논문 20-45-06-08 The Journal of Korean Institute of Communications and Information Sciences '20-06 Vol.45 No.06
https://doi.org/10.7840/kics.2020.45.6.998

# 그레이 부호 및 오류정정부호를 사용하여 256-비트 암호시스템 FrodoKEM을 개선하는 기법

이 은 상◆, 노 종 선*

# Method for Improving the 256-Bit Cryptosystem FrodoKEM Using Gray and Error-Correcting Codes

Eunsang Lee◆, Jong-Seon No*

요 약

격자 기반암호는 포스트 양자 암호(post-quantum cryptography, PQC)에서 가장 유망한 분야 중 하나이다. FrodoKEM은 가장 대표적인 격자 기반암호 중 하나이며 learning with errors (LWE) 문제에 기반을 둔다. 성능을 높이기 위해 오류정정부호를 사용하는 다양한 격자 기반암호 시스템들이 있다. 그러나 FrodoKEM과 같이 링(ring) 구조를 사용하지 않는 격자 기반암호 시스템에서는 데이터를 실을 수 있는 심볼의 개수가 적기 때문에 오류정정부호를 사용하기 힘들다. 따라서 대부분의 기존 논문들은 링 구조를 사용하는 시스템에만 오류정정부호를 사용하였다. 최근 FrodoKEM에 오류정정부호를 적용한 사례가 있는데 128-비트 시스템인 Frodo-640 및 192-비트 시스템인 Frodo-976에만 오류정정부호를 사용하였고 256-비트 시스템인 Frodo-1344에 오류정정부호를 사용하지는 못했다. 본 논문은 그레이 부호(Gray code)를 사용하고 암호시스템 앞뒤에 오류정정부호 인코딩 및 디코딩을 추가함으로써 Frodo-1344를 처음으로 변형시켰다. 또한 변형된 암호시스템의 보안성 수준(security level)을 계산하고 Frodo-1344와 비교하였는데 그 결과 Frodo-1344의 보안성 수준이 13비트 향상하였다.

**키워드 :** 포스트 양자 암호, 키-캡슐화 메커니즘, 그레이 부호, FrodoKEM, 오류정정부호
**Key Words :** Post-quantum cryptography, key-encapsulation mechanism, Gray code, FrodoKEM, error-correcting codes

ABSTRACT

Lattice-based cryptography is one of the most promising areas in post-quantum cryptography (PQC). FrodoKEM is one of the most representative lattice-based cryptosystems and it is based on the learning with errors (LWE) problem. There are some lattice-based cryptosystems that use error-correcting codes to improve the performance. However, it is difficult to use error-correcting codes in the lattice-based cryptosystems such as FrodoKEM which do not use ring structure, because the number of symbols to which the data is mapped is small. Therefore, most previous papers apply error-correcting codes only to systems which use ring structures. Recently, error-correcting codes have been used to FrodoKEM. Error-correcting codes were used only to 128-bit system Frodo-640 and 192-bit system Frodo-976, and not used to 256-bit system Frodo-1344. This paper first modified Frodo-1344 by using Gray code and adding encoding and decoding of error-correcting codes before and after the cryptosystem. As a result, the security level of Frodo-1344 was improved by 13 bits.

※ 본 연구는 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행되었습니다.
◆ First Author : Department of Electrical and Computer Engineering, INMC, Seoul National University, eslee3209@ccl.snu.ac.kr, 학생(박사), 학생회원
* Department of Electrical and Computer Engineering, INMC, Seoul National University, jsno@snu.ac.kr, 정교수, 종신회원
논문번호 : 2020-027-B-RE, Received February 11, 2020; Revised March 29, 2020; Accepted April 3, 2020

## Ⅰ. 서 론

RSA (Rivest Shamir Adleman) 혹은 타원곡선암호와 같은 기존 공개키 암호 시스템들은 향후 성능이 향상된 양자컴퓨터로 인해 무력화될 수 있다. 따라서 양자 컴퓨터에도 안전한 포스트 양자 암호(post-quantum cryptography, PQC)를 개발하는 것이 중요하다. 현재 대표적인 미국 표준화기관인 NIST (National Institute of Standards and Technology)는 PQC 알고리즘들을 제안받고 평가하며 표준화하는 작업을 진행하고 있다. 현재 26개의 알고리즘이 NIST PQC 표준화 라운드 2에 선정되었고 평가를 받는 중이다. 그중 격자 기반암호(lattice-based cryptography)는 가장 유망한 PQC 분야이다[1,2].

Learning with errors (LWE)는 2005년에 Regev[3]가 처음 제안한 문제이며 격자에 관련된 최악의 경우 문제(worst-case problem)보다 어렵다는 것이 증명되었다. Ring-LWE[4] 문제는 링(ring) 상에서의 LWE 문제이며 아이디얼(ideal) 격자 상의 최악의 경우 문제보다 어렵다는 것이 증명되었다. Ring-LWE 문제는 LWE 문제 기반 암호 시스템의 키 크기를 상당히 감소시킨다. 많은 격자 기반 공개키 암호화 스킴(public-key encryption, PKE) 및 키-캡슐화 메커니즘(key-encapsulation mechanism, KEM)은 LWE 및 Ring-LWE 문제의 어려움에 기반을 둔다.

NIST PQC 표준화 라운드 2에 선정된 격자 기반암호 알고리즘 중에는 오류정정부호를 사용하는 것이 많다. NewHope[5]는 덧셈 턱 인코딩(additive threshold encoding, ATE)이라는 간단한 오류정정 기법을 사용한다. Round5[6]는 XE5라는 부채널공격에 저항성 있는 오류정정부호를 사용한다. LAC[7] 및 ThreeBears[8]는 BCH (Bose Chaudhuri Hocquenghem) 부호를 사용한다. 그러나 오류정정부호를 사용하는 이들 알고리즘은 모두 링 기반 스킴이다.

최근에 링을 사용하지 않는 스킴인 FrodoKEM[9]에 오류정정부호를 적용한 사례가 있다[10,11]. 이 논문은 Frodo-640 및 Frodo-976에 그레이 부호(Gray code) 및 오류정정부호를 사용하여 보안성 수준(security level)을 높이거나 메시지 크기를 증가시키거나 데이터 전송량(bandwidth)을 감소시킨다. FrodoKEM은 NIST PQC 표준화 라운드 2에 선정된 가장 대표적인 PQC 스킴 중 하나이며 향후 표준이 될 가능성이 크다. 따라서 미래의 향상된 암호 공격에 대한 대비를 위해 이 스킴의 보안성 수준을 향상하는 것은 상당히 의미 있는 일이다.

그러나 기존 논문에서 256-비트 시스템인 Frodo-1344에는 오류정정부호를 적용하지 못했다. 본 논문에서는 처음으로 기존 맵핑(mapping) 대신 그레이 부호를 사용하고 암호시스템 앞뒤에 오류정정부호 부호기 및 복호기를 추가함으로써 Frodo-1344을 변형시켰다. 그 결과 Frodo-1344의 보안성 수준이 13비트 향상되었다.

NewHope과 같은 링 기반 스킴은 데이터를 실을 수 있는 심볼 개수가 많아서 오류정정부호를 사용하기가 수월하다. 그러나 FrodoKEM과 같이 링을 사용하지 않는 스킴은 심볼 개수가 매우 적기 때문에 한 심볼에 대응되는 비트 수를 늘려야 하며 이는 실패확률(decrption failure rate, DFR)을 증가시킨다. 따라서 FrodoKEM에 단순히 오류정정부호를 적용하는 것만으로는 성능이 향상된다는 보장을 할 수 없다. 본 논문은 Frodo-1344에 적합한 BCH 부호와 그레이 부호를 잘 설계하고 사용함으로써 Frodo-1344의 성능을 높일 수 있었다.

## Ⅱ. FrodoKEM에 오류정정부호를 사용하는 기존 방법

본 장에서는 FrodoKEM에 오류정정부호를 사용하는 기존 방법에 대하여 소개한다.

### 2.1 FrodoPKE 알고리즘 요약

본 절에서는 FrodoKEM의 기초가 되는 FrodoPKE 스킴에 관해서 설명한다. 본 논문은 FrodoKEM 대신 FrodoPKE에 오류정정부호를 적용하여 실패확률을 계산한다. FrodoKEM은 단순히 FrodoPKE를 적응 선택 암호문 공격에 안전(indistinguishability under adaptive chosen ciphertext attack, IND-CCA2)하도록 변형한 것이며 FrodoKEM의 실패확률은 FrodoPKE의 실패확률과 같다. FrodoPKE 알고리즘은 다음 파라미터들을 갖고 있다.

$\sigma$; 에러분포의 표준편차
$\chi$; $\mathbb{Z}$상의 확률 분포
$q$; 2의 거듭제곱 모듈러스(modulus)
$\overline{m},\overline{n},n$; 정수 행렬 차원
$B$; 각 심볼에 해당하는 비트 수. 여기서 비트 수는 오류정정부호가 사용되면 코드워드(codeword) 비트 수, 오류정정부호가 사용되지 않으면 메시지 비트 수

FrodoPKE를 구성하는 알고리즘은 다음과 같다.

Input: None
Output: Key pair $(pk, sk) \in (\{0,1\}^{len_A} \times \mathbb{Z}_q^{n \times \bar{n}}) \times \mathbb{Z}_q^{n \times \bar{n}}$

1. Generate pseudorandom matrix $A \in \mathbb{Z}_q^{n \times n}$ with $seed_A$
2. Generate matrices $S, E \in \mathbb{Z}_q^{n \times n}$ from $seed_E$
3. Compute $B = AS + E$
4. Return public key $pk \leftarrow (seed_A, B)$ and secret key $sk \leftarrow S$

알고리즘 1. FrodoPKE 키 생성 알고리즘[9]
Algorithm 1. FrodoPKE key generation algorithm

Input: Message $\mu \in \{0,1\}^{\bar{m}\bar{n}B}$ and public key $pk = (seed_A, B) \in \{0,1\}^{len_A} \times \mathbb{Z}_q^{n \times \bar{n}}$
Output: Ciphertext $c = (C_1, C_2) \in \mathbb{Z}_q^{\bar{m} \times n} \times \mathbb{Z}_q^{\bar{m} \times \bar{n}}$

1. Generate pseudorandom matrix $A \in \mathbb{Z}_q^{n \times n}$ with $seed_A$
2. Generate matrices $S', E' \in \mathbb{Z}_q^{\bar{m} \times n}$ and $E'' \in \mathbb{Z}_q^{\bar{m} \times \bar{n}}$ with $seed_E$
3. Compute $B' = S'A + E'$ and $V = S'B + E''$
4. Return ciphertext $c = (C_1, C_2) = (B', V + Frodo.Encode(\mu))$

알고리즘 2. FrodoPKE 암호화 알고리즘[9]
Algorithm 2. FrodoPKE encryption algorithm

Input: $C_1, C_2, S$
Output: $\mu'$

1. Compute
$M = C_2 - C_1 S = V + Frodo.Encode(\mu) - (S'A + E')S = Frodo.Encode(\mu) + S'E + E'' - E'S = Frodo.Encode(\mu) + E'''$
2. Return $\mu' \leftarrow Frodo.Decode(M)$

알고리즘 3. FrodoPKE 해독 알고리즘[9]
Algorithm 3. FrodoPKE decryption algorithm

알고리즘 2 및 알고리즘 3에서 사용되는 $Frodo.Encode$ 함수는 아래와 같은 맵핑이다. 각각 $B = 2,3,4$에 해당되고 Frodo-640, Frodo-976, Frodo-1344에서 사용되며 $B$-비트 메시지가 다음 규칙에 따라 심볼에 맵핑된다.

$$00 \rightarrow 0, 01 \rightarrow \frac{q}{4}, 10 \rightarrow \frac{2q}{4}, 11 \rightarrow \frac{3q}{4} \tag{1}$$

$$\begin{aligned} &000 \rightarrow 0,\ 001 \rightarrow \frac{q}{8},\ 010 \rightarrow \frac{2q}{8},\ 011 \rightarrow \frac{3q}{8} \\ &100 \rightarrow \frac{4q}{8},\ 101 \rightarrow \frac{5q}{8},\ 110 \rightarrow \frac{6q}{8},\ 111 \rightarrow \frac{7q}{8} \end{aligned} \tag{2}$$

$$\begin{aligned} &0000 \rightarrow 0,\ 0001 \rightarrow \frac{q}{16},\ 0010 \rightarrow \frac{2q}{16}, \\ &0011 \rightarrow \frac{3q}{16},\ 0100 \rightarrow \frac{4q}{16},\ 0101 \rightarrow \frac{5q}{16}, \\ &0110 \rightarrow \frac{6q}{16},\ 0111 \rightarrow \frac{7q}{16},\ 1000 \rightarrow \frac{8q}{16}, \\ &1001 \rightarrow \frac{9q}{16},\ 1010 \rightarrow \frac{10q}{16},\ 1011 \rightarrow \frac{11q}{16}, \\ &1100 \rightarrow \frac{12q}{16},\ 1101 \rightarrow \frac{13q}{16},\ 1110 \rightarrow \frac{14q}{16}, \\ &1111 \rightarrow \frac{15q}{16} \end{aligned} \tag{3}$$

알고리즘 3에 있는 $Frodo.Decode$ 함수는 다음과 같이 계산된다. $B = 2,3,4$에 대해 앨리스는 행렬 $M = C_2 - C_1 S$의 각 성분을 가장 가까운 $q/4$, $q/8$ 혹은 $q/16$의 배수로 반올림한다. 앨리스는 반올림한 값에 (1), (2) 혹은 (3)의 역 맵핑(inverse mapping)을 사용하여 메시지 $\mu'$를 얻는다.

### 2.2 그레이 부호 및 오류정정부호를 FrodoPKE에 적용하는 방법

FrodoPKE에 오류정정부호를 적용하고 분석하기 위해서 FrodoPKE를 디지털 통신 시스템의 측면에서 이해하는 것이 편리하다. 송신자는 밥에 해당되며 수신자는 앨리스에 해당된다. 밥이 보내기 원하는 공유키 $\mu$는 메시지 비트에 해당한다. 이진 비트를 심볼로 맵핑하는 것은 변조기(modulator)에 해당한다. 밥은 $Frodo.Encode(\mu)$를 계산하여 두 암호문 $C_1, C_2$를 생성한 후 송신한다. 앨리스는 $M = C_2 - C_1 S$를 계산한다. 그 결과 $Frodo.Encode(\mu)$에 에러 $E'''$가 더해지게 된다. 이 과정은 $Frodo.Encode(\mu)$가 디지털 통신에서의 노이즈 채널(noisy channel)을 통과한 것으로 볼 수 있다. $Frodo.Deocde$함수는 디지털 통신에서의 복조기(demodulator)에 해당한다.

만약 FrodoPKE에 오류정정부호를 사용한다면 부호기가 변조기 앞에, 복호기가 복조기 뒤에 위치하게 될 것이다. 구체적으로 $\mu$에 $Frodo.Encode$ 함수를 적용하기 전에 먼저 BCH 부호화를 수행하며, $Frodo.Decode(M)$을 계산한 후에 BCH 복호화를 수

행하여 $\mu'$을 얻는다. 또한 오류정정부호를 사용하는 경우 더 좋은 오류정정 성능을 위해 그레이 부호가 오류정정부호와 함께 사용되어야 한다. 이때, $Frodo.Encode$ 함수가 그레이 부호로 대체된다.

기존 연구[11]에서는 이진 BCH 부호(binary BCH code)를 사용한다. BCH 부호의 부호 길이는 소수 $q$에 대해 $n=q^m-1$이다. 이진 부호이므로 $q=2$이다. BCH 부호는 $(l_n,l_k,l_t)$로 표기될 수 있으며 여기서 $l_n$은 코드워드 길이, $l_k$는 메시지 길이, $l_t$는 최대 오류정정 가능 개수이다. (192, 128, 8), (256, 128, 18), (256, 192, 8) 이진 BCH 부호가 사용된다. 이들 변형된 BCH 부호는 기존 BCH 부호에 쇼트닝(shortening)[12] 혹은 익스텐딩(extending)[13]을 사용하여 얻어진다.

$l_n$이 192 혹은 256인 경우 128-비트 메시지 $\mu$는 192-비트 혹은 256-비트 코드워드 $c$로 부호화된다. 이때 각각 $B=3$ 및 $B=4$가 되며 기존 $Frodo.Encode$ 함수 대신 다음 그레이 부호가 각각 사용된다[11].

$$\begin{aligned}&000\rightarrow 0,\ 001\rightarrow \frac{q}{8},\ 011\rightarrow \frac{2q}{8},\ 010\rightarrow \frac{3q}{8}\\&110\rightarrow \frac{4q}{8},\ 111\rightarrow \frac{5q}{8},\ 101\rightarrow \frac{6q}{8},\ 100\rightarrow \frac{7q}{8}\end{aligned} \tag{4}$$

$$\begin{aligned}&0000\rightarrow 0,\ 0001\rightarrow \frac{q}{16},\ 0011\rightarrow \frac{2q}{16},\\&0010\rightarrow \frac{3q}{16},\ 0110\rightarrow \frac{4q}{16},\ 0111\rightarrow \frac{5q}{16},\\&0101\rightarrow \frac{6q}{16},\ 0100\rightarrow \frac{7q}{16},\ 1100\rightarrow \frac{8q}{16},\\&1101\rightarrow \frac{9q}{16},\ 1111\rightarrow \frac{10q}{16},\ 1110\rightarrow \frac{11q}{16},\\&1010\rightarrow \frac{12q}{16},\ 1011\rightarrow \frac{13q}{16},\ 1001\rightarrow \frac{14q}{16},\\&1000\rightarrow \frac{15q}{16}\end{aligned} \tag{5}$$

### 2.3 Frodo-640 및 Frodo-976 개선

BCH 부호를 사용하여 Frodo-640 및 Frodo-976의 보안성 수준이 개선될 수 있다. 오류정정부호를 사용하려면 $B$가 증가하여야 하는데 이는 실패확률을 증가시킨다. 그러나 BCH 부호의 사용으로 인해 실패확률은 감소할 수 있다. Frodo-640, Frodo-976, Frodo-1344에서 실패확률은 각각 $2^{-148.8}$,$2^{-199.6}$, $2^{-252.5}$이다. 암호시스템 및 파라미터의 변경으로 달라진 실패확률이 기존 실패확률보다 크지 않도록 하는 범위 내에서 $\sigma$를 최대로 증가함으로써 보안성 수준을 최대로 향상한다. 다음 세 가지 경우와 같이 Frodo-640 혹은 Frodo-976의 보안성 수준을 향상할 수 있다[11].

i) 경우 1: Frodo-640에 (192, 128, 8) BCH 부호 사용 보안성 수준; 149.30→156.98
ii) 경우 2: Frodo-640에 (256, 128, 18) BCH 부호 사용 보안성 수준; 149.30→152.25
iii) 경우 3: Frodo-976에 (256, 192, 8) BCH 부호 사용 보안성 수준; 215.66→225.97

## Ⅲ. 오류정정부호를 이용한 Frodo-1344 개선 방법

본 논문에서는 처음으로 Frodo-1344에 오류정정부호를 사용하여 암호 시스템을 변형하여 분석한다. 먼저 기존 $Frodo.Encode$ 함수를 그레이 맵핑으로 바꾼다. 이때, $Frodo.Decode$ 함수도 사용한 그레이 맵핑에 맞게 수정되어야 한다. 또한 암호 시스템 앞뒤에 BCH 부호 부호화 및 복호화를 추가한다. 이렇게 Frodo-1344를 변형시켜서 보안성 수준을 향상하고자 한다.

그레이 부호를 사용하는 이유는 심볼오류확률(symbol error rate, SER)을 낮추기 위해서이다. Frodo-1344에 오류정정부호를 적용하기 때문에 $B=5$에 대한 다음과 같은 그레이 부호를 사용한다.

$$\begin{aligned}&00000\rightarrow 0,\ 00001\rightarrow \frac{q}{32},\ 00011\rightarrow \frac{2q}{32},\\&00010\rightarrow \frac{3q}{32},\ 00110\rightarrow \frac{4q}{32},\ 00111\rightarrow \frac{5q}{32},\\&00101\rightarrow \frac{6q}{32},\ 00100\rightarrow \frac{7q}{32},\ 01100\rightarrow \frac{8q}{32},\\&01101\rightarrow \frac{9q}{32},\ 01111\rightarrow \frac{10q}{32},\ 01110\rightarrow \frac{11q}{32},\\&01010\rightarrow \frac{12q}{32},\ 01011\rightarrow \frac{13q}{32},\ 01001\rightarrow \frac{14q}{32},\\&01000\rightarrow \frac{15q}{32},\ 11000\rightarrow \frac{16q}{32},\ 11001\rightarrow \frac{17q}{32},\\&11011\rightarrow \frac{18q}{32},\ 11010\rightarrow \frac{19q}{32},\ 11110\rightarrow \frac{20q}{32},\\&11111\rightarrow \frac{21q}{32},\ 11101\rightarrow \frac{22q}{32},\ 11100\rightarrow \frac{23q}{32},\\&10100\rightarrow \frac{24q}{32},\ 10101\rightarrow \frac{25q}{32},\ 10111\rightarrow \frac{26q}{32},\\&10110\rightarrow \frac{27q}{32},\ 10010\rightarrow \frac{28q}{32},\ 10011\rightarrow \frac{29q}{32},\\&10001\rightarrow \frac{30q}{32},\ 10000\rightarrow \frac{31q}{32}\end{aligned} \tag{6}$$

$Frodo.Decode$ 함수는 행렬 M의 각 성분을 가장 가까운 $q/32$의 배수로 반올림한 다음 그레이 부호 (6)의 역 맵핑을 적용하면 된다.

오류정정부호를 사용하기 위해서 $B$는 4에서 5로 증가하여야 하므로 메시지 길이가 $64\times4=256$, 코드워드 길이가 $64\times5=320$인 BCH 부호를 필요로 한다. $l_n=320$, $l_k=256$인 BCH 부호 파라미터는 기본으로 존재하는 파라미터가 아니므로 알려진 BCH 부호 파라미터로부터 쇼트닝 및 익스텐딩과 같은 기법을 사용하여 얻어야한다. (511, 448, 7) BCH 부호를 192번 쇼트닝을 수행하면 (319, 256, 7) BCH 부호가 된다. 이 때, $l_t=7$이 홀수이므로 코드워드의 모든 비트의 합이 짝수가 되도록 익스텐딩을 수행하면 (320, 256, 8) BCH 부호를 얻을 수 있다. 그림 1은 Frodo-1344에 (320, 256, 8) BCH 부호를 사용한 것을 나타낸 것이다.

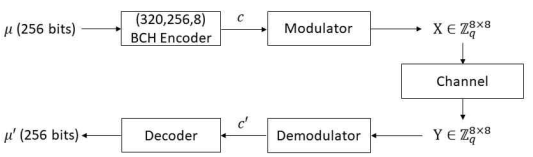


그림 1. Frodo-1344에 (320, 256, 8) BCH 부호를 사용
Fig. 1. Frodo-1344 with (320, 256, 8) BCH code

## Ⅳ. 실 험

본 논문에서는 변형된 시스템에 대한 보안성 수준을 구하여 기존 보안성 수준 대비 향상 정도를 계산하였다. 실험의 과정은 다음과 같다.

1) 주어진 $\sigma$에 대한 노이즈 분포 $\chi'$을 구한다. 이때, 알고리즘 3에 있는 $E'''$의 각 성분이 $\chi'$를 따른다.
2) SER = $\sum_{e\notin[-q/2^{B+1},q/2^{B+1})}\chi'(e)$ 식을 계산한다. 여기서 SER는 각 심볼에서 오류가 발생할 확률인 심볼오류확률이다.
3) 심볼오류확률을 이용하여 실패확률을 구한다.
4) 파라미터 $\sigma,q,n$에 대해 보안성 수준을 구한다.

본 논문에서 위 실험을 수행하기 위해서 FrodoKEM에서 제공한 파이썬 언어로 된 소스 파일을 이용하였다[9]. 노이즈 분포 $\chi'$, 심볼오류확률 및 보안성 수준은 위 소스 파일을 이용하여 구할 수 있다. 한편 실패확률은 심볼오류확률을 사용하여 다음과 같이 계산된다. 오류정정부호가 최대 $t$개의 오류를 정정한다고 하자. 64개의 $B$-비트 메시지는 64개의 심볼로 부호화된다. 각 심볼에서 오류가 발생할 확률은 $p$이다. 이때, 실패확률은 64개의 심볼 중 $t$개보다 많은 심볼오류가 발생할 확률이며 아래와 같이 계산된다. 본 논문은 (320, 256, 8) BCH 부호를 사용하므로 여기서 $t$는 8이다.

$$\sum_{i=t+1}^{64}\binom{64}{i}p^i(1-p)^{64-i} \tag{7}$$

$\sigma,q,n$과 같은 파라미터 값들을 입력으로 넣어 FrodoKEM 소스 파일을 실행시키면 심볼오류확률 및 보안성 수준을 구할 수 있고, 심볼오류확률을 통해 (7)을 사용하여 실패확률을 계산할 수 있다. 실패확률이 기존 Frodo-1344의 실패확률인 $2^{-252}$를 초과하지 않게 하는 범위 내에서 $\sigma$를 최대한 증가시키면 그때 보안성 수준의 최댓값을 얻을 수 있다.

Frodo-1344에 (320, 256, 8) BCH 부호를 적용해 변형시킨 시스템의 성능을 계산한 결과는 다음과 같다.

표 1. Frodo-1344 및 Frodo-1344에 (320, 256, 8) BCH 부호를 적용한 시스템의 성능 비교
Table 1. Comparison between Frodo-1344 and Frodo-1344 with (320, 256, 8) BCH code

| | $B$ | $\sigma$ | 보안성 수준 | SER | DFR |
|---|---|---|---|---|---|
| Frodo-1344 | 4 | 1.4 | 281.57 | $2^{-258}$ | $2^{-252}$ |
| Frodo-1344에 BCH(320, 256, 8) 적용 | 5 | 1.74 | 294.51 | $2^{-31}$ | $2^{-252}$ |

이 결과로부터 Frodo-1344에 BCH 부호를 적용함으로써 실패확률은 그대로 유지하면서 보안성 수준이 281.57비트부터 294.51비트까지 약 13비트 향상함을 확인할 수 있다. 즉, 이 암호 시스템을 공격자가 무력화시키기 위해서는 기존 Frodo-1344 시스템을 공격할 때에 비해 $2^{13}$배의 연산을 더 필요로 한다.

## Ⅴ. 결 론

본 논문에서는 Frodo-1344의 보안성 수준을 높이기 위해 BCH 부호를 적용할 것을 제안하였다. Frodo-1344 파라미터에 맞는 부호를 사용하기 위해 쇼트닝 및 익스텐딩 기법을 사용하였다. 또한 심볼오류확률을 낮추기 위해 $B=5$에 대한 그레이 부호를

사용하였다. 그 결과 실패확률을 기존 Frodo-1344의 실패확률과 동일하게 유지하면서 Frodo-1344의 보안성 수준을 13비트 향상할 수 있었다.

## References


[1] D. Micciancio, "Lattice-based cryptography," *Post - Quantum Cryptography*, LNCS, Berlin, Germany: Springer, pp. 147-191, 2011.

[2] C. Peikert, "A decade of lattice cryptography," *Foundations and Trends in Theoretical Comput. Sci.*, vol. 10, no. 4, pp. 283-424, 2016.

[3] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 1-37, 2009.

[4] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Proc. Advances in Cryptology - Eurocrypt*, LNCS, vol. 6110, pp. 1-23, Berlin, Germany, Springer, 2010.

[5] T. Poppelmann, E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. Piedra, P. Schwabe, and D. Stebila, *NewHope*(2019), Retrieved Feb. 9, 2020, from https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[6] S. Bhattacharya, O. Garcia- Morchon, T. Laarhoven, R. Rietman, M. O. Saarinen, L. Tolhuizen, and Z. Zhang, *Round5: compact and fast post-quantum public-key encryption* (2019), Retrieved Feb. 9, 2020, from https://eprint.iacr.org/2019/090.

[7] X. Lu, Y. Liu, D. Jia, H. Xue, J. He, and Z. Zhang, *LAC*(2019), Retrieved Feb. 9, 2020, from https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[8] M. Hamburg, *Three Bears* (2019), Retrieved Feb., 9, 2020, from https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[9] M. Naehrig, E. Alkim, J. Bos, L. Ducas, K. Easterbrook, B. LaMacchia, P. Longa, I. Mironov, V. Nikolaenko, C. Peikert, A. Raghunathan, and D. Stebila, *FrodoKEM* (2019), Retrieved Feb. 9, 2020, from https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[10] E. S. Lee and J. S. No, "Improvement of Frodo scheme using error correcting code," in *Proc. KICS Winter Conf.*, pp. 1506-1507, Jan. 2019.

[11] E. S. Lee, Y. S. Kim, J. S. No, M. K. Song, and D. J. Shin, "Modification of FrodoKEM using Gray and error-correcting codes," *IEEE Access*, vol. 7, pp. 179564-179574, 2019.

[12] H. Helgert and R. Stinaff, "Shortened BCH codes," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 818-820, 1973.

[13] R. E. Blahut, *Algebraic codes for data transmission*, Cambridge, U.K.: Cambridge university press, 2003.



**이 은 상** (Eunsang Lee)

2014년 8월 : 서울대학교 전기·정보공학부 졸업

2014년 9월~현재 : 서울대학교 전기·정보공학부 석박사 통합 과정

<관심분야> 암호, 부호이론

[ORCID:0000-0002-5270-2405]

**노 종 선** (Jong-Seon No)

1981년 2월 : 서울대학교 전자공학과 공학사

1984년 2월 : 서울대학교 대학원 전자공학과 공학석사

1988년 5월 : University of Southern California 전기공학과 공학박사

1988년 2월~1990년 7월 : Hughes Network Systems, Senior MTS

1990년 9월~1999년 7월 : 건국대학교 전자공학과 부교수

1999년 8월~현재 : 서울대학교 전기·컴퓨터공학부 교수

<관심분야> 시퀀스, 협력통신, 시공간부호, 네트워크코딩, LDPC 부호, OFDM, 이동통신, 암호학

[ORCID:0000-0002-3946-0958]