정 보 보 호 학 회 지
제 33 권 제3호, 2023. 6

KpqC에 제출된 코드기반 암호의 소개 및 분석

홍 지 훈*, 원 병 선**, 김 종 락***

요 약

Shor 알고리즘의 영향으로 RSA 등 기존 잘 알려진 암호 알고리즘들은 양자컴퓨터가 등장하면 안전성에 대한 위험을 받게 된다. NIST에서는 양자컴퓨터 환경으로부터 보안성이 유지되는 암호를 선정하기 위한 양자내성암호 공모전을 개최하였다. 또한, 국내에서도 양자내성암호 분야의 경쟁력 향상과 기술 저변 확대를 위해 양자내성암호 국가공모전(KpqC 공모전)을 개최하였다. 본 논문은 KpqC 공모전 1라운드에 제출된 4개의 코드 기반 양자내성암호 알고리즘에 대한 소개, 퍼포먼스 및 안전성에 관하여 살펴볼 것이다.

Ⅰ. 서 론

1994년 Shor의 알고리즘이 제시되면서 현재 사용하고 있는 여러 암호 알고리즘들은 미래 양자컴퓨터가 등장하면 다항 시간 내에 분석될 것으로 예측되고 있다. 이에 대항하기 위한 양자내성암호 알고리즘은 격자 기반, 코드 기반, 다변수 기반 등 다양한 수학적 난제에 기반하고 있다. 그 중 코드 기반 암호는 신드롬 복호화 문제가 NP-hard라는 것 또는 랭크 신드롬 복호화 문제가 NP-completeness라는 것에 기반하고 있다. 이에 따라 NIST는 2016년부터 양자 내성 암호 공모전을 시작하여 2022년 7월 5일 4개의 암호화 알고리즘을 채택하였다. 국내에서도 양자내성암호 분야의 경쟁력 향상과 기술 저변 확대를 위해 국가보안연구소와 양자내성암호 연구단(이하 KpqC 연구단)이 주관하여 2021년 후반기에 양자내성암호 국가공모전(이하 KpqC 공모전)을 개최하였다. 현재 1라운드가 진행 중에 있으며 제출된 알고리즘들 중 4개의 코드 기반 양자내성암호 알고리즘들에 대해 소개하고자 한다.

Ⅱ. KpqC에 제출된 코드 기반 암호 알고리즘

2.1. Enhanced pqsigRM

Enhanced pqsigRM은 4개의 코드 기반 양자내성암호 알고리즘 중 유일한 디지털 서명에 대한 알고리즘이다. 기존에 있던 CFS 서명 기법은 고파 코드를 기반으로 하며, 매개 변수의 확장과 선택 메시지 공격에서 일부 단점이 존재하고 EUF- CMA에서 안전하지 않다. 또한 고파 코드의 오류 정정 범위를 t라 할 때, 서명 시간은 $t!$에 의존하게 되어 t가 작아야 한다. 이러한 제약 때문에 높은 보안성을 위해 키 크기 자체를 크게 증가시켜야 한다는 단점이 존재한다. Enhanced pqsigRM은 CFS에 사용된 코드를 수정된 리드-뮬러(modified Reed-Muller) 코드로 변경하여 EUF-CMA 안전하도록 설계하였으며 이는 NP-hard인 신드롬 복호화 문제에 기반한다.

리드-뮬러 코드는 1954년 I.리드[2]와 D.E.뮬러[3]에 의해 처음 소개되었다. 이 코드는 여러 가지 정의가 있지만, 재귀적인 정의[4]를 채택하며 이 구조를 이용한 재귀 디코딩을 사용한다. 재귀적 정의는 다음과 같다.

리드-뮬러 코드는 이진 선형 $(n=2^m, k=\sum_{i=0}^{r}\binom{m}{i})$ 코드이며 다음과 같이 정의된다.

$$RM_{(r,m)} := \{(u|u+v)\,|\,u\in RM_{(r,m-1)},\ v\in RM_{(r-1,m-1)}\}$$

여기서, $RM_{(0,m)} := \{(0,...,0),(1,...,1)\}$ 의 코드 길이는 2^m 이며, $RM_{(m,m)} := \mathbb{F}_2^{2^m}$ 이다.

* 서강대학교 수학과 (대학원생, rjekfl@sogang.ac.kr)
** 서강대학교 수학과 (대학원생, bswon@sogang.ac.kr)
*** 서강대학교 수학과 (교수, jlkim@sogang.ac.kr)

이렇게 정의된 리드-뮬러 코드 $RM_{(r,m)}$의 생성 행렬 $G_{(r,m)}$은 다음(1)과 같이 주어진다.

$$G_{(r,m)} = \begin{bmatrix} G_{(r,m-1)} & G_{(r,m-1)} \\ 0 & G_{(r-1,m-1)} \end{bmatrix}$$

그러나, 이러한 재귀적인 리드-뮬러 코드의 생성 행렬의 성질은 계산이 빠르고 디코딩이 간단하다는 장점이 있지만, 공격자가 암호 알고리즘의 구조를 파악하여 정보를 얻을 가능성 또한 높아진다. 이에 따라 기존의 생성 행렬 $G_{(r,m)}$의 일부를 교체(replacing), 추가(appending), 패딩(padding)의 방법을 사용하여 수정된 리드 뮬러 코드를 생성한다. [그림 1]은 수정된 리드 뮬러 코드에 사용할 수 있는 디코딩 알고리즘을 나타낸 것이다.

Enhanced pqsigRM의 서명 기법은 다음과 같다.

1. 키 생성 :
   G_M : 수정된 리드-뮬러 코드의 $k \times n$ 생성 행렬
   H_M : 수정된 리드-뮬러 코드의 $(n-k) \times n$ 패리티 검사 행렬

Algorithm 3 Decoding for modified RM code

```
function DECODE(s; H)
    r ← PRANGE(H, s)
    while True do
        r ← r+ random codeword
        c ← MODDEC(r, r, M)
        if wt(r + c) ≤ w then
            Output r + c
        end if
    end while
end function

function MODDEC(y, r, M)
    y ← y^{σ^{-1}}
    if r = 0 then
        Output MD decoding on RM(0, m)
    else if r = m then
        Output MD decoding on RM(r, r)
          or replaced (2^r, k_rep) code
    else
        (y'|y'') ← y
        y^v = y' · y''
        v̂ ← MODDEC(y^v, r − 1, m − 1)
        y^u ← (y' + y'' · v̂)/2
        û ← MODDEC(y^u, r, m − 1)
        y ← (û|û · v̂)
    end if
    Output y^σ
end function
*σ is σ_p^1 or σ_p^2 for permuted block and identity, otherwise.
```

〔그림 1〕 수정된 리드-뮬러 코드의 디코딩 알고리즘

$S \xleftarrow{\$} F_2^{(n-k)\times(n-k)}$, $Q \xleftarrow{\$} F_2^{n\times n}$

$H' \leftarrow SH_MQ$

$H'_{sys} = (I|T)$: H'의 systematic 형태

공개키 : T

비밀키 : Q, σ_p^1, σ_p^2, $k_{rep} \times 2^r$ (반복적인) replacing 코드, $k_{app} \times n$ appending 코드, $1 \times n$ padding 쌍대 코드의 codeword.

2. 서명 :
   M : 메시지, $i \leftarrow \{0,1\}^{\lambda_0}$: 카운터
   $s \leftarrow h(M|i)$: 신드롬
   $s'^T \leftarrow S^{-1}s^T$
   $e' \leftarrow \text{Decode}(s'; H_M)$
   $e^T \leftarrow Q^{-1}e'^T$
   서명 : (M, e, i)
3. 인증 :
   If $wt(e) \le w$ and $H'_{sys}e^T = h(M|i)$,
   return ACCEPT
   Else, return REJECT
   ※ h : SHAKE-128/256 해쉬 함수
   ※ DECODE : 수정된 리드-뮬러 코드의 디코딩 알고리즘
   ※ $wt(a)$: 벡터 a의 해밍 weight(해밍 가중치)
   ※ w : 수정된 리드-뮬러 코드의 오류 정정 능력

2.2. Layered ROLLO-1

Layered ROLLO-1는 공개 키 암호화 및 키 설정 알고리즘으로 랭크 거리 코드를 사용한 암호 시스템이며 NP-completeness인 랭크 신드롬 복호화 문제에 기반한다. 기존에 작은 키 크기의 저랭크 패리티 체크(LRPC) 코드를 기반으로 하는 ROLLO와 McNie 암호 시스템이 NIST PQC 경연에 진출하였으나, 격자 기반 암호 시스템에 비해 복호화 성능이 다소 떨어지는 것으로 알려져 있다[5]. 기존 알고리즘의 한계를 극복하기 위해 Layered ROLLO-1 알고리즘은 (LRPC) 코드의 구조를 수정한 ideal LRPC (BII-LRPC) 코드를 새롭게 도입하였다. 보안 수준을 향상시키기 위해 저랭크 벡터를 두 개의 무작위 다항식에 곱하여 공격자가 구조적인 특성을 사용할 수 없도록 하였다.

Ideal LRPC 코드[6]의 정의는 다음과 같다. F를 랭크 무게가 d인 n튜플 벡터의 $\mathbb{F}_q$-부분집합을 F라 하자. 벡터 $x,y \in F$에 대해 $(2n,n)$ ideal LRPC 코드 C의 패리티 검사 행렬 H는 $H=[\Gamma(x,P)|\Gamma(y,P)]$로 정의되며 $n\times n$이상 행렬 $\Gamma(x,P)$는

$$\Gamma(x,P)=\begin{pmatrix} x \\ Xx \bmod P \\ \vdots \\ X^{n-1}x \bmod P \end{pmatrix}$$

이다.

Layered-ROLLO 1 KEM 알고리즘은 다음과 같다.

1. 키 생성 :

   F : 랭크 가중치가 d인 $\frac{n}{b}$-튜플 벡터의 집합

   앨리스(Alice)는 집합 F에서 랜덤한 $\frac{n}{b}$-튜플 벡터 x, y 2개를 선택한다. 또한, 랜덤 $(b-1)$차 원시 다항식 $P_I \in \mathbb{F}_{q^m}[X]/<P>$와 n차 다항식 P_O, $P_N \in \mathbb{F}_{q^m}[X]/<P^b>$라 한다.

   x, y에 대하여,

   $z = P_I x^{-1} y \mod P$.

   마지막으로 앨리스는 공개키(PK)를 아래와 같이 구성한다.

   $P_P = P_O P_I \mod P^b$,

   $h = P_O z' + P_N P \mod P^b$,

   여기서 $z' = [0, z]$이다.

   비밀키(SK)는 x, y, P_I, P_O 이다.

2. 캡슐화 (Encapsulation) :

   E : 랭크 가중치가 r이고, 최대 차수가 $\frac{n}{b}-b$인 $\frac{n}{b}$-튜플 벡터의 집합

   밥(Bob)은 집합 E에서 랜덤한 두 개의 $\frac{n}{b}$-튜플 벡터 (e_1, e_2)를 선택한다.

   또한, 길이가 n인 벡터 $e'_1 = [0, e_1]$, $e'_2 = [0, e_2]$라 한다. 그러면 암호문은 아래와 같이 생성한다.

   $$c = P_P e'_1 + h e'_2 \mod P^b$$

   여기서, P_P와 h는 공개키이다. 앨리스와 밥이 알고있는 해쉬 함수 Hash(·)를 이용하여

   $k_1 = \text{Hash}(E)$

   를 계산한다.

   마침내, c는 앨리스에게 보내고, k_1는 공유 암호(SS)로 사용한다.

3. 역캡슐화 (Decapsulation) :

   비밀키(SK)인 x와 P_P를 사용하여, 아래를 계산한다.

   $$c' = P_O^{-1} c \mod P^b,$$
   $$c'' = P_I^{-1}\{c' \mod P\} \mod P,$$
   $$xc'' = xe_1 + ye_2 \mod P.$$

   그 다음, RSR 알고리즘을 이용하여 xc''를 복호화하여 E'를 재구성한다. $k_2 = \text{Hash}(E')$으로부터 앨리스는 $k_1 = k_2$를 확인함으로서 정확성을 검증할 수 있다.

2.3. PALOMA

PALOMA는 전통적인 McEliece 암호시스템에 사용되는 고파 코드를 기반으로 하는 공개키 암호화 알고리즘이다. Niederreiter 암호시스템과 비슷하게 가역 행렬과 순환 행렬을 사용하여 패리티 검사 행렬을 섞어서 사용한다. Niederreiter 암호시스템은 복호화를 위해 특정한 해밍 무게로 메세지를 변환하는 과정을 수행하지만 이 과정은 암호화 및 복호화 과정의 계산량을 증가시키는 역할을 한다. 따라서 PALOMA는 이러한 변환을 사용하지 않도록 설계되었으며 NP-hard인 신드롬 복호화 문제에 기반한다.

PALOMA에 사용되는 고파 코드는 다음과 같다. 이진 분리 가능한 고파 코드[7] $C=[n,k,\geq 2t+1]_2$는 지원 집합(support set) L과 분리 가능한 고파 다항식 $g(X)$으로 정의한다. PALOMA에서는 지원 집합과 고파 다항식을 다음과 같이 정의한다.

$[\alpha_0,\alpha_1,\ldots,\alpha_{2^m-1}] \leftarrow \text{SUFFLE}(\mathbb{F}_{2^m})$,

$$L \leftarrow [\alpha_0,\alpha_1,\ldots,\alpha_{n-1}],\ g(X) \leftarrow \prod_{j=n}^{n+t-1}(X-\alpha_j)$$

이렇게 생성하면 $g(X)$는 $\mathbb{F}_{2^{13}}[X]$에서 reducible separable이므로 고파 코드를 상수 시간 내에 효율적으로 생성할 수 있다.

PALOMA의 암호화 알고리즘은 다음과 같다.

1. 키 생성 :

PALOMA의 키 생성은 아래의 방법을 사용한다.

Step 1. 무작위 이진 분할가능 고파 코드 C를 생성한다. 지원 집합 $L \subset \mathbb{F}_{2^{13}}$과 고파 코드 $C_{L,g}$를 위한 고파 다항식 $g(X) \in \mathbb{F}_{2^{13}}[X]$을 생성하고, 고파 코드 $C_{L,g}$의 패리티 검사 행렬 $H \in \mathbb{F}_2^{13t \times n}$을 계산한다.

Step 2. C의 뒤섞인 코드 $\hat{C}$을 생성한다.

C의 패리티 검사 행렬 H를 아래와 같이 계산하여 뒤섞는다.

- SHUFFLE을 사용하여 임의의 256비트 문자열 r로 $[n]=[1, 2, ..., n-1]$의 요소를 재정렬한다.
- 순환 행렬 $P \in P_n$, 가역 행렬 $S \in \mathbb{F}_2^{(n-k)\times(n-k)}$을 이용하여 $\hat{H}$를 계산한다.

$\hat{H} = SHP$.

Step 3. $\hat{H}$는 systematic 형태 행렬이다. 즉, $\hat{H}_{[n-k]} = I_{n-k}$ 이다.

공개키(pk) : $\hat{H}_{[n-k:n]} \in \mathbb{F}_2^{(n-k)\times k}$

비밀키(sk) : $(L,\ g(X),\ S^{-1},\ r)$

※ SHUFFLE : 256비트 임의의 비트 문자열 r을 16비트 시퀀스로 구문 분석하고 각각을 Fisher-Yates 셔플에 필요한 임의의 정수로 사용한다.

2. 암호화 :

PALOMA의 암호화는 아래의 방법을 사용한다.

Step 1. 공개 키 pk=$\hat{H}_{[n-k:n]} \in \mathbb{F}_2^{(n-k)\times k}$로 부터 뒤섞인 코드 C의 패리티 검사 행렬 $\hat{H}=[I \mid \hat{H}_{[n-k:n]}]$를 계산한다.

Step 2. $\hat{H}$에 대한 n-비트 입력 $\hat{e} \in \{0,1\}^n$의 $(n-k)$-비트 신드롬 $\hat{s}(=\hat{H}\hat{e})$를 $\hat{e}$의 암호문으로 반환한다.

3. 복호화 :

PALOMA의 복호화는 아래의 방법을 사용한다.

Step 1. 비밀키 S^{-1}를 곱하여, $\hat{C}$의 신드롬 $\hat{s} \in \{0,1\}^{n-k}$을 C의 신드롬 s $(=S^{-1}\hat{s})$로 변환한다.

Step 2. C의 복호화 정보인 비밀키 L, $g(X)$로 s에 대응하는 오류 벡터 e를 복구한다. 그 때, 우리는 확장된 Patterson 디코딩 알고리즘을 사용한다.

Step 3. 오류 벡터 e 에서 얻은 $\hat{C}$의 오류 벡터 $\hat{e}(=P^{-1}e)$과 비밀키 r에 의해 생성된 순환 행렬 P^{-1}을 계산한다.

PALOMA의 KEM 알고리즘은 다음과 같다.

1. 캡슐화(Encapsulation)

Step 1. SUFFLE을 사용하여 랜덤 256 비트 스트링 r과 $[n]$의 원소들을 재요청한다. $([n] \rightarrow [l_0, ..., l_{n-1}])$

Step 2. $\mathrm{supp}(e^*) = \{l_0, ..., l_{t-1}\}$ 을 만족하는 n비트 에러 벡터 $e^* \in \{0,1\}^n$를 정의한다.

Step 3. e^*를 랜덤 오라클 RO_G에게 질문하여 256 비트의 스트링 $\hat{r} \in \{0,1\}^{256}$을 얻는다.

Step 4. $\hat{r}$에 대응하는 순환 행렬 P, P^{-1}를 계산한다.

Step 5. $\hat{e} = Pe^*$를 계산한다.

Step 6. ENCRYPT와 공개키 pk를 사용하여 $\hat{e}$의 신드롬 $\hat{s} \in \{0,1\}^{n-k}$를 얻는다.

Step 7. 그 후 $(e^* \| \hat{r} \| \hat{s})$를 랜덤 오라클 RO_H에게 질문하여 256 비트의 키 $k \in \{0,1\}^{256}$을 얻는다.

Step 8. k와 그 암호문 $c = (\hat{r}, \hat{s})$를 반환한다.

2. 역캡슐화(Decaptulation)

Step 1. 비밀키 sk로 설정된 DECRYPT 함수에 $\hat{s}$를 넣어 에러 벡터 $\hat{e}$를 얻는다.

Step 2. 암호문 c의 일부인 $\hat{r}$로부터 순환 행렬 P, P^{-1}을 생성한다.

Step 3. $e^{*}=P^{-1}\hat{e}$를 계산한다.

Step 4. RO_G에게 e^{*}를 질문하여 256비트 스트링 $\hat{r}'\in\{0,1\}^{256}$을 얻는다

Step 5. GENRANDERRVEC와 비밀키 r을 사용하여 에러 벡터 $\tilde{e}$를 생성한다.

Step 6. 만약 $\hat{r}'=\hat{r}$이면, $(e^{*}\|\hat{r}\|\hat{s})$를 랜덤 오라클 RO_H에 질문하고, 그렇지 않으면 $(\hat{e}\|\hat{r}\|\hat{s})$를 질문한다. RO_H로부터 받은 비트 스트링을 키 k로 반환한다.

2.4. REDOG

NIST PQC 경연에 제안했던 McNie[8]는 코드 기반 암호 알고리즘인 McEliece와 Niederreiter의 특징을 모두 포함하여 알려진 구조화 공격(known structural attack)으로부터 안전하도록 설계되었다. 그러나 Gaborit[9]이 공개키 랜덤 코드의 차원을 감소시키는 메세지 복원 공격을 제안하여 안전성이 감소되었다. 이를 개선하기 위해 McNie를 개선한 Dual-Ouroboros 알고리즘[10]이 제안되지만, 여기에 사용된 LRPC 코드에 대한 단점이 발견되어 랭크 거리를 사용하는 다른 코드인 가비둘린 코드로 대체한 DO.Gab-PKE[11]를 제안하였다. 그러나 기존의 알고리즘에서 가역행렬 S를 선택하는 제한조건이 없는 경우에 암호 알고리즘이 안전하지 않다는 것이 밝혀져, 이를 보안하여 REDOG을 제안하였다. REDOG은 NP-completeness인 랭크 신드롬 복호화 문제에 기반한다.

REDOG에 사용된 가비둘린 코드는 다음과 같다. 랭크가 n인($n\le m$) $g\in\mathbb{F}_{q^m}^{n}$를 선택한 뒤 $\mathbb{F}_{q^m}^{n}$ 위의 차원이 k인 생성 벡터가 g인 가비둘린 코드 $Gab_{n,k}(g)$는 g로부터 유도되는 무어 행렬(Moore matrix)로부터 생성되는 코드이다. 무어 행렬 G의 형태는 다음과 같다.

$$G=\begin{bmatrix} g_1 & g_2 & \cdots & g_n \\ g_1^{[1]} & g_2^{[1]} & \cdots & g_n^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{[n-1]} & g_2^{[n-1]} & \cdots & g_n^{[n-1]} \end{bmatrix}$$

REDOG의 암호화 알고리즘은 다음과 같다.

1. 설정 :

   $l<n$과 $\lambda t\le r\le\left\lfloor\frac{n-k}{2}\right\rfloor$가 되는 정수 $m,\ n,\ l,\ r,\ k$를 사용하여 전역 매개변수를 생성한다. 매개변수 $(m,\ n,\ l,\ k,\ r,\ \lambda,\ t)$를 출력한다.

2. 키 생성 :

   $[H_1\ H_2]$는 $\mathbb{F}_{q^m}$위에서의 $[2n-k,n]$ 가비둘린 코드의 패리티 검사 행렬이다. 여기서 $H_2\in GL_{n-k}(\mathbb{F}_{q^m})$이다.

   Φ_H는 $r=\left\lfloor\frac{n-k}{2}\right\rfloor$의 오류 정정 능력을 가진 C의 디코딩 알고리즘이다.

   $\boldsymbol{\mathcal{H}}$는 $\mathbb{F}_{q^m}^{2n-k}$에서 $\mathbb{F}_{q^m}^{l}$로 가는 해쉬 함수이다.

   $\mathbb{F}_{q^m}$위에서의 무작위 $[n,l]$ 코드의 생성 행렬 G와 $n\times n$크기의 isometric 행렬 P를 생성한다.

   $1\in\Lambda$이 있는 무작위의 λ-차원 부분 공간 $\Lambda\subset\mathbb{F}_{q^m}$을 생성한다.

   $(n-k)\times(n-k)$크기의 무작위적인 가역 행렬 $S^{-1}\in GL_{n-k}(\Lambda)$을 생성한다.

   아래의 공개키(pk)와 비밀키(sk)를 출력한다.

   pk = (G, $F=GP^{-1}H_1^{T}[H_2^{T}]^{-1}S$)

   sk = (P, H, S, Φ_H)

3. 암호화 :

   평서문 $m\in\mathbb{F}_{q^m}$을 암호화한다.

   무작위 벡터 $e=(e_1,e_2)\in\mathbb{F}_{q^m}^{2n-k}$을 생성한다.

   여기서 rk(e)=t이고, $e_1\in\mathbb{F}_{q^m}^{n}$, $e_2\in\mathbb{F}_{q^m}^{n-k}$이다.

   $m'=m+\boldsymbol{\mathcal{H}}(e)$라 한다. $c_1=m'G+e_1$, $c_2=m'F+e_2$를 계산한다.

   암호문 $c=(c_1,c_2)$를 출력한다.

4. 복호화 :

   $c_1P^{-1}H_1^{T}-c_2S^{-1}H_2^{T}=(e_1P^{-1},\ -e_2S^{-1})\begin{bmatrix}H_1^{T}\\ H_2^{T}\end{bmatrix}$

   를 계산한다.

   $e'=(e_1P^{-1},\ -e_2S^{-1})$라 한다. rk($e'$)$\le r$이기 때문에 Φ_H를 적용하여 e'를 얻는다.

   $e_1=e_1P^{-1}P$, $e_2=e_2S^{-1}S$ 계산하여

$e = (e_1, e_2)$를 얻는다.
마지막으로 시스템 $m'G = c_1 - e_1$를 풀어서
$m = m' - \mathcal{H}(e)$을 복구한다.

Ⅲ. 암호 알고리즘 별 퍼포먼스

위에서 소개한 KpqC 공모전에 제출된 4가지의 코드 기반 암호 알고리즘들의 파라미터를 소개한다.

3.1. Enhanced pqsigRM

아래 [표 1]은 Enhanced pqsigRM의 공개키 크기와 서명 크기를 나타낸 것이다. NIST PQC 경연에서 최종 선정된 디지털 서명 알고리즘들의 파라미터보다 공개키는 크게 나타났으나, 서명 크기는 가장 작게 나타났다.

〔표 1〕 Enhanced pqsigRM의 공개키와 서명 크기 (Bytes)

security	public key size	Signature
128	474,445	512
256	2,000,000	1,024

3.2. Layered ROLLO-1

아래 [표 2]는 Layered ROLLO-1의 공개키(PK), 비밀키(SK), 암호문(CT) 크기를 나타낸 것이다. 기존에 NIST PQC 경연에 제안했던 ROLLO-1보다 공개키, 비밀키, 암호문 크기가 전부 증가했다.

〔표 2〕 Layered ROLLO-1의 공개키, 비밀키, 암호문 크기(Bytes)

security	PK size	SK size	CT size
128	1240	120	620
192	1699	120	850
256	2571	120	1286

3.3. PALOMA

아래 [표 3]은 PALOMA의 공개키(PK), 비밀키(SK), 암호문(CT) 크기를 나타낸 것이다. NIST PQC 4라운드에 진출했던 다른 코드 기반 암호들인 hqc, BIKE, McEliece와 비교해 보았을 때 공개키, 비밀키는 상당히 크게 나타났으나, 암호문 크기는 가장 작은 수준으로 나타났다.

〔표 3〕 PALOMA의 공개키, 비밀키, 암호문 크기 (Bytes)

security	PK size	SK size	CT size
128	319,488	94,496	136
192	812,032	355,400	240
256	1,025,024	257,064	240

3.4. REDOG

아래 [표 4]는 REDOG의 공개키(PK), 비밀키(SK), 암호문(CT) 크기를 나타낸 것이다. NIST PQC 4라운드에 진출했던 다른 코드 기반 암호들인 hqc, BIKE, McEliece와 비교해 보았을 때 공개키, 비밀키는 상당히 크게 나타났으나 암호문 크기는 가장 작은 수준으로 나타났다.

〔표 4〕 REDOG의 공개키, 비밀키, 암호문 크기(Bytes)

security	PK size	SK size	CT size
128	14,250	1,450	830
192	32,840	2,520	1,440
256	62,980	3,890	2,230

Ⅳ. 안전성 분석

제안된 각각의 암호시스템에서 고려한 주요 공격 방법들에 대해 간단히 소개한다.

4.1. Enhanced pqsigRM

리드-뮬러 코드에 대한 잘 알려진 공격으로는 Minder-Shokrollahi[12] 공격과 Cizhov-Borodin [13] 공격이 있다. Enhanced pqsigRM은 수정된 리드-뮬러 코드를 사용하기 때문에 해당 공격에 대해 안전하다는 것을 보였다[14]. 또한 전자 서명 알고리즘을 공격하는 키 교체 공격에도 안전하다는 것을 보였다[12].

4.2. Layered ROLLO-1

랭크 거리 코드를 사용하는 암호 알고리즘에 대한 공격들은 Layered ROLLO-1의 암호시스템에서 E를 복구하는 것을 목표로 한다. 이것을 위해 전수조사 기법을 사용하여 정확한 P_O를 얻어야 하는데, 이것은 저랭크 다항식이 다항식 링 연산으로 계산되지 않기 때문에 어렵다. 또한 잘 알려진 구조적 공격과 일반적인 공격에 대한 안전성 수준은 [표 5]에 나타내었다.

〔표 5〕 Layered ROLLO-1의 KEM 처리 주기

security	Keygen	Encap	Decap
128	2,609,907	661,423	5,570,494
192	2,921,813	755,759	5,253,698
256	3,757,592	918,300	10,424,395

4.3. PALOMA

이진 분리 가능한 고파 코드에 대한 알려진 치명적인 공격은 없으나, 일반적인 랜덤 행렬과의 구별불가능성에 대한 고려가 필요하여 정보 집합 복호화(Information Set Decoding)에 대한 비트 계산을 진행하였다. 이는 잘 알려진 NP 난해 문제에 해당하는 신드롬 복호화 문제(Syndrome Decoding Problem)이며, 이에 해당하는 공격들로는 Exhaustive 검색, Birthday-type 복호화, 개선된 Birthday-type 복호화, Becker-Joux-May- Meurer 정보 집합 복호화가 있다. 해당 공격들에 대한 계산복잡도는 [표 6]과 같다.

〔표 6〕 PALOMA의 공격들에 대한 계산복잡도

security	BJMM-ISD	Improved Birthday-type decoding	Birthday-type decoding	Exhaustive Search
128	$2^{166.21}$	$2^{225.78}$	$2^{244.11}$	$2^{476.52}$
192	$2^{266.77}$	$2^{399.67}$	$2^{448.91}$	$2^{885.11}$
256	$2^{289.66}$	$2^{415.59}$	$2^{464.66}$	$2^{916.62}$

4.4. REDOG

REDOG 암호시스템의 안전성 분석에는 키 복구 공격, 평문 복구 공격, 등을 고려하였으며, IND- CPA 안전성 또한 확인하였다. 키 복구 공격은 $2(n-k)^2$개의 알려지지 않은 이차항 변수와 $n(n-k)$개의 알려지지 않은 일차항 변수가 존재하여 높은 기하급수 난이도를 가진다. 평문 복구 공격에 대해서는 REDOG에서 공개키가 r-Frobenius weak 코드를 생성하지 않는다. 이러한 경우 Frobenius weak 공격에 대해 저항한다[15]는 것이 잘 알려져 있으므로 REDOG은 평문 복구 공격에 안전하다. 또한 REDOG의 전신인 Do.Gab-PKE 알고리즘이 IND –CPA 안전성을 만족한다[11]는 것을 이미 보였고, 바뀐 비밀 행렬 S에 대해서 고려한 내용은 [16]의 정리 2에 잘 나타나 있다.

Ⅴ. 결 론

본 논문에서는 현재 진행 중인 KpqC 공모전에 제시된 4개의 코드 기반 암호 알고리즘에 대해 소개하였다. 이 중 1개는 디지털 서명, 3개는 공개키 암호화 및 키 설정 알고리즘이며, 각 알고리즘에 사용되는 기반 코드는 수정된 리드-뮬러 코드, LRPC 코드, 이진 분리 가능 고파 코드, 가비둘린 코드로 각 알고리즘마다 다양한 코드를 사용하였다. 암호시스템의 안전성 향상을 위해서는 시스템의 구성과 암호화 과정 등에 대한 지속적인 연구가 필요하다. 또한 코드 기반 암호시스템은 복호화 과정이 NP 난해라는 문제에 기반하고 있으므로, 다양한 코드에 대한 연구도 코드 기반 암호의 안전성 향상에 도움이 될 것이다.

참 고 문 헌

[1] N. Courtois, M. Finiasz, and N. Sendrier, “How to achieve a McEli ece-based digital signature scheme,” in Proc. Asiacrypt, 2248, pp. 157–174, Dec 2001.

[2] I. Reed, “A class of multiple-error -correcting codes and the decoding scheme,” Transactions of the IRE Professional Group on Information Theory, 4(4), pp. 38-49, Sep 1954.

[3] D. E. Muller, “Application of boolean algebra to switching circuit design and to error detection,” Transactions of the IRE Professional Group on Electronic Computers, 3, pp. 6-12, Sep 1954.

[4] I. Dumer, “Recursive decoding and its perform-

ance for low-rate Reed‐Muller codes," IEEE Trans. Inf. Theory, 50(5), pp. 811‐823, May 2004.

[5] J. Labalnche, L. Mortajine, O. Bench aalal, P.-L. Cayrel, and N. E. Marbet, "Optimized implementation of the NIST PQC submission ROLLO on microcontroller," Cryptology ePrint Archive: Report 2019/787, Jul 2019.

[6] N. Aragon, P. gaborit, A. Hauteville, O. Ruatta, and G. Zemor, "Low rank parity check codes: New decoding algorithms and applications to cryptography," IEEE Trans. on Inf. Theo., 65(12), pp. 7697-7717, Dec 2019.

[7] V.D. Goppa, "A new class of linear correcting codes," Probl. Inf. Transm. 6(3), pp. 24‐30, 1970

[8] McNie and other cryptosystems, https://csrc.nist.gov/Projects/ Post-Quantum- Cryptography/Post-Qua ntum-Cryptography-Standardization. Accessed 21 Nov 2019.

[9] McNie comment from https:// csrc.nist. gov/Projects/Post-Quantum‐Cryptogra phy/Round-1- Submissions, read official comments on McNie dated Dec 24, 2017 and Dec. 26, 2017.

[10] P. Gaborit, L. Galvez, A.Hauteville, J.-L. Kim, M.J. Kim, Y.-S. Kim, "Dual-Ouroboros: an improvement of the McNie scheme," Advances in Mathematics of Communications, 14(2), pp. 301-306, 2020.

[11] J.-L. Kim, Y.-S. Kim, L.E. Galvez, M.J. Kim, "A modified Dual-Ouroboros public-key encryption using Gabidulin codes," Applicable Algebra in Engineering, Communication and Computing, 32(2), pp. 147-156, 2021.

[12] L. Minder and A. Shokrollahi, "Cryptanalysis of the Sidelnikov crypto system," in Proc EUROCRYPT 2007, 26 pp. 347‐360, May 2007.

[13] I. V. Chizhov and M. A. Borodin, "The failure of McEliece PKC based on Reed‐Muller codes," Cryptology ePrint Archive, 2013.

[14] W. Lee, Y. S. Kim, Y. W. Lee, and J. S. No, "Post quantum signature scheme based on modified Reed‐Muller code pqsigRM," in First Round Submission to the NIST Postquantum Cryptography Call, Nov. 2017. [On-line]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Round-1-Submissions.

[15] A. -L. Horlemann-Trautmann, K. Marshall, and J. Rosenthal. "Considera tions for rank-based cryptosystems." 2016 IEEE International Symposium on Information Theory (ISIT), pp. 2544-2548, 2016.

[16] T. S. C. Lau, C. H. Tan, "New rank codes based encryption scheme using partial circulant matrices," Designs, Codes and Cryptography, 87(12), pp. 2979－2999, 2019.

〈저자소개〉

홍 지 훈 (Ji-Hoon Hong)

2018년 8월 : 서강대학교 수학과 졸업
2018년 8월~현재 : 서강대학교 수학과 석·박사과정
<관심분야> 부호론, 암호학, 인공지능

원 병 선 (Byung-Sun Won)

2017년 2월 : 서강대학교 수학과 석사
2017년 3월~현재 : 서강대학교 수학과 박사과정
<관심분야> 부호론, 암호학, 인공지능

김 종 락 (Jon-Lark Kim)

1993년 2월 : 포스텍 수학과 졸업
1997년 2월 : 서울대학교 수학과 석사
2002년 5월 : 일리노이 주립대학교 (시카고) 수학과 박사
2002년 8월~2005년 8월 : 네브라스카 대학교 연구조교수
2005년 8월~2012년 8월 : 루이빌 대학교 조교수, 부교수
2012년 9월~현재 : 서강대학교 수학과 교수
2016년 3월~현재 : 딥헬릭스(주) 대표
<관심분야> 부호론, 암호학, 정보이론, 인공지능