# Two New Module-Code-Based KEMs with Rank Metric

Li-Ping Wang[1,2(✉)] and Jingwei Hu[3]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
wangliping@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

[3] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, Singapore
davidhu@ntu.edu.sg

**Abstract.** In this paper, we use a class of module codes to construct a suite of code-based public-key schemes—Piglet, which includes a new IND-CPA-secure public-key encryption scheme Piglet-1.CPAPKE and an IND-CCA-secure key encapsulation mechanism (KEM for short) Piglet-1.CCAKEM by applying the KEM variant of Fujisaki-Okamoto transform to Piglet-1.CPAPKE. We also put a new IND-CPA-secure KEM Piglet-2.CPAKEM into Piglet. Then, we present the parameters comparison between our schemes and some code-based NIST submissions. The results show that our schemes are good long-term-secure candidates for post-quantum cryptography.

**Keywords:** Code-based post-quantum cryptography · Rank syndrome decoding problem · Quasi-cyclic codes · Gabidulin codes · LRPC codes

## 1 Introduction

### 1.1 Background

Perceivable advances in quantum computers render Shor's quantum algorithm a threat to the widely used public key cryptosystems based on integer factoring and discrete logarithm problems [43]. As a consequence, NIST develops a post-quantum cryptography standardization project to solicit, evaluate, and standardize one or more quantum-resistant public cryptographic algorithms in recent years [38]. The cryptographic research community is stimulated by this initiation to construct practicable cryptographic systems that are secure against both quantum and classic computers, and can incorporate with existing communications protocols and networks. It is commonly thought that code-based cryptosystems can be resistant to quantum computing attack and so they are still becoming a hot topic even if NIST has ended the call.

J. Jang-Jaccard and F. Guo (Eds.): ACISP 2019, LNCS 11547, pp. 176–191, 2019.
https://doi.org/10.1007/978-3-030-21548-4_10

The first code-based cryptosystem was proposed by McEliece in 1978 by hiding a generator matrix of a Goppa code [33]. Another equivalent Niederreiter-type code-based scheme is constructed by scrambling a parity-check matrix of a Goppa code [35]. They are still secure under approximate parameters. However, the size of public keys in above schemes using Goppa codes is very huge. In order to reduce the size of public keys, LDPC (Low Density Parity Check) codes, convolutional codes, Gabidulin codes, Reed-Muller codes, and generalized Reed-Solomon codes were used to replace Goppa codes in the above cryptosystems framework, however, all were proven to be insecure [7,27,36,45,46].

As we all know, there are significant analogies between lattices and coding theory and the difference mainly consists in the use of different metrics (Euclidean metric for lattices, Hamming metric or rank metric for codes). Recently, inspired by the merits of lattices such as ideal rings and ring-LWE [2,32,39,40], diverse code-based public-key schemes such as RQC, HQC, BIKE, LOCKER, and Ouroboros-R, were proposed by using specific quasi-cyclic codes so that the size of public key is significantly reduced [1,4,5,17]. Those quasi-cyclic codes, i.e., we called one-dimensional module codes here, are also used in the many other code-based cryptosystems to advance compact key size [8,9,34]. However, the added quasi-cyclic structure may be exploited to initiate an algebraic attack and therefore brings about less confidence in the underlying security [18,41,42].

In lattice-based public key cryptosystems, Kyber which employs module lattices was proposed to thwart attacks from exploiting the algebraic structure of cyclotomic ideal lattices [11–15]. However, in code-based cryptosystems, there are no similar schemes.

In this paper, motivated by Kyber based on module lattices, we use the concept of module codes to redefine quasi-cyclic codes and propose an alternative assumption that rank module syndrome decoding (RMSD for short) problem is difficult so that our schemes are distinguishable from those so-called quasi-cyclic-code-based cryptosystems. It is worth mentioning that a handful of cryptosystems using rank codes exist in literature due to nice properties of rank metric such as RQC, Ouroboros-R, GPT's variant [31]. Therefore, based on the hardness of RMSD problem, we construct a suite of code-based public-key schemes—Piglet, which includes a new IND-CPA-secure public-key encryption scheme Piglet-1.CPAPKE and an IND-CCA-secure key encapsulation mechanism (KEM for short) Piglet-1.CCAKEM by applying the KEM variant of Fujisaki-Okamoto transform to Piglet-1.CPAPKE. We also put a new IND-CPA-secure KEM Piglet-2.CPAKEM into this suite. Then, we present the parameters comparison between our schemes and some code-based NIST submissions. The results show that our schemes are good long-term-secure candidates for post-quantum cryptography.

### 1.2 Our Contribution and Techniques

In this paper, the main contribution is that we propose a semantically secure public-key encryption scheme Piglet-1.CPAPKE and a new IND-CPA-secure

KEM Piglet-2.CPAKEM based on the hardness of rank module syndrome decoding problem. We believe that our schemes would be good candidates for post-quantum public-key cryptosystems with long-term security. The following are some advantages:

**Security.** The security of our schemes is established on the hardness of RMSD problem with two dimensions, while current code-based schemes are built upon rank quasi-cyclic syndrome decoding (RQCSD) problem which is RMSD problem with one dimension. In [42], the authors used the quasi-cyclic algebraic structure to propose a generic decoding attack. It shows that higher dimension of a module code can diminish the impact that possible attacks introduce. Furthermore, it cannot be excluded that some fatal attacks which exploits the quasi-cyclic structure embedded in the code might be proposed in the future. Therefore, we use module codes with two dimensions to construct new schemes, which would be good candidates for post-quantum public-key cryptosystems with long-term security.

**More Plaintext Bits.** In kyber, the size of plaintext is fixed to 256 bits, however, in our schemes, the size of plaintext depends on the extension degree of the finite field and the dimension of the auxiliary code in our scheme Piglet-1. So the sizes of plaintexts in Piglet-1 in 128, 192, and 256 bits security level are 267, 447, and 447 bits, respectively.

**Efficiency.** Although the operations in our schemes are implemented in large finite fields, it is also efficient in practice.

**Decoding Failure.** There is no decoding failure in Piglet-1.CPAPKE and Piglet-1.CCAKEM since we use the decoding algorithm for Gabidulin codes. As to Piglet-2.CPAKEM, the decoding failure rate is extremely low and tolerable.

### 1.3 Road Map

The rest of the paper is organized as follows. Section 2 introduces some basic concepts and some results needed in our paper. In Sect. 3, we describe a difficult problem on which the security of our schemes is based. In Sect. 4, we propose Piglet-1.CPAPKE and give the security proof. Then, we apply Fujisaki-Okamoto transform to Piglet-1.CPAPKE and then construct Piglet-1.CCAKEM with CCA security. Next, we give three parameter sets achieving 128, 192 and 256 bits of security, and make comparison on parameters between our schemes and some NIST candidates. In Sect. 5, we present Piglet-2.CPAKEM, whose session key is the hash value of error vectors without encrypting plaintexts. In Sect. 6, we provide analysis on the existing attacks to our schemes. Finally, Sect. 7 is devoted to our conclusions.

## 2 Preliminaries

### 2.1 Results on Rank Codes

We represent vectors by lower-case bold letters and matrices by upper-case letters, and all vectors will be assumed to be row vectors. Let $\mathbb{F}_{q^m}^n$ be an $n$-

dimensional vector space over a finite field $\mathbb{F}_{q^m}$ where $q$ is a prime power, and $n$, $m$ are positive integers.

Let $\beta = \{\beta_1, \ldots, \beta_m\}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Let $\mathcal{F}_i$ be the map from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q$ where $\mathcal{F}_i(u)$ is the $i$-th coordinate of an element $u \in \mathbb{F}_{q^m}$ in the basis representation with $\beta$. To any $\mathbf{u} = (u_1, \ldots, u_n)$ in $\mathbb{F}_{q^m}^n$, we associate the $m \times n$ matrix $(\mathcal{F}_i(u_j))_{1 \le i \le m, 1 \le j \le n}$ over $\mathbb{F}_q$. The rank weight of a vector $\mathbf{u}$ can be defined as the rank of its associated matrix, denoted by $w_R(\mathbf{u})$. We refer to [29] for more details on rank codes.

For integers $1 \le k \le n$, an $[n, k]$ linear rank code $C$ over $\mathbb{F}_{q^m}$ is a subspace of dimension $k$ of $\mathbb{F}_{q^m}^n$ embedded with the rank metric. The minimum rank distance of the code $C$, denoted by $d_R(C)$, is the minimum rank weight of the non-zero codewords in $C$. A $k \times n$ matrix is called a generator matrix of $C$ if its rows span the code. The dual code of $C$ is the orthogonal complement of the subspace $C$ of $\mathbb{F}_{q^m}^n$, denoted by $C^\perp$. A parity-check matrix $H$ for a linear code $C$ is a generator matrix for $C^\perp$.

For any vector $\mathbf{x} = (x_1, \ldots, x_n)$ in $\mathbb{F}_{q^m}^n$, the support of $\mathbf{x}$, denoted by $\mathrm{Supp}(\mathbf{x})$, is the $\mathbb{F}_q$-linear subspace of $\mathbb{F}_{q^m}$ spanned by the coordinates of $\mathbf{x}$, that is, $\mathrm{Supp}(\mathbf{x}) = <x_1, \ldots, x_n>_{\mathbb{F}_q}$. So we have $w_R(\mathbf{x}) = \dim(\mathrm{Supp}(\mathbf{x}))$.

Let $r$ be a positive integer and a vector $\mathbf{v} = (v_1, \ldots, v_r) \in \mathbb{F}_{q^m}^r$. The circulant matrix $\mathrm{rot}(\mathbf{v})$ induced by $\mathbf{v}$ is defined as follows:

$$\mathrm{rot}(\mathbf{v}) = \begin{pmatrix} v_1 & v_r & \ldots & v_2 \\ v_2 & v_1 & \ldots & v_3 \\ \vdots & \vdots & \ddots & \vdots \\ v_r & v_{r-1} & \ldots & v_1 \end{pmatrix} \in \mathbb{F}_{q^m}^{r \times r},$$

where $\mathbb{F}_{q^m}^{r \times r}$ denotes the set of all matrices of size $r \times r$ over $\mathbb{F}_{q^m}$.

For any two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{F}_{q^m}^r$, $\mathbf{u} \cdot \mathbf{v}$ can be expressed to vector-matrix product as follows.

$$\mathbf{u} \cdot \mathbf{v} = \mathbf{u} \times \mathrm{rot}(\mathbf{v})^T = (\mathrm{rot}(\mathbf{u}) \times \mathbf{v}^T)^T = \mathbf{v} \times \mathrm{rot}(\mathbf{u})^T = \mathbf{v} \cdot \mathbf{u}.$$

Let $\mathcal{R} = \mathbb{F}_{q^m}[x]/(x^r - 1)$. Then $\mathbb{F}_{q^m}^r$ is an $\mathbb{F}_{q^m}$-algebra isomorphic to $\mathcal{R}$ defined by $(v_1, v_2, \ldots, v_r) \mapsto \sum_{i=1}^r v_i x^i$.

**Definition 1.** *An $[n, k]$-linear block code $\mathcal{C} \in \mathbb{F}_{q^m}^n$ is a quasi-cyclic with index $s$ if for any $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_s) \in \mathcal{C}$ with $s|n$, the vector obtained after applying a simultaneous circulant shift to every block $\mathbf{c}_1, \ldots, \mathbf{c}_s$ is also a codeword.*

When $n = sr$, it is convenient to have parity-check matrices composed by $r \times r$ circulant blocks. In this paper, we use another viewpoint to describe quasi-cyclic codes so that it is clear to distinguish the quasi-cyclic codes used in our schemes from the many other quasi-cyclic-code-based cryptosystems.

**Definition 2.** *An $[n, k]$-linear block code $\mathcal{C}$ over $\mathcal{R}$ is called an $\mathcal{R}$-module code if $C$ is a $k$-dimensional $\mathcal{R}$-submodule of $\mathcal{R}^n$.*

*Remark 1.* 1. The module code $C$ over $\mathcal{R}$ is also quasi-cyclic over $\mathbb{F}_{q^m}$ since $(xc_1, \cdots, xc_n)$ is also a codeword of $C$ for any $(c_1, \cdots, c_n) \in C$.

2. The quasi-cyclic codes over $\mathbb{F}_{q^m}$ used in RQC, HQC, Ouroboros-R, BIKE, etc, are module codes over $\mathcal{R}$ with dimension $k = 1$.
3. The module codes are reduced to a general linear cyclic code if $n = 1$.
4. The module codes are a general linear code if $r = 1$.

**Definition 3.** *A systematic* $[n, k]$ *module code over* $\mathcal{R}$ *has the form of a parity-check matrix as* $H = (I|A)$, *where* $A$ *is an* $(n-k) \times k$ *matrix over* $\mathcal{R}$.

For example, in our schemes we use a systematic $[4, 2]$ module code over $\mathcal{R}$ and $A$ has the form $\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$, where $a_{ij} \in \mathcal{R}$, $i = 1, 2, j = 1, 2$, and so $a_{ij}$ can also be seen a circulant matrix over $\mathbb{F}_{q^m}$. In fact, the systematic cyclic codes used in RQC, HQC, Ouroboros-R, BIKE are $[2, 1]$ module codes over $\mathcal{R}$ and have such forms $A = (a)$, where $a \in \mathcal{R}$.

Next, we generalize the rank weight of a vector in $\mathbb{F}_{q^m}^n$ to $\mathcal{R}^n$.

**Definition 4.** *Let* $\mathbf{v} = (v_1, \ldots, v_n) \in \mathcal{R}^n$, *where* $v_i = \sum_{j=0}^{r-1} a_{ij} x^j$ *for* $1 \leq i \leq n$. *The support of* $\mathbf{v}$ *is defined by* $Supp(\mathbf{v}) = \langle a_{1,0}, \ldots, a_{1,r-1}, \ldots, a_{n,0}, \ldots, a_{n,r-1} \rangle_{\mathbb{F}_q}$. *The rank weight of* $\mathbf{v}$ *is defined to be the dimension of the support of* $\mathbf{v}$, *also denoted by* $w_R(\mathbf{v})$.

### 2.2 Gabidulin Codes and Their Decoding Technique

Gabidulin codes were introduced by Gabidulin in [20] and independently by Delsarte in [16]. They exploit linearized polynomials instead of regular ones, which was introduced in [37].

A $q$-linearized polynomial over $\mathbb{F}_{q^m}$ is defined to be a polynomial of the form

$$L(x) = \sum_{i=0}^{d} a_i x^{q^i}, a_i \in \mathbb{F}_{q^m}, a_d \neq 0$$

where $d$ is called the $q$-degree of $f(x)$, denoted by $\deg_q(f(x))$. Denote the set of all $q$-linearized polynomials over $\mathbb{F}_{q^m}$ by $\mathcal{L}_q(x, \mathbb{F}_{q^m})$.

Let $g_1, \ldots, g_n \in \mathbb{F}_{q^m}$ be linearly independent over $\mathbb{F}_q$ and the Gabidulin code $\mathcal{G}$ is defined by

$$\mathcal{G} = \{(L(g_1), \ldots, L(g_n)) \in \mathbb{F}_{q^m}^n \,|\, L(x) \in \mathcal{L}_q(x, \mathbb{F}_{q^m}) \text{ and } \deg_q(L(x)) < k\}.$$

The Gabidulin code $\mathcal{G}$ with length $n$ has dimension $k$ over $\mathbb{F}_{q^m}$ and the generator matrix of $\mathcal{G}$ is

$$G = \begin{pmatrix} g_1 & \cdots & g_n \\ g_1^q & \cdots & g_n^q \\ \vdots & \ddots & \vdots \\ g_1^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{pmatrix}. \quad (1)$$

The minimum rank distance of Gabidulin code $\mathcal{G}$ is $n-k+1$, and so it can efficiently decode up to $\frac{n-k}{2}$ rank errors [20]. The decoding algorithm employed in our scheme was proposed in [44], which is the generalization of Berlekamp-Massey algorithm and its computational complexity is $O(n^2)$, see details in [44].

### 2.3 Low Rank Parity Check Codes and Their Decoding Algorithm

The Low Rank Parity Check (LRPC) codes have been introduced in [24]. LRPC codes are widely used in code-based cryptosystems because they have a weak algebraic structure and efficient decoding algorithms.

An LRPC code of rank $d$, length $n$ and dimension $k$ is an $[n,k]$-linear block code over $\mathbb{F}_{q^m}$ that has its parity-check matrix $H = (h_{ij})_{1 \le i \le n-k, 1 \le j \le n}$ such that the dimension of the subspace spanned by all $h_{ij}$ is $d$.

The rank syndrome decoding for an LRPC code is that given a parity-check matrix $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ of an LRPC code of rank $d$ and a syndrome $\mathbf{s} \in \mathbb{F}_{q^m}^{n-k}$, the goal is to find a vector $\mathbf{x} \in \mathbb{F}_{q^m}^{n}$ with $w_R(\mathbf{x}) \le r$ such that $H\mathbf{x}^T = \mathbf{s}^T$.

In fact, what we want in Piglet-2.CPAKEM is just to recover the subspace $E$ spanned by $\mathbf{x}$ instead of $\mathbf{x}$, which is called rank support recovery problem. The rank support recovery algorithm was provided in [17], which combines the general decoding algorithm of LRPC codes in [21] and a tweak of the improved algorithm in [3]. The following is the rank support recovery algorithm in detail (RS-Recover for short).

In the following algorithm, $S$ and $E$ are the vector spaces generated by the coordinates of the syndrome $\mathbf{s} = (s_1, \cdots, s_{n-k})$ and of the vector $\mathbf{x}$, respectively. $S_i$ is defined by $S_i = F_i^{-1}.S = \langle F_1^{-1}s_1, F_1^{-1}s_2, \cdots, F_d^{-1}s_{n-k}\rangle$, with $F_i$ an element of a basis of $H$, and $S_{ij} = S_i \cap S_j$.

```
RS-recover($H, \mathbf{s}, r$)

Input: $H = \langle F_1, F_1, \dots F_d \rangle$, $\mathbf{s} = (s_1, \dots, s_{n-k})$, $r$ (the dimension of $E$)
Output: The vector space $E$
// Part 1: Compute the vector space $E.F$
Compute $S = \langle s_1, \dots, s_{n-k} \rangle$
Precompute every $S_i$ for $i = 1$ to $d$
Precompute every $S_{i,i+1}$ for $i = 1$ to $d-1$
for $i$ from 1 to $d-2$ do
    tmp $\leftarrow S + F.(S_{i,i+1} \oplus S_{i_1,i+2} \oplus S_{i,i+2})$
    if dim($tmp$) $\le rd$ then
      $S \leftarrow tmp$
    end
end
// Part 2: Recover the vector space $E$
$E \leftarrow F_1^{-1}.S \cap \dots \cap F_d^{-1}.S$
return $E$
```

The above algorithm will probably fail in some cases and the decode failure probability is given in Ouroboros-R [17].

**Proposition 1.** *The probability of failure of the above algorithm is* $\max(q^{(2-r)(d-2)} \times q^{-(n-k-rd+1)}, q^{-2(n-k-rd+2)})$*, where* $r$ *is the rank weight of the error vector.*

## 3 Difficult Problems for Code-Based Cryptography

In this section, we describe some difficult problems which are used in code-based cryptography. In particular, we introduce a difficult problem, i.e., rank module syndrome decoding (RMSD for short) problem, which is the security assumption for our schemes.

**Definition 5 (Rank Syndrome Decoding (RSD for short) Problem).** *Given a parity-check matrix* $H = (I_{n-k}|\, A_{(n-k)\times k}) \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *of a random linear code, and* $\mathbf{y} \in \mathbb{F}_{q^m}^{n-k}$*, the goal is to find* $\mathbf{x} \in \mathbb{F}_{q^m}^{n}$ *with* $w_R(\mathbf{x}) \leq w$ *such that* $H\mathbf{x}^T = \mathbf{y}^T$*.*

The RSD problem has recently been proven difficult with a probabilistic reduction to the Hamming setting in [22]. As we all know, syndrome decoding problem in Hamming metric is NP-hard [10]. Most of QC-code-based cryptosystems in rank metric are built upon the following difficult problem.

**Definition 6 (Rank Quasi-Cyclic Syndrome Decoding (RQCSD) Problem).** *Given a parity-check matrix* $H = (I_{n-1}|A_{(n-1)\times 1}) \in \mathcal{R}^{(n-1)\times n}$ *of a systematic random module code over* $\mathcal{R}$ *and a syndrome* $\mathbf{y} \in \mathcal{R}^{n-1}$*, to find a word* $\mathbf{x} \in \mathcal{R}^n$ *with* $\omega_R(\mathbf{x}) \leq w$ *such that* $\mathbf{y}^T = H\mathbf{x}^T$*.*

RQCSD problem is not proven to be NP-hard, however, the size of public-key is much shorter of variant code-based cryptosystems constructed on this problem such as RQC, Ouroboros-R, LOCKER. As for Hamming metric, one use quasi-cyclic syndrome decoding (QCSD for short) problem as security assumption [8], [34]. We give a new difficult problem as follows:

**Definition 7 (Rank Module Syndrome Decoding (RMSD) Problem).** *Given a parity-check matrix* $H = (I_{n-k}|\, A_{(n-k)\times k}) \in \mathcal{R}^{(n-k)\times n}$ *of a systematic random module code over* $\mathcal{R}$ *and a syndrome* $\mathbf{y} \in \mathcal{R}^{n-k}$*, to find a word* $\mathbf{x} \in \mathcal{R}^n$ *with* $\omega_R(\mathbf{x}) \leq w$ *such that* $\mathbf{y}^T = H\mathbf{x}^T$*.*

*Simply denote the above problem by the* $(n, k, w, r)$*-RMSD problem over* $\mathcal{R}$*.*

*Remark 2.* 1. If $k = 1$, the $(n, k, w, r)$-RMSD problem over $\mathcal{R}$ is the RQCSD problem, which is used in some NIST submissions such as RQC, Ouroboros-R, LOCKER. The result holds for the Hamming metic.
2. If $r = 1$, the $(n, k, w, r)$-RMSD problem over $\mathcal{R}$ is the usual RSD problem over $\mathbb{F}_{q^m}$.
3. The RSD problem is proved to be NP-hard [22], however, the RQCSD and the RMSD problem are still not yet proven to be NP-hard. Furthermore, smaller $k$ implies more algebraic structure makes the scheme potentially susceptible to more avenues of attacks. Therefore, the security of RMSD-based schemes ($k \geq 2$ by default) is supposed to be in between RSD and RQCSD based cryptosystems.

The above problem is also called the search version of RMSD problem. We also give the definition of the decisional rank module syndrome decoding problem (DRMSD). Since the best known attacks on the $(n, k, w, r)$-DRMSD problem consist in solving the same instance of the $(n, k, w, r)$-RMSD problem, we make the assumption that the $(n, k, w, r)$-DRMSD problem is difficult.

**Definition 8.** *Given input* $(H, \mathbf{y}) \in \mathcal{R}^{(n-k)\times n} \times \mathcal{R}^{n-k}$*, the decisional RMSD problem asks to decide with non-negligible advantage whether* $(H, \mathbf{y}^T)$ *came from the RMSD distribution or the uniform distribution over* $\mathcal{R}^{(n-k)\times n} \times \mathcal{R}^{n-k}$*.*

The above problem is simply denoted as $(n, k, w, r)$-DRMSD problem.

## 4 Piglet-1: A New Module-Code-Based Public-Key Scheme

### 4.1 Piglet-1.CPAPKE

In this subsection, we first present a new IND-CPA-secure public-key encryption, i.e., Piglet-1.CPAPKE, in which XOF(·) denotes an extendable output function and $S := \mathrm{XOF}(x)$ denotes the output of the function is distributed uniformly over a set $S$ while $x$ is as input.

In this scheme, we exploit an $[r, l]$-Gabidulin code $\mathcal{G}$, since the Gabidulin code is a unique rank code family with an efficient decoding algorithm. The minimum distance is $r - l + 1$ and so one can efficiently decode up to $\frac{r-l}{2}$ rank errors. The plaintext $\mathbf{m}$ is chosen from the plaintext space $\mathbb{F}_{q^m}^l$.

Piglet-1.CPAPKE.keyGen(): key generation

1. $\rho \xleftarrow{\$} \{0, 1\}^{256}$, $\sigma \xleftarrow{\$} \{0, 1\}^{320}$
2. $H \in \mathcal{R}^{k\times k} := \mathrm{XOF}(\rho)$
3. $(\mathbf{x}, \mathbf{y}) \in \mathcal{R}^k \times \mathcal{R}^k := \mathrm{XOF}(\sigma)$ with $w_R(\mathbf{x}) = w_R(\mathbf{y}) = w$
4. $\mathbf{s} := \mathbf{x}H + \mathbf{y}$
5. return $(pk := (H, \mathbf{s}), sk := \mathbf{x})$

Piglet-1.CPAPKE.Enc($\rho, \mathbf{s}, \mathbf{m} \in \mathbb{F}_{q^m}^l$): encryption

1. $\tau \xleftarrow{\$} \{0, 1\}^{320}$
2. $H \in \mathcal{R}^{k\times k} := \mathrm{XOF}(\rho)$
3. $(\mathbf{r}, \mathbf{e}, \mathbf{e}') \in \mathcal{R}^k \times \mathcal{R}^k \times \mathcal{R} := \mathrm{XOF}(\tau)$ with $w_R(\mathbf{r}) = w_R(\mathbf{e}) = w_R(\mathbf{e}') = w_e$
4. $\mathbf{u} := H\mathbf{r}^T + \mathbf{e}^T$
5. $\mathbf{v} := \mathbf{s}\mathbf{r}^T + \mathbf{e}' + \mathbf{m}G$, where $G$ is an $l \times r$ generator matrix over $\mathbb{F}_{q^m}$ of a Gabidulin code $\mathcal{G}$.
6. return a ciphertext pair $\mathbf{c} := (\mathbf{u}, \mathbf{v})$

Piglet-1.CPAPKE.Dec($sk = \mathbf{x}, \mathbf{c} = (\mathbf{u}, \mathbf{v})$): decryption

1. Compute $\mathbf{v} - \mathbf{x}\mathbf{u} := \mathbf{m}G + \mathbf{y}\mathbf{r}^T + \mathbf{e}' - \mathbf{x}\mathbf{e}^T$

2. $\mathbf{m} := \mathcal{D}_G(\mathbf{v} - \mathbf{x}\mathbf{u})$, where $\mathcal{D}_G(\cdot)$ is a decoding algorithm for the Gabidulin code $\mathcal{G}$.

*Remark 3.* 1. The secret key $\mathbf{x}$ and $\mathbf{y}$ share the same support including 1 with dimension $w$. The $\mathbf{r}$, $\mathbf{e}$ and $\mathbf{e}'$ share the same support with dimension $w_e$. So that the rank weight of overall error vector $\mathbf{y}\mathbf{r}^T + \mathbf{e}' - \mathbf{x}\mathbf{e}^T$ is less than or equal to $ww_e$.
2. The plaintext $\mathbf{m}$ can be obtained by decoding algorithm of the Gabidulin code $\mathcal{G}$ if $w_R(\mathbf{y}\mathbf{r}^T + \mathbf{e}' - \mathbf{x}\mathbf{e}^T) = ww_e \leq \frac{r-l}{2}$.

### 4.2 Proof of Security

In this subsection, we show that Piglet-1.CPAPKE is IND-CPA secure under the RMSD hardness assumption.

**Theorem 1.** *For any adversary $A$, there exists an adversary $B$ such that* $Adv^{CPA}_{Piglet\text{-}1.CPAPKE}(A) \leq Adv^{DRMSD}_{2k,k,w,r}(B) + Adv^{DRMSD}_{2k+1,k,w_e,r}(B)$.

**Proof.** Let $A$ be an adversary that is executed in the IND-CPA security experiment which we call game $G_1$, i.e.,

$$\mathrm{Adv}^{\mathrm{CPA}}_{\mathrm{Piglet\text{-}1.CPAPKE}}(A) = |Pr[b = b' \text{ in game } G_1] - 1/2|,$$

In game $G_2$, the view of $\mathbf{s} = \mathbf{x}H + \mathbf{y}$ generated in KeyGen is replaced by a uniform random matrix. It is possible to verify that there exists an adversary B with the same running time as that of A such that

$$|Pr[b = b' \text{ in game } G_1] - Pr[b = b' \text{ in game } G_2]| \leq \mathrm{Adv}^{\mathrm{DMRSD}}_{2k,k,w,r}(B),$$

since $(I\ H^T)\begin{pmatrix}\mathbf{y}^T \\ \mathbf{x}^T\end{pmatrix} = \mathbf{s}^T$, where $(I\,H^T)$ is a systematic parity-check matrix of a module code over $\mathcal{R}$ while $\mathbf{x}$ and $\mathbf{y}$ are drawn randomly with low rank weight $w$.

In game $G_3$, the values of $\mathbf{u} = H\mathbf{r}^T + \mathbf{e}^T$ and $\mathbf{v} = \mathbf{s}\mathbf{r}^T + \mathbf{e}' + \mathbf{m}G$ used in the generation of the challenge ciphertext are simultaneously substituted with uniform random values. Again, there exists an adversary B with the same running time as that of A such that

$$|Pr[b = b' \text{ in game } G_2] - Pr[b = b' \text{ in game } G_3]| \leq \mathrm{Adv}^{\mathrm{DMRSD}}_{2k+1,k,w_e,r}(B),$$

since $\begin{pmatrix} I_k & & H \\ & I_1 & \mathbf{s}\end{pmatrix}\begin{pmatrix}\mathbf{e}^T \\ \mathbf{e}' \\ \mathbf{r}^T\end{pmatrix} = \begin{pmatrix}\mathbf{u} \\ \mathbf{v} - \mathbf{m}G\end{pmatrix}$, where $\begin{pmatrix} I_k & & H \\ & I_1 & \mathbf{s}\end{pmatrix}$ is a systematic parity-check matrix of a module code while $H$, $\mathbf{s}$ are uniform and $\mathbf{r}, \mathbf{e}, \mathbf{e}'$ are drawn randomly with low rank weight $w_e$.

Note that in game $G_3$, the value $\mathbf{v}$ from the challenge ciphertext is independent of $b$ and therefore $Pr[b = b' \text{ in game } G_3] = \frac{1}{2} + \epsilon$, in which $\epsilon$ is arbitrarily small. We build a sequence of games allowing a simulator to transform a ciphertext of a message $\mathbf{m}_0$ to a ciphertext of a message $\mathbf{m}_1$. Hence the result is required. □

### 4.3 Piglet-1.CCAKEM: A New IND-CCA-Secure KEM

In this subsection, let $G : \{0,1\}^* \rightarrow \{0,1\}^{3\times 256}$ and $H : \{0,1\}^* \rightarrow \{0,1\}^{2\times 256}$ be hash functions, and $z$ is a random, secret seed. Then, we apply the KEM variant of Fujisaki-Okamoto transform to Piglet-1.CPAPKE to construct an IND-CCA-secure KEM, i.e., Piglet-1.CCAKEM when the hash functions $G$ and $H$ are modeled random oracle.

Piglet-1.CCAKEM.Keygen() is the same as Piglet-1.CPAPKE. Keygen()
Piglet-1.CCAKEM.Encaps(pk = $(\rho, \mathbf{s})$)

```
$\mathbf{m} \leftarrow \mathbb{F}_{q^m}^l$
$(\hat{K}, \sigma, d) := G(pk, \mathbf{m})$
$(\mathbf{u}, \mathbf{v}) :=$ Piglet-1.CPAPKE.Enc($(\rho, \mathbf{s}), \mathbf{m}; \sigma$)
$\mathbf{c} := (\mathbf{u}, \mathbf{v}, d)$
$K := H(\hat{K}, \mathbf{c})$
return$(\mathbf{c}, K)$
```

Piglet-1.CCAKEM.Decaps(sk = $(\mathbf{x}, z, \rho, \mathbf{s}), \mathbf{c} = (\mathbf{u}, \mathbf{v}, d)$)

```
$\mathbf{m}' :=$ Piglet-1.CPAKEM.Dec($\mathbf{x}, (\mathbf{u}, \mathbf{v})$)
$(\hat{K}', \sigma', d') := G(pk, \mathbf{m}')$
$(\mathbf{u}', \mathbf{v}') :=$ Piglet-1.CPAKEM.Enc($(\rho, \mathbf{s}), \mathbf{m}'; \sigma'$)
if $(\mathbf{u}', \mathbf{v}', d') = (\mathbf{u}, \mathbf{v}, d)$ then
    return $K := H(\hat{K}', \mathbf{c})$
else
    return $K := H(z, \mathbf{c})$
end if
```

### 4.4 Parameter Sets

In this subsection, we give three sets of parameters for Piglet-1.CCAKEM, achieving 128, 192 and 256 bits of security, respectively.

First we choose the dimension of the module code used in our schemes $k = 2$ so that the size of public key is as small as possible. In this case, we consider $1 \in \mathrm{Supp}(\mathbf{x}, \mathbf{y})$, since finding a small weight codeword of weight $w$ with support containing 1 is harder than finding a small weight codeword of $w-1$. Therefore, the security of the $(2k, k, w, r)$-RMSD over $\mathcal{R}$ in our scheme can be reduced to decoding $[4r, 2r]$-linear codes over $\mathbb{F}_{q^m}$ with rank weight $w-1$. The security of the $(2k+1, k, w_e, r)$-RMSD over $\mathcal{R}$ can be reduced to decoding $[5r, 2r]$-linear codes over $\mathbb{F}_{q^m}$ with rank weight $w_e$. One can use the best combinatorial attack algorithm in [22] to determine the choice of parameters such as $m, r, w, w_e$. Furthermore, we can determine $l$ since $ww_e \leq \frac{r-l}{2}$. Those parameters also need to resist the algebraic attacks which are presented in Sect. 6. The concrete parameters are listed in Table 1.

**Table 1.** Parameter sets of Piglet-1.CCAKEM

| Instance | $k$ | $q$ | $m$ | $r$ | $w$ | $w_e$ | $l$ | Security level |
|---|---|---|---|---|---|---|---|---|
| Piglet-1.CCAKEM-I | 2 | 2 | 89 | 53 | 5 | 5 | 3 | 128 |
| Piglet-1.CCAKEM-II | 2 | 2 | 149 | 53 | 5 | 5 | 3 | 192 |
| Piglet-1.CCAKEM-III | 2 | 2 | 149 | 75 | 6 | 6 | 3 | 256 |

**Table 2.** The theoretical sizes in bytes for Piglet-1.CCAKEM

| Instance | pk size | sk size | ct size | ss size | Security level |
|---|---|---|---|---|---|
| Piglet-1.CCAKEM-I | 1212 | 40 | 1801 | 64 | 128 |
| Piglet-1.CCAKEM-II | 2007 | 40 | 2994 | 64 | 192 |
| Piglet-1.CCAKEM-III | 2826 | 40 | 4223 | 64 | 256 |

Table 2 presents the theoretical sizes in bytes for Piglet-1.CCAKEM. The size of pk is $kmr + 256$ bits, i.e., $\frac{2mr+256}{8}$ bytes. The size of sk is 256 bits, i.e., 32 bytes. The size of ciphertext is $3mr + 256$ bits, i.e., $3mr/8 + 32$ bytes. The size of ss (session secret) is $2 \times 256$ bits, i.e., 64 bytes.

**Table 3.** Comparison on sizes of public keys (in bytes)

| Instance | 128 bits | 192 bits | 256 bits |
|---|---|---|---|
| Classic McEliece | 368,282 | | 1,046,737 |
| NTS-kem | 319,488 | 929,760 | 1,419,704 |
| Piglet-1.CCAKEM | 1212 | 2007 | 2826 |
| Piglet-2.CPAKEM | 1212 | 2007 | 2826 |
| RQC | 786 | 1411 | 1795 |
| HQC | 2819 | 5115 | 7417 |
| LEDAKem | 3,480 | 7,200 | 12,384 |
| BIKE-I | 2541 | 5474 | 8181 |
| BIKE-II | 1271 | 2737 | 4094 |
| BIKE-III | 2757 | 5421 | 9033 |
| Ouroboros-R | 676 | 807 | 1112 |
| LOCKER | 737 | 1048 | 1191 |

Table 3 presents parameters comparison between our scheme and some NIST submissions which proceed the second round of NIST PQC standardization process. As we have analyzed in Sect. 3, it shows that the size of public key in our schemes is slightly larger than those in RQC, Ouroboros-R and LOCKER, which are based RQCSD hardness problem. The size of public key in our schemes is better than those in HQC, LEDAkem, BIKE which are based on the QCSD hardness problem. And it is much better than those in Classic McEliece and NTS-kem which are original McEliece cryptosystems.

## 5 Piglet-2: A New Module-Code-Based KEM

In this section, we propose a new IND-CPA-secure KEM Piglet-2.CPAKEM. The difference lies in choice of the auxiliary codes we use (LRPC codes for Piglet-2.CPAKEM, Gabidulin codes for Piglet-1.CPAPKE). The session key is the hash value of error vectors without encrypting a plaintext. As for LRPC codes, we introduced them in Sect. 2. In addition, $G : \{0,1\}^* \rightarrow \{0,1\}^{2\times 256}$ denotes a hash function.

Piglet-2.CPAKEM.Keygen(): key generation

```
1. $\rho \xleftarrow{\$} \{0,1\}^{256}$, $\sigma \xleftarrow{\$} \{0,1\}^{320}$
2. $H \in \mathcal{R}^{k\times k} := \mathrm{XOF}(\rho)$
3. $(\mathbf{x},\mathbf{y}) \in \mathcal{R}^k \times \mathcal{R}^k := \mathrm{XOF}(\sigma)$ with $w_R(\mathbf{x}) = w_R(\mathbf{y}) = w$
4. $\mathbf{s} := \mathbf{x}H + \mathbf{y}$
5. return $(pk := (H, \mathbf{s}), sk := (\mathbf{x}, \mathbf{y}))$
```

Piglet-2.CPAKEM.Encaps($\rho, \mathbf{s}$): encapsulation

```
1. $\tau \xleftarrow{\$} \{0,1\}^{320}$
2. $H \in \mathcal{R}^{k\times k} := \mathrm{XOF}(\rho)$
3. $(\mathbf{r},\mathbf{e},\mathbf{e}') \in \mathcal{R}^k \times \mathcal{R}^k \times \mathcal{R} := \mathrm{XOF}(\tau)$ with $w_R(\mathbf{r}) = w_R(\mathbf{e}) = w_R(\mathbf{e}') = w_e$
4. $E := \mathrm{Supp}(\mathbf{r},\mathbf{e},\mathbf{e}')$ and $K := G(E)$
5. $\mathbf{u} := H\mathbf{r}^T + \mathbf{e}^T$
6. $\mathbf{v} := \mathbf{s}\mathbf{r}^T + \mathbf{e}'$
7. return a ciphertext pair $\mathbf{c} := (\mathbf{u}, \mathbf{v})$
```

Piglet-2.CPAKEM.Decaps($sk = (\mathbf{x},\mathbf{y}), \mathbf{c} = (\mathbf{u},\mathbf{v})$): decapsulation

```
1. $F := \mathrm{Supp}(\mathbf{x},\mathbf{y})$
2. Compute $\mathbf{v} - \mathbf{x}\mathbf{u} := \mathbf{y}\mathbf{r}^T + \mathbf{e}' - \mathbf{x}\mathbf{e}^T$
3. $E := \text{RS-recover}(F, \mathbf{v} - \mathbf{x}\mathbf{u}, w_e)$
4. $K := G(E)$
```

*Remark 4.* 1. In the above scheme, $E = \text{RS-recover}(F, \mathbf{v} - \mathbf{x}\mathbf{u}, w_e)$ denotes that the decoding algorithm outputs the support $E$ of error vectors $\mathbf{r}, \mathbf{e}$ and $\mathbf{e}'$ with dimension $w_e$ given the support $F$ of $\mathbf{x}$ and $\mathbf{y}$ and the syndrome $\mathbf{v} - \mathbf{x}\mathbf{u}$.

2. The security proof of Piglet-2.CPAKEM is the same as that of Piglet-1.CPAPKE and so we omit it here.
3. The choice of parameter sets for Piglet-2.CPAKEM are the same as that for Piglet-1.CCAKEM.

4. The rank support recovery algorithm is probabilistic and the decoding failure probability can be computed by Proposition 1. So in our case the result is $\max(q^{(2-w)(w_e-2)} \times q^{-(r-ww_e+1)}, q^{-2(r-ww_e+2)}) = 2^{-38}$ for both 128 and 192 bits security levels, and $2^{-52}$ for 256 bits security level.
5. Since rank support recovery decoding techniques do not attain a negligible decoding failure rate, this makes it challenge to achieve higher security notions such as IND-CCA.

## 6 Known Attacks

There are two types of generic attacks on our schemes, which play an important role in choice of parameter sets in our schemes. One is general combinatorial decoding attack and the other is algebraic attack using Gröbner basis.

The decoding algorithm was proposed in [3,21] and the best result is as follows.

For an $[n, k]$ rank code $\mathcal{C}$ over $\mathbb{F}_{q^m}$, the time complexity of the known best combinatorial attack to decode a word with rank weight $d$ is

$$O((nm)^3 q^{d\lceil \frac{m(k+1)}{n} \rceil - m}). \tag{2}$$

As for algebraic attack, the time complexity is much greater than the decoding attack when $q = 2$. The complexity of the above problem is $q^{d\lceil \frac{d(k+1)-(n+1)}{d} \rceil}$ [28].

Next, the general attacks from [42] which use the cyclic structure of the code have less impact on module codes than quasi-cyclic codes in RQC, Ouroboros-R, LOCKER, etc.

In addition, as for the choice of $r$, no attacks of quasi-cyclicity of a code are known if there are only two factors of $x^r - 1 \mod q$ [26]. Therefore, $r$ should be prime, and $q$ is a generator of the multiplicative group of $(\mathbb{Z}/r\mathbb{Z})^*$.

## 7 Conclusions

In this paper, we propose an IND-CCA-secure KEM Piglet-1.CCAKEM and an IND-CPA-secure Piglet-2.CPAKEM, both of which are based on the RMSD difficult problem. More importantly, the size of public key in our schemes is much shorter than those of NIST submissions which entered the second round except the candidates based on RQCSD hardness problem. The shorter keys from the RQCSD-problem related candidates are due to simple quasi-cyclic structure used. However, the advantage of our new construction is the elimination of possible quasi-cyclic attacks and thus makes our schemes strong and robust. The parameter comparison between Piglet and other NIST proposals shows that our schemes would be good candidates for post-quantum cryptosystems with long-term security. Moreover, we expect to further reduce the public key size by using similar Kyber's approach in our future work.

**Acknowledgment.** The author would like to thank the anonymous reviewers for their valuable comments and suggestions which improved the quality of this paper. The work of L.-P. Wang was supported in part by the National Natural Science Foundation of China (Grant No. 61872355) and National Cryptography Development Fund (MMJJ 20170124).

## References

1. Aguilar-Melchor, C., Blazy, O., Deneuville, J.-C., Gaborit, P., Zémor, G.: Efficient encryption from random quasi-cyclic codes. IEEE Trans. Inf. Theory **64**(5), 3927–3943 (2018)
2. Alekhnovich, M.: More on average case vs approximation complexity. Comput. Complex. **20**(4), 755–786 (2011)
3. Aragon, N., Gaborit, P., Hautevile, A., Tillich, J.-P.: Improvement of generic attacks on the rank syndrome decoding problem (2017). Pre-print https://www.unilim.fr/pages_perso/philippe.gaborit/newGRS.pdf
4. Aragon, N., Barreto, P., Bettaieb, S., Bidoux, L., Blazy, O., et al.: BIKE: bit flipping key encapsulation. Submission to the NIST Post Quantum Standardization Process (2017)
5. Aragon, N., Blazy, O., Deneuville, J.-C., Gaborit, P., Hauteville, A., et al.: LOCKER: low rank parity check codes encryption. Submission to the NIST Post Quantum Standardization Process (2017)
6. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. J. Math. Cryptol. **9**(3), 169–203 (2015)
7. Baldi, M.: QC-LDPC Code-Based Cryptography. Springer Briefs in Electrical and Computer Engineering. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-02556-8
8. Baldi, M., Barenghi, A., Chiaraluce, F., Pelosi, G., Santini, P.: LEDAkem: a post-quantum key encapsulation mechanism based on QC-LDPC codes. In: Lange, T., Steinwandt, R. (eds.) PQCrypto 2018. LNCS, vol. 10786, pp. 3–24. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-79063-3_1
9. Barreto, P.S.L.M., Lindner, R., Misoczki, R.: Monoidic codes in cryptography. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 179–199. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_12
10. Berlekamp, E., McEliece, R., Van Tilborg, H.: On the inherent intractability of certain coding problems. IEEE Trans. Inf. Theory **24**(3), 384–386 (1978)
11. Biasse, J.-F., Song, F.: Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields. In: Krauthgamer, R. (ed.) 27th SODA, pp. 893–902. ACM-SIAM (2016)
12. Bos, J.W., et al.: CRYSTALS- Kyber: a CCA-secure module-lattice-based KEM. In: EuroS&P 2018, pp. 353–367 (2018)
13. Campbell, P., Groves, M., Shepherd, D.: Soliloquy: a cautionary tale. In: ETSI 2nd Quantum-Safe Crypto Workshop, pp. 1–9 (2014)
14. Cramer, R., Ducas, L., Peikert, C., Regev, O.: Recovering short generators of principal ideals in cyclotomic rings. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 559–585. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_20

15. Cramer, R., Ducas, L., Wesolowski, B.: Short stickelberger class relations and application to ideal-SVP. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10210, pp. 324–348. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_12
16. Delsarte, P.: Bilinear forms over a finite field, with applications to coding theory. J. Comb. Theory Ser. A **25**(3), 226–241 (1978)
17. Deneuville, J.-C., Gaborit, P., Zémor, G.: Ouroboros: a simple, secure and efficient key exchange protocol based on coding theory. In: Lange, T., Takagi, T. (eds.) PQCrypto 2017. LNCS, vol. 10346, pp. 18–34. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59879-6_2
18. Faugère, J.-C., Otmani, A., Perret, L., Tillich, J.-P.: Algebraic cryptanalysis of McEliece variants with compact keys. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 279–298. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_14
19. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_34
20. Gabidulin, E.M.: Theory of codes with maximum rank distance. Probl. Inf. Transm. **21**(1), 3–16 (1985)
21. Gaborit, P., Ruatta, O., Schrek, J.: On the complexity of the rank syndrome decoding problem. IEEE Trans. Inf. Theory **62**(2), 1006–1019 (2016)
22. Gaborit, P., Zémor, G.: On the hardness of the decoding and the minimum distance problem for rank codes. IEEE Trans. Inf. Theory **62**(12), 7245–7252 (2016)
23. Gaborit, P., Hauteville, A., Phan, D.H., Tillich, J.-P.: Identity-based encryption from codes with rank metric. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 194–224. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_7
24. Gaborit, P., Murat, G., Ruatta, O., Zémor, G.: Low rank parity check codes and their application to cryptography. In: Proceedings of the Workshop on Coding and Cryptography WCC 2013, Bergen, Norway (2013)
25. Gabidulin, E.M., Paramonov, A.V., Tretjakov, O.V.: Ideals over a non-commutative ring and their application in cryptology. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 482–489. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-46416-6_41
26. Hauteville, A., Tillich, J.-P.: New algorithms for decoding in the rank metric and an attack on the LRPC cryptosystem. In: 2015 IEEE International Symposium on Information Theory (ISIT), pp. 2747–2751 (2015)
27. Landais, G., Tillich, J.-P.: An Efficient attack of a McEliece cryptosystem variant based on convolutional codes. In: Gaborit, P. (ed.) PQCrypto 2013. LNCS, vol. 7932, pp. 102–117. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38616-9_7
28. Levy-dit-Vehel, F., Perret, L.: Algebraic decoding of rank metric codes. In: Proceedings of YACC 2006 (2006)
29. Loidreau, P.: Properties of codes in rank metric. http://arxiv.org/abs/cs/0610057
30. Loidreau, P.: A Welch–Berlekamp like algorithm for decoding Gabidulin codes. In: Ytrehus, Ø. (ed.) WCC 2005. LNCS, vol. 3969, pp. 36–45. Springer, Heidelberg (2006). https://doi.org/10.1007/11779360_4
31. Loidreau, P.: A new rank metric codes based encryption scheme. In: Lange, T., Takagi, T. (eds.) PQCrypto 2017. LNCS, vol. 10346, pp. 3–17. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59879-6_1

32. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_1
33. McEliece, R.J.: A public key crytosystem based on algebraic coding theory. DSN progress report 44, pp. 114–116 (1978)
34. Misoczki, R., Tillich, J.-P., Sendrier, N., Barreto, P.S.L.M.: MDPCMcEliece: new McEliece variants from moderate density parity-check codes. In: Proceedings of the IEEE International Symposium on Information Theory - ISIT 2013, pp. 2069–2073 (2013)
35. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Probl. Control. Inf. Theory **15**, 159–166 (1986)
36. Overbeck, R.: Structural attacks for public key cryptosystems based on Gabidulin codes. J. Cryptol. **21**, 280–301 (2008)
37. Ore, O.: Theory of non-commutative polynomials. Ann. Math. **34**(3), 480–508 (1933)
38. NIST. Post Quantum Crypto Project (2017). http://csrc.nist.gov/groups/ST/post-quantum-crypto. Available at https://csrc.nist.gov/Projects/Post-Quantum-for-Cryptography/Post-Quantum-Cryptography-Standardization/call-for-Proposalls. List of First Round candidates available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions
39. Pietrzak, K.: Cryptography from learning parity with noise. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) SOFSEM 2012. LNCS, vol. 7147, pp. 99–114. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27660-6_9
40. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005, pp. 84–93 (2005)
41. Santini, P., Baldi, M., Cancellieri, G., Chiaraluce, F.: Hindering reaction attacks by using monomial codes in the McEliece cryptosystem. In: IEEE International Symposium on Information Theory (ISIT) 2018, pp. 951–955 (2018)
42. Sendrier, N.: Decoding one out of many. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 51–67. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_4
43. Shor, P.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
44. Sidorenko, V., Richter, G., Bossert, M.: Linearized shift-register synthesis. IEEE Trans. Inf. Theory **57**(9), 6025–6032 (2011)
45. Sidelnikov, V.M.: A public-key cryptosystem based on binary Reed-Muller codes. Discrete Math. Appl. **4**, 191–207 (1994)
46. Sidelnikov, V.M., Shestakov, S.O.: On insecurity of cryptosystems based on generalized Reed-Solomon codes. Discrete Math. Appl. **2**, 439–444 (1992)