Layered ROLLO-I: Faster rank-metric code-based KEM using ideal LRPC codes ⋆

Chanki Kim[1], Young-Sik Kim[1], and Jong-Seon No[2]

[1] Dept. of Information and Communication Engineering, Chosun University,
Gwangju, Republic of Korea
{carisis,iamyskim}@chosun.ac.kr

[2] Dept. of Electrical and Computer Engineering, INMC, Seoul National University,
Seoul, Republic of Korea
jsno@snu.ac.kr

Abstract. For the fast cryptographic operation, we newly propose a key encapsulation mechanism (KEM) called layered ROLLO-I by using block-wise interleaved ideal LRPC (BII-LRPC) codes. By multiplying random polynomials by small-sized ideal LRPC codes, faster operation can be obtained with an additional key size. Finally, some parameters of the proposed algorithm are suggested and compared with that of the existing ROLLO-I scheme.

Keywords: Code-based cryptography · coding theory · low-rank parity-check (LRPC) codes · key encapsulation mechanism (KEM) · rank-metric codes · post-quantum cryptography (PQC).

1 Introduction

Rank-metric codes have drawn interest for an application of alternative cryptographic algorithms in upcoming quantum computing (QC) era, which is expected to be a new frontier to overcome the limitation of the classical computing system. However, there is a risk for disrupting secure communication gauranteed by hardness problem by the limitation of the conventional computing system. For example, a classical RSA algorithm is expected to be broken in a polynomial time by the Shor QC algorithm. Therefore, it is crucial to design new alternatives cryptosystem resilient also to the QC algorithm, called post-quantum cryptography (PQC) [1]. One of the well-known category is code-based cryptography [2]. For example, new cryptosystems using rank metric codes were proposed for the hardness for recovering the rank support problem. For example, ROLLO and McNie schemes based on the ideal low-rank parity-check (LRPC) codes was proposed by the advantage of the small key size and selected to second round in the NIST submission [3–6].

However, one of the challenges for rank-metric-based cryptosystem is that the decryption performance are not superior compared to the other lattice-based

⋆ This work is submitted to 'Korean Post-Quantum Cryptography Competition' (www.kpqc.or.kr).

candidates in the NIST submission [7]. Here, our proposal is to improve the decryption performance without loss of the security level, which can be possible by modifying the structure of the ideal LRPC codes.

1.1 Design rationale

In this proposal, we newly introduce a layered code structure by using blockwise interleaved ideal LRPC (BII-LRPC) codes. For the ideal LRPC (BII-LRPC) codes, a smaller codelength and rank distance is chosen for the faster cryptographic operation. In order to compensate the security level, the low-rank vector is multiplied into the two randomized polynomials, which interleaves a structure of the ideal LRPC codes and thus, the attacker cannot exploit the structural property for the attack scenario.

1.2 Advantages and limitations

In this proposal, the advantages and limitation of the proposed scheme can be compared with those of ROLLO-I. First, faster cryptographic operation from ROLLO-I can be obtained by adopting the layered structure. However, additional secret and public key size should be necessary for the information for the information of two randomized polynomials.

2 Preliminaries

In order to describe a rank-metric based cryptosystem, some mathematical notations are defined as follows. First, all the operation is based on the finite field $\mathbb{F}_{q^m}$ by the field extension of $\mathbb{F}_q$. For vector notations, let $\mathbf{v} = \{v_0, v_1, ..., v_{|\mathbf{v}|-1}\} \in \mathbb{F}_{q^m}^{|\mathbf{v}|}$ be an $|\mathbf{v}|$-tuple row vector, where v_i is the i-th component of $\mathbf{v}$. Let $[a, b] = \{i;\ a \leq i \leq b\}$, $a, b \in \mathbb{N}$, $[a] = [0, a]$ for the set of positive integers $\mathbb{N}$. Let $\mathbf{1}$ and $\mathbf{0}$ be all-one and all-zero vectors, respectively. For matrix notation, denote $\mathbf{O}$ and $\mathbf{I}$ as the $n \times n$ zero matrix and the $n \times n$ identity matrix, respectively. For a vector $\mathbf{v}$, a rank weight is defined as follows.

Definition 1 (Rank weight). *Let $\{1, \beta, \beta^2, ..., \beta^{m-1}\}$ be an m-dimensional basis over $\mathbb{F}_q$. Then, $\boldsymbol{v}$ can be represented as $m \times n$ matrix in $\boldsymbol{M}(\boldsymbol{v})$ $v_j \in \mathbb{F}_q^{m \times n}$ with element v_{ij} satisfying $v_j = \sum_{i=0}^{m-1} v_{ij}\beta^i$, Then, the rank weight $||\boldsymbol{v}||$ for $\boldsymbol{v}$ is defined by*

$$||\boldsymbol{v}|| = Rd(\boldsymbol{M}(\boldsymbol{v})) \tag{1}$$

where $Rd(\cdot)$ is the value of rank of the matrix. Then, an vector $\mathbf{v}$ is said to have a rank weight $||\mathbf{v}||$. For $||\mathbf{v}|| = d$, let $supp(\mathbf{v}) = F$ be a $\mathbb{F}_q$-subspace with rank weight d generated by $v_0, v_1, ..., v_{n-1}$ or equivalently, $supp(\mathbf{v}) =< v_0, v_1, ..., v_{n-1} >$. For two $\mathbb{F}_q$-subspaces $supp(\mathbf{e}) = E$ and $supp(\mathbf{v}) = F$ with rank weight r and d, let EF a product space with rank weight rd. In the next subsection, a concept of ideal LRPC codes is introduced.

2.1 Ideal LRPC Codes

For the code operation, the existing ideal LRPC codes are (n, k) $\mathbb{F}_{q^m}$-linear codes $\mathcal{C}$, where a codeword $\mathbf{c} \in \mathbb{C}$ is satisfied as

$$\mathbf{H}\mathbf{c}^\top = \mathbf{0}, ||\mathbf{c}|| \geq d, \tag{2}$$

for codelength n, dimension k, and an $(n-k)\times n$ parity check matrix (PCM) $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$. Also, denote a map $\varPhi$ from an n-tuple vectors $\mathbf{u} = (u_0, u_1, ..., u_{n-1}) \in \mathbb{F}_{q^m}^n$ to polynomial ring $\mathbf{u}(X) \in \mathbb{F}_{q^m}[X]/ < P >$ as

$$\varPhi : \mathbf{u} = (u_0, u_1, ..., u_{n-1}) \rightarrow \mathbf{u}(X) = \sum_{i=0}^{n-1} u_i X^i \tag{3}$$

for an n-degree polynomial $\mathbf{u}(X) \in \mathcal{F}_{q^m}[X]/ < P >$. For simplicity, we define $\mathbf{uv}$, $X^j\mathbf{v}$, and $P'\mathbf{v}$ in the modular operation as

$$\begin{aligned} \mathbf{uv} \bmod P = \varPhi^{-1}(\varPhi(\mathbf{u})\varPhi(\mathbf{v})) &= \sum_{i=0}^{n-1}\sum_{j=0}^{i} u_{i-j}v_j X^i \text{mod } P \\ X^j\mathbf{v} \bmod P = \varPhi^{-1}(X^j\varPhi(\mathbf{v})) &= \sum_{i=0}^{n-1} u_i X^{i+j} \text{mod } P \\ P'\mathbf{v} \bmod P = \varPhi^{-1}(P'\varPhi(\mathbf{v})) &= \sum_{i=0}^{n-1} P' u_i X^i \text{mod } P \end{aligned} \tag{4}$$

In this paper, we use two polynomials P and P^b as modulus for the proposed cryptosystem. Also, Proposition 1 is introduced.

Proposition 1. *For an n_1-tuple vector $\boldsymbol{u}$ and an n_2-tuple vector $\boldsymbol{v}$ with $n_1 \leq n$ and $n_2 \leq n$, extended bn-length vectors $\boldsymbol{u'} = [\boldsymbol{0}, \boldsymbol{u}]$ and $\boldsymbol{v'} = [\boldsymbol{0}, \boldsymbol{v}]$ can be defined. For an n-degree primitive polynomial P, we have*

$$\boldsymbol{uv} \mod P = (\boldsymbol{u'v'} \mod P^b) \mod P \tag{5}$$

Proof. By the map (3), each nonzero cofficeint of $\mathbf{u}(X)$ and $\mathbf{v}(X)$ is the same as that of $\mathbf{u}(X)'$ and $\mathbf{v}'(X)$. However, $\mathbf{u'v'} \mod P^b$ returns the unreduced modular polynomial because there is no higher degrees than P_b after multiplication. Therefore, $(\mathbf{u'v'} \mod P^b) \mod P$ returns the same results on $\mathbf{uv}modP$ and it concludes the proof.

For an l-degree primitive polynomial, denote the ideal matrix $\mathcal{I}(\mathbf{v}, P)$ with size $l \times l$ defined as

$$\mathcal{I}(\mathbf{v}, P) = \begin{pmatrix} \mathbf{v} \\ X\mathbf{v} \bmod P \\ ... \\ X^{l-1}\mathbf{v} \bmod P \end{pmatrix}. \tag{6}$$

Then, it is easy to check that the ideal condition is satisfied as $\mathbf{uv} = \mathbf{u}\mathcal{I}(\mathbf{v}, n) = \sum_{i=0}^{n-1} u_i v_i X^i = \mathbf{v}\mathcal{I}(\mathbf{u}, n) = \mathbf{vu}$. Using two ideal matrices, ideal LRPC codes are introduced as follows.

Definition 2 (Ideal LRPC codes [8]). *Let F be a $\mathbb{F}_q$-subspace set of n-tuple vector with rank weight d. For a vector $\boldsymbol{x}, \boldsymbol{y} \in F$, $(2n, n)$ ideal LRPC code $\mathcal{C}$ is defined by the $n \times 2n$ sized PCM $\boldsymbol{H} = [\mathcal{I}(\boldsymbol{x}, P)|\mathcal{I}(\boldsymbol{y}, P)]$ or equivalently, $\boldsymbol{H} = [I|\mathcal{I}(\boldsymbol{x}^{-1}\boldsymbol{y}, P)]$.*

For any $2n$-length vectors $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$ and syndrome $\mathbf{s} = \mathbf{e}_1\mathbf{x} + \mathbf{e}_2\mathbf{y}$. For the rank-metric codes, it is known to be hard to attack the cryptosystem using only for the syndrome $\mathbf{s}$ and PCM of the rank-metric codes. It is called as ideal rank syndrome decoding (I-RSD) algorithm as follows.

Definition 3 (Ideal-Rank syndrome decoding (I-RSD) problem). *For a vector $\boldsymbol{h}$ and syndrome vector $\boldsymbol{s}$, it is hard to find a vector $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2)$ lower than the rank weight w satisfying that $\boldsymbol{e}_1 + \boldsymbol{e}_2\boldsymbol{h} = \boldsymbol{s}$.*

In general, I-RSD problem is known to be zero-error probabilistic polynomial time (ZPP) [9]. For the selection of vector $\mathbf{h}$, the vector of $\mathbf{x}^{-1}\mathbf{y}$ can be used by the *indistinguishability* property of ideal LRPC codes as

Definition 4 (Indistinguishability over Ideal LRPC codes). *For the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ with small rank weight d, it is hard to distinguish between the uniformly sampled random vector $\boldsymbol{h}'$ and $\boldsymbol{x}^{-1}\boldsymbol{y} \mod P$.*

Thus, the rank-metric codes can be used to design the cryptosystem when $\mathbf{x}^{-1}\mathbf{y}$ is used as a public key (PK). In the following subsection, KEM algorithm using ideal LRPC codes is introduced.

2.2 Existing KEM algorithm Using Ideal LRPC Codes

In this subsection, KEM algorithm, security analysis, and decoding complexity analysis of the existing KEM scheme called ROLLO-I was introduced [8].

KEM Algorithm For an example of the ideal LRPC codes, ROLLO-I was proposed as key encapsulation mechanism (KEM), which returns the shared key from given security level as follows.

Definition 5 (ROLLO-I). *Suppose that shared key is generated between Alice and Bob. Then, ROLLO-I consists of the following three phases.*

1. *Key generation: Select two n-tuple vectors $\boldsymbol{x}, \boldsymbol{y} \in F$, where F denotes a set of n-tuple vectors with rank weight d. Then, Alice construct secret key (SK) as $(\boldsymbol{x}, \boldsymbol{y})$ and public key (PK) as $\boldsymbol{h} = \boldsymbol{x}^{-1}\boldsymbol{y} \mod P$.*
2. *Encryption: Bob select two vector $(\boldsymbol{e}_1, \boldsymbol{e}_2) \in E$, where E denotes a set of an n-tuple vector with rank weight r. Then, derive $\boldsymbol{c} = \boldsymbol{e}_1 + \boldsymbol{e}_2\boldsymbol{h} mod P$ using PK $\boldsymbol{h}$, where $Hash(\cdot)$ denotes a hash function known to Alice and Bob. Then, derive $K = Hash(E)$. Finally, send $\boldsymbol{c}$ to Alice and use $\boldsymbol{k}_1 = Hash(E)$ as a shared secret (SS).*
3. *Decryption: Using SK $\boldsymbol{x}$, derive $\boldsymbol{s} = \boldsymbol{x}\boldsymbol{c} = \boldsymbol{x}\boldsymbol{e}_1 + \boldsymbol{y}\boldsymbol{e}_2 \mod P$ and thus, we can find the basis E with the ideal RSR algorithm. Using the basis E, derive $Hash(E)$ for a hash function. Finally, the remaining procedure are divided as follows. Finally, Alice validate the correctness by checking the SS $\boldsymbol{k}_2 = Hash(E)$. Then, Alice can validate the correctness by checking $\boldsymbol{k}_1 = \boldsymbol{k}_2$.*

Security Analysis For the security analysis, indistinguishability and attacks using RSD were discussed [8], where ROLLO-I is indistinguishable against chosen plaintext attack (IND-CPA) using random oracle model (ROM).

For the security of I-RSD, it is known that the existing attacks for the ideal LRPC codes are classified as structural and general attacks. Here, the structural attack is designed to recover F from the structure of ideal LRPC codes in Construction 2, where the most efficient attack has the complexity of

$$S_S = \mathcal{O}(n^3 m^3 q^{d\lceil \frac{m}{2} \rceil - m - n}) \tag{7}$$

On the other hands, generic attack was introduced to recover the support E for the random ideal codes with rank weight r as

$$S_G = \mathcal{O}((nm)^3 q^{r\lceil \frac{m(n+1)}{2n} \rceil - m}) \tag{8}$$

Here, security level of the existing ROLLO-I and the proposed cryptosystems are represented by the minimum of the two logarithms from (7) and (8).

Computational Complexity Analysis In the existing ROLLO scheme, computational complexity is mainly based on the addition, multiplication, and inversion operation over polynomial ring modulus over P.

For the decoding of ideal LRPC codes, it is known that decryption complexity mainly depends on the rank support recovery (RSR), which consumes polynomial time for the decoding with small probability of decryption failure rate (DFR). For example, the computational complexity for syndrome space expansion algorithm in Fig. 8 of [8] was obtained as follows.

Definition 6 (Syndrome space expansion algorithm for RSR [8]). *For the two $\mathbb{F}_q$-spaces E with rank weight r and vector space F with rank weight d, syndrome space expansion algorithm can recover the space E from the product space EF with the complexity bounded by*

$$\mathcal{O}(r^2 d^3 m) \tag{9}$$

with DFR less than $q^{-(rd-(n-k))}$.

In the next section, new design and schemes using proposed BII-LRPC codes are proposed and the corresponding security and complexity analysis are presented.

3 Specification

For the propsed KEM, some notations with the new BII-LRPC codes are firstly introduced. Accordingly, the modified KEM will be constructed using the proposed BII-LRPC codes.

3.1 Notation

Firstly, the proposed design of BII-LRPC codes is represented as in Definition 7.

Definition 7 (BII-LRPC codes). *Let F be an $\mathbb{F}_q$-subspace with rank weight d and $\boldsymbol{x}, \boldsymbol{y} \in F$ be vectors. P_I and P_O denote a degree-$(b-1)$ inner and a degree-n outer polynomial in a polynomial ring $P_I \in \mathbb{F}_{q^m}[X]/<P>$ and $P_O \in \mathbb{F}_{q^m}[X]/<P^b>$ for a degree-$\frac{n}{b}$ primitive polynomial P. Then, let an n-length vector $\boldsymbol{h}' = [\boldsymbol{0}, P_I\boldsymbol{x}^{-1}\boldsymbol{y} \mod P]$ from length-$\frac{n}{b}$ vector $P_I\boldsymbol{x}^{-1}\boldsymbol{y} \mod P$. Then, a $(2n, n)$ ideal LRPC code $\mathcal{L}$ is defined by the $n \times 2n$ sized parity check matrix $\boldsymbol{H} = [\boldsymbol{I}|(\mathcal{I}(P_O\boldsymbol{h}'), P^b)]$.*

From a random n-degree polynomial P_O, the corresponding codeword $\mathbf{c}$ in Construction 7 is interleaved from the concatenated code for the small LRPC code, which is useful to maintain the security level by hiding small structural patterns. Note that the decoding of BII-LRPC codes can be conducted with the shorter codeword length and lower rank weight, which is advantageous for the decryption. On the other hand, attacker cannot benefit from the small-sized codes without the knowledge of P_I and P_O.

3.2 Specification of Layered ROLLO-I

Using the BII-LRPC codes, the KEM algorithms are represented by three parts; key generation, encapsulation, and decapsulation.

1. Key generation: Let F be a set of $\frac{n}{b}$-tuple vector with rank weight d. Alice selects two $\frac{n}{b}$-tuple random vectors $\mathbf{x}, \mathbf{y}$ satisfying $\mathbf{x}, \mathbf{y} \in F$. Also, denote random degree-$(b-1)$ primitive polynomial $P_I \in \mathbb{F}_{q^m}[X]/<P>$ and a degree-n $P_O, P_N \in \mathbb{F}_{q^m}[X]/<P^b>$, For $\mathbf{x}$ and $\mathbf{y}$, derive an $\frac{n}{b}$-tuple vector

$$\mathbf{z} = P_I\mathbf{x}^{-1}\mathbf{y} \mod P. \tag{10}$$

   Finally, Alice constructs a public key (PK) as

$$\begin{aligned} P_P &= P_O P_I && \mod P^b, \\ \mathbf{h} &= P_O\mathbf{z}' + P_N P && \mod P^b, \end{aligned}$$

   for $\mathbf{z}' = [\mathbf{0}, \mathbf{z}]$ and secret key (SK) as $\mathbf{x}, \mathbf{y}, P_I$, and P_O.

2. Encapsulation: Bob selects random two $\frac{n}{b}$-length vectors $(\mathbf{e}_1, \mathbf{e}_2) \in E$, where E denotes a set of $\frac{n}{b}$-tuple vector with maximum degree $\frac{n}{b} - b$ and rank weight r. Also, let length-n vectors $\mathbf{e}_1' = [\mathbf{0}, \mathbf{e}_1]$ annd $\mathbf{e}_2' = [\mathbf{0}, \mathbf{e}_2]$. Then, a ciphertext is generated by
$$\mathbf{c} = P_P \mathbf{e}_1' + \mathbf{h}\mathbf{e}_2' \mod P^b \tag{11}$$
using PK of P_P and $\mathbf{h}$. For a hash function Hash$(\cdot)$ known to Alice and Bob, $\mathbf{k}_1 = \text{Hash}(E)$ can be calculated. Finally, send $\mathbf{c}$ to Alice and use $\mathbf{k}_1$ as a shared secret (SS).
3. Decapsulation: Using SK of $\mathbf{x}$ and P_P, we have
$$\begin{aligned} \mathbf{c}' &= P_O^{-1}\mathbf{c} \mod P^b, \\ \mathbf{c}'' &= P_I^{-1}\{\mathbf{c}' \mod P\} \mod P, \\ \mathbf{x}\mathbf{c}'' &= \mathbf{x}\mathbf{e}_1 + \mathbf{y}\mathbf{e}_2 \mod P. \end{aligned}$$
Then, it is easy to check that $\mathbf{x}\mathbf{c}''$ can be decoded using the RSR algorithm, which can reconstruct E'. From $\mathbf{k}_2 = \text{Hash}(E')$, Alice can validate the correctness by checking $\mathbf{k}_1 = \mathbf{k}_2$.

Note that Appendix explains the detailed procedure of the proposed KEM scheme. For the key sizes, PK has the size of $\frac{2n\lceil\log_2(m)\rceil}{8}$ [Byte], which uses additional information of P_P with the bit size of $\frac{n\lceil\log_2 m\rceil}{8}$. In addition, SK use the key size of 3×40 [Bytes] by using the different random seed for the information of $\{\mathbf{x}, \mathbf{y}\}$, P_I, and P_O, where the conventional ROLLO-I uses 40 [Bytes] for the information of $\{\mathbf{x}, \mathbf{y}\}$. Note that SK size of the proposed KEM can be further reduced by using the same random seed for the $\{\mathbf{x}, \mathbf{y}\}$, P_I, and P_O. Lastly, CT size of proposed KEM amounts to $\frac{n\lceil\log_2 m\rceil}{8}$, which is the same of the existing ROLLO-I with the same parameters of n and m.

3.3 Parameter sets

For the parameter sets, instances in ROLLO-I for the second round of NIST PQC submission and the suggested instances for the proposed KEM with $b = 2$ are listed in Table 1. Also, the same field size m are fixed as the same as that of the existing parameter. The other parameters of the proposed KEM are chosen as the smallest ones with lower DFR compared to ROLLO-I.

4 Performance analysis

In this subsection, the performance improvements for the proposed KEM is shown using the simulation result. Here, we observed that additional ring operation for $b \geq 3$ requires lots of processing cycles to offset the advantage of reduced complexity in a RSR algorithm with low r and d. Therefore, we select parameter b as two.

Table 1. Suggested parameters of the existing ROLLO-I and proposed KEM with $b = 2$

Instances	q	n	m	r	d	b	DFR	PK size	SK size	CT size
ROLLO-I-128	2	83	67	7	8	1	2^{-27}	696	40	696
ROLLO-I-192	2	97	79	8	8	1	2^{-33}	958	40	958
ROLLO-I-256	2	113	97	9	9	1	2^{-32}	1371	40	1371
Proposed-128	2	74	67	3	2	2	2^{-31}	1240	120	620
Proposed-192	2	86	79	4	3	2	2^{-35}	1699	120	850
Proposed-256	2	106	97	5	3	2	2^{-38}	2571	120	1286

4.1 Description of platform

For the computer simulation, we modified the existing source codes of the rank-based cryptography (RBC) library [11], which generalizes ROLLO-I optimized implementations in the NIST submission. For the performance measurement, we use the processing cycle for the key generation, encapsulation, and decapsulation procedure. Also, we evaluated the number of the processing cycles on the workstation with the architecture of x86 12th Gen. Intel Core i9-12900K multi-core CPU with 32GB 4,800MHz DDR5 memory and operating system of Ubuntu Linux 20.04 LTS. Also, AVX-2 instructions are used for high-performance operations.

4.2 Performance in the reference implementation

The processing cycles for the suggested instances are represented as the worst total processing cycle value from 100 iterations and the results are listed in Table 2.

Table 2. The number of processing cycles of the ROLLO-I and proposed KEM for Parameters of Table 1

Instances	Keygen.	Encap.	Decap.	Total
ROLLO-I-128	6,019,622	574,711	8,287,089	14,881,422
ROLLO-I-192	4,388,835	577,348	7,955,763	12,922,035
ROLLO-I-256	8,361,499	672,956	10,878,644	19,903,099
Proposed-128	2,609,907	661,423	5,570,494	8,841,824
Proposed-192	2,921,813	755,759	5,253,698	8,931,270
Proposed-256	3,757,592	918,300	10,424,395	15,100,287

For the same security level, it is observed that the processing cycle is reduced compared to the ROLLO-I. Note that ROLLO-I-128 and Proposed-128 use a primitive polynomial P which is not a trinomial, which requires an extra cycle for the modular operation. By lowering d and r in the proposed instances, the number of processing cycles for the key generation and decapsulation are lowered, whereas that of encapsulation is increased. In addition, the size of PK becomes larger than the ROLLO-I due to P_P. For the higher security level, the gap between PK sizes for ROLLO-I and proposed KEM becomes larger.

Computational Complexity Analysis For measuring the computational complexity of the proposed KEM, advantages and drawbacks can be discussed respectively compared to the ROLLO-I. A computational advantage is mainly obtained by of key generation and decapsulation phases with an RSR algorithm which has a complexity of (9) and DFR less than $q^{-\left(rd-\frac{(n-k)}{b}\right)}$. However, the computational complexity can be increased by additional operations of the larger polynomial ring operation in $\mathbb{F}_{q^m}[X]/ < P^b >$. In addition, computational complexity of finding polynomial modular inverses of P_I and P_O in the decapsulation phase becomes larger for larger b.

5 Security

5.1 Security definition

In this subsection, the security analysis is presented for indistinguishability and attacks on RSD. In summary, the proposed scheme is also IND-CPA. Also, it is hard for an attacker to exploit the low-rank structure or reconstruct P_O and P_I without knowledge of SK and thus, the proposed KEM gaurantee the required security level with the lower decoding complexity.

5.2 Security strength categories

For the distinguisher problem between random $\mathbf{h}$ and $\mathbf{x}^{-1}\mathbf{y}$ in the proposed scheme, it is at least the same for the ideal LRPC codes as that of the existing one with the ROM. Actually, this proprosal is the same for the ideal LRPC codes as that of the existing one with the ROM. and thus, the proposed KEM is also IND-CPA.

5.3 Description and cost of known attack

For KEM attacks using RSD, an aim for attacking the proposed cryptosystem is to recover E. a possible attack can be divided into the two phases. For the first phase, an attacker tries to obtain the exact P_O, which is possible by the brute-force algorithm for guessing each coefficient $c_i = \alpha^i$ for $i \in [q^m - 1]$ in $P'_I = \sum_{i=0}^{b-1} c_i X^i$. Then, we have $P'_O = \frac{P_P}{P'_I} \mod P^b$. By guessing, the required complexity of finding the correct P_I is $\mathcal{O}(q^{(b-1)m})$. Note that it is hard to verify that P'_O is correct just using P_P because the structure of low-rank polynomial is not discovered by polynomial ring operation. Instead, the attacker should proceed the second phase for each P'_O.

In the second phase, the required complexity for the attack is obtained by either a structural or generic attack of the ideal LRPC code. Therefore, the corresponding security levels S_S and S_G are represented as

$$S_S = \log_2 \left\{ \left(\frac{n}{b}\right)^2 m^3 q^{(b-1)m+d\lceil \frac{m}{2} \rceil - m - \frac{n}{b}} \right\} \tag{12}$$

for structural attack,

$$S_G = \log_2 \left\{ \left(\frac{nm}{b}\right)^3 q^{(b-1)m+r\left\lceil \frac{m\left(\frac{n}{b}+1\right)}{\frac{2n}{b}} \right\rceil - m} \right\} \tag{13}$$

for generic attack.

Note that (12) and (13) are modified from the values inside parenthesis from the $\mathcal{O}$-notations in (7) and (8), where the values are identical when $b = 1$. Also, larger b can enhance the values of (12) and (13) and thus, lower r and d can be used for the proposed KEM. Based on the S_S and S_G, design parameters of the proposed KEMs with security levels 128, 192, and 256 are suggested as in Table 2.

6 Conclusion

In this proposal, we proposed a new design of BII-LRPC codes and KEM scheme based on the proposed codes. From the block-wise structure and the small rank weight of the proposed codes, it is shown that the proposed KEM can be implemented with a lower complexity via computer simulations. By numerical comparison, the proposed cryptosystem has the advantage of lower computational complexity compared with the existing ROLLO-I parameters.

The proposed block-wise design for the code-based cryptography is expected to be applicable to other rank-metric code-based cryptography. As a future work, the proposed KEMs will be generalized for the future PQC.

Acknowledgement

This work was supported in part by the Chosun University, in 2021, under a research fund, and in part by the Ministry of Science and ICT (MSIT) of the Korean Government (2021-0-00400, Development of Highly Efficient PQC Security and Performance Verification for Constrained Devices) under the Institute for Information and Communication Technology Promotion (IITP) Grant.

References

1. D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, pp. 188-194, 2017.
2. A. Kuznetsov, A. Kiian, M. Lutsenko, I. Chepurko and S. Kavun, "Code-based cryptosystems from NIST PQC," *IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Kyiv, Ukraine, 2018, pp. 282-287.
3. J. L. Kim, Y.-S. Kim, L. Galvez, M. J. Kim, and N. Lee, "McNie: A new code-based public-key encryption," arXiv:1812.05008v2, Jan. 2019
4. P. Gaborit, O. Ruatta, J. Schrek, and G. Zemor, "RankSign: An efficient signature algorithm based on the rank metric," *International Workshop on Post-Quantum Cryptography (PQCrypto)*, pp. 88-107, 2014.
5. Rollo, "ROLLO specification," Available on [online], https://pqc-rollo.org/documentation.html, 2020.
6. N. Aragon, O. Blazy, J. C. Deneuville, P. Gaborit, and G. Zemor, "Ouroboros: An efficient and provably secure KEM family," *IEEE Trans. Inf. Theo.*, vol. 68, no. 9, pp. 6233-6244, Sep. 2022.
7. J. Labalnche, L. Mortajine, O. Benchaalal, 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.
8. 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.*, vol. 65, no. 12, pp. 7697-7717, Dec. 2019.
9. P. gaborit and G. Zemor, "On the hardness of the decoding and the minimum distance problmes for rank codes," *IEEE Trans. on Inf. Theo.*, vol. 62, no. 12, pp. 7245-7252, Dec. 2016
10. N. Aragon, P. Gaborit, A. Hauteville, and J.-P. Tillich. "Improvement of generic attacks on the rank syndrome decoding problem," Oct. 2017, Available on [Online]. Available: https://hal.archives-ouvertes.fr/hal-01618464.
11. N. Aragon, S. Bettaieb, L. Bidoux, Y. Connan, J. Coulaud, P. Gaborit, and A. Kominiarz. "The rank-based cryptography library," *Code-Based Cryptography Workshop (CBCrypto)*, Munich:Germinay, Jun 21-22 2021.

A Appendix

For $\{\mathbf{c}' \mod P\} = \{P_O^{-1}\mathbf{c} \mod P^b\} \mod P$ in the decapsulation phase, we have

$$\begin{aligned}&\{P_O^{-1}\mathbf{c} \mod P^b\} \mod P\\ &= \{P_I\mathbf{e}_1' \mod P^b\} + \{P_O^{-1}\mathbf{h}'\mathbf{e}_2' \mod P^b\} \mod P.\end{aligned}$$

By Proposition 1, it is easy to check that the left part is modified as

$$\begin{aligned}&\{(\{P_I \mod P\}[\mathbf{0},\mathbf{e}_1]) \mod P^b\} \mod P\\ &= \{(P_I\mathbf{e}_1) \mod P\} \mod P.\end{aligned}$$

For the right part, we have

$$\begin{aligned}&\{P_O^{-1}\mathbf{h}'\mathbf{e}_2' \mod P^b\} \mod P\\ &= \{(P_N P + [\mathbf{0},\{P_I\mathbf{x}^{-1}\mathbf{y} \mod P\}])[\mathbf{0},\mathbf{e}_2]\\ &\quad \mod P^b\} \mod P.\end{aligned}$$

Here, term of $P_N P$ is for masking the structure $\{P_O^{-1}\mathbf{c} \mod P^b\}$, which can have low degree if $P_N = 0$ and the attacker can find the right P_O and P_I using only polynomial factoring. Also,

$$\begin{aligned}&= \{P_I\mathbf{x}\mathbf{y}^{-1} \mod P\}\{\mathbf{e}_2 \mod P\}\\ &= \{P_I\mathbf{x}\mathbf{y}^{-1}\mathbf{e}_2 \mod P\}.\end{aligned}$$

Then, we have

$$\begin{aligned}&\{\mathbf{c}' \mod P\} \mod P\\ &= \{(P_I\mathbf{e}_1') \mod P\} + \{P_I\mathbf{x}^{-1}\mathbf{y}\mathbf{e}_2 \mod P\} \qquad (14)\\ &= P_I\mathbf{e}_1 + P_I\mathbf{x}^{-1}\mathbf{y}\mathbf{e}_2 \mod P.\end{aligned}$$

From $P_I^{-1}\mathbf{c}' \mod P$, $\mathbf{e}_1 + \mathbf{x}^{-1}\mathbf{y}\mathbf{e}_2 \mod P$ is finally obtained, which can be decoded by multiplying $\mathbf{x}$ and the RSR algorithm.