여 것을 확인할 수 있었다. 본 논문에서 계산된 복호화 실패율은 각 파라미터의 안전성 수준 κ에 대해서 Kiltz 등이 제시한[4] 요구조건인 $2^{-\kappa}$에 크게 미치지 못하므로, 안전성 및 기법의 정상적인 동작을 위하여 복호화 실패율을 낮추는 방향으로의 기법 및 파라미터 수정이 필요할 것으로 예상하였다. 현재는 위 내용을 TiGER 연구진에서 받아들여 최신 버전의 TiGER(v3.0)에서는 Table 6.과 같이 파라미터가 수정되었고, Table 7.과 같이 암호문 크기가 증가하게 되는 결과로 이어졌다.

References

[1] JP. D'Anvers, Q. Guo, T. Johansson, A. Nilsson, F. Vercauteren, and I. Verbauwhede, "Decryption failure attacks on IND-CCA secure lattice-based schemes," Public-Key Cryptography, PKC 2019, LNCS 11443, pp. 565-598, 2019.

[2] Seunghwan Park, Chi-Gon Jung, Aesun Park, Joongeun Choi, and Honggoo Kang, "TiGER: Tiny bandwidth key encapsulation mechanism for easy miGration based on RLWE(R)," KpqC Competition Algorithms, https://www.kpqc.or.kr/competition.html, (7/9, 2023).

[3] JP. D'Anvers, F. Vercauteren, and I. Verbauwhede, "The impact of error dependencies on Ring/Mod-LWE/LWR based schemes," Post-Quantum Cryptography, PQCrypto 2019, LNCS 11505, pp. 103-115, 2019.

[4] D. Hofheinz, K. Hövelmanns, and E. Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation," Theory of Cryptography, TCC 2017, LNCS 10677, pp. 341-371, 2017.

[5] Seungwoo Lee, Jonghyun Kim, and Jong Hwan Park, "Analysis of decryption failure rate of TiGER considering error dependencies," KIMST 2023, pp. 1032-1033, June 2023.

[6] T. Pöppelmann and T. Güneysu, "Towards practical lattice-based public-key encryption on reconfigurable hardware," Selected Areas in Cryptography, SAC 2013, LNCS 8282, pp. 68-85, 2013.

〈저자소개〉

이 승 우 (Seungwoo Lee) 학생회원
2023년 2월: 고려대학교 물리학과/수학과 졸업
2023년 3월~현재: 고려대학교 정보보호학과 석사과정
〈관심분야〉 암호 프로토콜, 양자내성암호

김 종 현 (Jonghyun Kim) 학생회원
2014년 2월: 성균관대학교 수학과 졸업
2014년 3월~현재: 고려대학교 정보보호학과 석박사 통합과정
〈관심분야〉 암호 프로토콜, 양자내성암호

박 종 환 (Jong Hwan Park) 정회원
1999년 2월: 고려대학교 수학과 졸업
2005년 2월: 고려대학교 정보보호학과 석사
2008년 8월: 고려대학교 정보보호학과 박사
2013년 9월~2019년 8월: 상명대학교 컴퓨터과학과 조교수
2019년 9월~현재: 상명대학교 컴퓨터과학과 부교수
〈관심분야〉 함수 암호, 브로드캐스트 암호, 암호 프로토콜