REDOG [⋆]

Jon-Lark Kim[1], Jihoon Hong[2], Terry Shue Chien Lau[3], YounJae Lim[4], and
Byung-Sun Won[5] Bo-seung Yang[6]

[1] Sogang University
jlkim@sogang.ac.kr
[2] Sogang University
rjekfl@sogang.ac.kr
[3] Multimedia University
terry.lau@mmu.edu.my
[4] DeepHelix Corp.
yjlim@deephelix.net
[5] Sogang University
bswon@deephelix.net
[6] Sogang University
didqhtmd@gmail.com

Abstract. We propose a REinforced modified Dual-Ouroboros based on Gabidulin codes, shortly called REDOG to the KpqC conpetition round 2. This is a code-based cryptosystem based on the well-known rank metric codes, Gabidulin codes. The public key sizes of REDOG are 4.38KB, 14.11KB, 32.66KB at the security levels of 128, 192, 256 bits respectively. There is no decoding failure in decryption. REDOG is IND-CPA. This paper describes the current status of REDOG and its attacks for KpqC conpetition round 2.

Keywords: Modified Dual-Ouroboros · Gabidulin code · λ-dimensional subspace · KpqC conpetition Round 2 .

Changelog

The following are the changes of (ALG)

- Considering about selection of the secret key S.
  It is described in Section 3.2.
- Considering about decryption failure(depending on how the weight for the error vector $e = (e_1, e_2)$ was configured).
  It is described in Section 3.3.
- Considering about new attacks on the rank-metric code(BBB+, BBC+, BBB+23 attacks).
  It is described in Section 4.3.

[⋆] This work is submitted to 'Korean Post-Quantum Cryptography Competition' (www.kpqc.or.kr). Jon-Lark Kim is a principal investigator.

- Reflecting the changes mentioned above, modifications have been made to the parameters in the setup part and the selection of the secret key S in the key generation part of the REDOG algorithm.
  These changes are reflected in Section 3 specification.

1 Introduction

In this paper, we introduce a cryptographic system based on rank-metric codes called REDOG. Cryptographic systems play a crucial role in ensuring secure communication and information protection.

We initially proposed the REDOG system at the 2022 KpqC competition. Subsequently, various attacks were proposed on multiple occasions. This paper describes the new version of REDOG and its modifications following these attacks and explains about modified version to KpqC conpetition round 2.

After the initial proposal of REDOG in the 2022 KpqC competition, several attacks suggested by T. Lange et al. occurred on July 13, 2023, and August 9, 2023. We provided solutions and responses for each attack. The effectiveness of these measures was later validated in a document presented by Lange et al. on November 15, 2023. Additionally, further considerations were made for various attacks on the newly proposed rank-metric code, and adjustments were implemented in areas prone to decoding and decryption failures.

Through these processes, we were able to propose parameters superior to those initially suggested for each security level in 2022. The history of attacks can be checked in [17], with versions released on August 8, August 10, and November 15.

1.1 Design rationale

The original version of the McNie series called McNie [21] had the features of both McEliece and Niederreiter cryptosystems and was designed to be secure against known structural attacks on code-based cryptosystems. Gaborit [22] suggested a message recovery attack that reduced the dimension of a random code in the public key. The security level of McNie decreased by almost a factor of 2, and the original parameters suggested for McNie suffer from relatively high decryption failure probability since LRPC (low-rank parity check codes) decoding is a probabilistic decoding algorithm.

To overcome those disadvantages, Dual-Ouroboros, a modification of McNie, was proposed [10]. It was a non-cyclic dual version of Ouroboros-R [2], which also employed the LRPC codes. Kim et al. [12] suggested a modified Dual-Ouruboros(DO.Gab-PKE), which is a variant of Dual-Ouroboros obtained by replacing LRPC codes from Dual-Ouroboros by Gabidulin codes over $\mathbb{F}_{q^m}$. Gadibulin $[n, k]$ codes have the advantage of the zero-decoding failure probability and have a fast decoding complexity of $O(n^2)$ operations over $\mathbb{F}_{q^m}$ [15] and an improved decoding complexity of $O(nm^2 \log m)$ operations over the ground field $\mathbb{F}_q$. Moreover, the modified DO.Gab-PKE using Gabidulin codes provides much stronger security against known plaintext-recovery attacks, including Overbeck's attack [18]. It was also shown in [12] that the DO.Gab-PKE achieves $\mathsf{IND-CPA}$ security, and the parameters achieve relatively lower key sizes compared to the other code-based PKE that has no decryption failure.

However, the modified DO.Gab-PKE did not specify the secret key S selection to ensure the modified DO's security.Gab-PKE. If the secret key S is invertible over $\mathbb{F}_{q^m}$ without any restriction, then the modified DO.Gab-PKE would be incorrect. If S is invertible over $\mathbb{F}_q$ without any restriction, the modified DO.Gab-PKE would be insecure. Therefore, we need to select S specifically for the modified DO.Gab-PKE can be secure. This reinforced version was called the modified DO.Gab[Λ]-PKE in [14]. Therefore, in this proposal, we describe the modified DO.Gab[Λ]-PKE in [14], which is shortly called REDOG meaning a **RE**inforced modified **D**ual-**O**uroboros based on **G**abidulin codes.

There were several identified issues in the initial version of REDOG which is proposed to KpqC conpetition round 1. One concern was the potential for decryption failure depending on how the weight for the error vector e was configured. Additionally, there was a lack of consideration for various newly proposed attacks on the rank-metric code. Efforts have been made to address these issues, and by making slight modifications to the cryptographic scheme, it was possible to further reduce the key size.

1.2 Advantages and limitations

At first, REDOG adopts the same scheme as the modified DO.Gab-PKE [12] and just clearly specifies how to select the secret key S to avoid the Frobenius weak attack [13]. By using the same encryption algorithm, the structural stability of the algorithm and the resistance to known attacks can be brought as it is. Moreover, by selecting the secret key S to be an invertible matrix over a λ-dimensional subspace of $\mathbb{F}_{q^m}$, the public key matrix does not generate a r-Frobenius weak code [16].

However, after we proposed the REDOG at the KpqC competition round 1, we had some problems with decoding and decryption. During the process of making adjustments to enhance defense against multiple attacks, we were able to prevent decoding and decryption failures through slight modifications to the cryptographic algorithm. Additionally, we considered various rank-metric attacks that were not previously taken into account. As a result, the size of the public key, which was initially 14.25 KB when proposed at the KpqC competition round 1 to achieve a 128-bit security level, has significantly decreased to 4.38 KB while maintaining the same security level.

2 Preliminaries

In this section, we introduce essential concepts related to rank metric codes and discuss some key elements used in our cryptographic algorithm.

2.1 Rank metric codes

Let q be a prime power and $\mathbb{F}_{q^m}$ be the finite field with q^m elements. Consider a basis $\{\beta_1, ..., \beta_m\}$ of $\mathbb{F}_{q^m}$ over the base field $\mathbb{F}_q$.

Definition 1. An $[n, k]$ linear code of length n and dimension k is a linear subspace $\mathcal{C}$ of the vector space $\mathbb{F}_{q^m}^n$, i.e. $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$. Let $\ell \leqslant k$, then an $[n, \ell]$ linear subcode $\mathcal{C}'$ is an $[n, \ell]$ linear code such that $\mathcal{C}' \subseteq \mathcal{C}$.

Definition 2. Let $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{F}_{q^m}^n$. For each $1 \leqslant j \leqslant n$, $x_j = \sum_{i=1}^m c_{ij}\beta_i$ where $c_{ij} \in \mathbb{F}_q$. The rank of $\mathbf{x}$ in $\mathbb{F}_q$, denoted by $\mathrm{wt}_R(\mathbf{x})$ is defined as $\mathrm{wt}_R(\mathbf{x})$=$\mathrm{wt}_R(X)$ where $X = [c_{ij}] \in \mathbb{F}_q^{m \times n}$.

2.2 Gabidulin codes and partial cyclic codes

Definition 3. Let $\mathbf{x} = (x_0, ..., x_{n-1}) \in \mathbb{F}_{q^m}^n$. The circulant matrix $\mathrm{Cir}_n(\mathbf{x})$ induced by $\mathbf{x}$ is defined as

$$\mathrm{Cir}_n(\mathbf{x}) = \left[x_{i-j \pmod n}\right]_{ij} = \begin{bmatrix} x_0 & x_{n-1} & \cdots & x_1 \\ x_1 & x_0 & \cdots & x_2 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n-1} & x_{n-2} & \cdots & x_0 \end{bmatrix}$$

The $k \times n$-partial circulant matrix induced by $\mathbf{x}$, denoted by $\mathrm{Cir}_k(\mathbf{x})$ is defined as the first k rows of $\mathrm{Cir}_n(\mathbf{x})$.

Lau and Tan [13] defined the following code generated by $\mathrm{Cir}_k(\mathbf{x})$.

Definition 4. An $[n, k]$-partial cyclic code $\mathsf{PC}_{n,k}[\mathbf{x}]$ generated by $\mathbf{x} \in \mathbb{F}_{q^m}^n$ is a linear code with generator matrix $\mathrm{Cir}_k(\mathbf{x})$

These circulant matrices will be used as generator matrices of Gabidulin codes as in Section 3.3 in order to reduce the key sizes.

The following are the definitions for the Moore matrix and Gabidulin codes.

Definition 5. Denote $[\ell] = q^\ell$ as the ℓth Frobenius power for an integer ℓ. A matrix $G = [G_{ij}] \in \mathbb{F}_{q^m}^{k \times n}$ is called a *Moore matrix* induced by $\mathbf{g}$ if there exists a vector $\mathbf{g} = (g_1, ..., g_n) \in \mathbb{F}_{q^m}^n$ such that the ith row of G is equal to $\mathbf{g}^{[i-1]} = (g_1^{[i-1]}, ..., g_n^{[i-1]})$ for $1 \leqslant i \leqslant k$, i.e., G is of the form

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

Similarly, we define $G^{[\ell]} = \left[G_{ij}^{[\ell]}\right]$. For any set $S \subset \mathbb{F}_{q^m}^n$, we denote $S^{([\ell])} = \{\mathbf{s}^{[\ell]} | \mathbf{s} \in S\}$

Definition 6. *(Gabidulin code)* Let $\mathbf{g} \in \mathbb{F}_{q^m}^n$ with $\mathrm{wt}_R(\mathbf{g}) = n \leqslant m$. The $[n, k]$ Gabidulin code $\mathrm{Gab}_{n,k}(\mathbf{g})$ over $\mathbb{F}_{q^m}^n$ of dimension k with generator vector $\mathbf{g}$ is the code generated by a Moore matrix G induced by $\mathbf{g}$ in the form of Equation (1).

Theorem 1. There exists a Moore Matrix $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ such that H is a parity-check matrix of a Gabidulin code. In other words, the dual of a Gabidulin code is also a Gabidulin code.

The error-correcting capability of $\mathrm{Gab}_{n,k}(\mathbf{g})$ is $r = \left\lfloor \frac{n-k}{2} \right\rfloor$. There exist efficient decoding algorithms for Gabidulin codes that can correct errors up to rank r (for instance [15] with decoding complexity $5/2\ n^2 - 3/2\ k^2$).

Definition 7. *(r-Frobenius weak)* Let C be an $[n, k]$-linear code. We say that C is r-Frobenius weak if for some s relatively prime to m and for a generic $\mathbf{e} \in \mathbb{F}_{q^m}^n$ of rank r, the space U spanned by the elements of rank one in $C_{\mathsf{ext}} = \sum_{i=0}^{r-1} \left(C + \langle \mathbf{e} \rangle_{\mathbb{F}_{q^m}}\right)^{[si]}$, fulfills $C \cap U = \{0\}$.

The algorithm of Frobenius weak attack [13] is as follows.

```
Algorithm : FrobeniusWeakAttack
Data : $\mathbf{y} = \mathbf{m}G_{\mathsf{pub}} + \mathbf{e}$ (a ciphertext where $\mathbf{m}$ is the plaintext), the public key
       $\mathsf{pk} = G_{\mathsf{pub}}$ with parameter $r = \mathrm{wt}_R(\mathbf{e})$
Result : The plaintext $\mathbf{m}$
Construct the matrix
    $$G_{\mathsf{pub,ext}} = \begin{bmatrix} G_{\mathsf{pub}} \\ \mathbf{y} \\ \vdots \\ G_{\mathsf{pub}}^{[r-1]} \\ \mathbf{y}^{[r-1]} \end{bmatrix}.$$
Compute the space $\mathcal{U}$ generated by the elements of rank one in $\mathcal{C}_{\mathsf{ext}} = \langle G_{\mathsf{pub,ext}} \rangle_{\mathbb{F}_{q^m}}$.
Compute $u = \dim_{\mathbb{F}_{q^m}}(\mathcal{U})$.
if $u \leqslant n - k$ then
    Compute a parity-check matrix $H_U \in \mathbb{F}_q^{(n-u)\times n}$ for $\mathcal{U}$.
    Solve $\mathbf{y}(H_U)^T - \mathbf{m}[G_{\mathsf{pub}}(H_U)^T]$ for $\mathbf{m}$,
    return $\mathbf{m}$.
else
    return $\perp$
```

3 Specification

We describe key generation, encryption, and decryption of REDOG as follows.

Setup: Generate global parameters with integers m, n, l, r, k such that $\ell < n$ and $t_1 + \lambda t_2 \leqslant r \leqslant \left\lfloor \frac{n-k}{2} \right\rfloor$. Output parameters $= (m, n, \ell, k, r, \lambda, t_1, t_2)$.

Key.Gen: Select $H = [H_1 H_2]$, $H_2 \in \mathsf{GL}_{n-k}(\mathbb{F}_{q^m})$, a parity check matrix of a $[2n-k, n]$
Gabidulin code $\mathcal{C}$, with syndrome decoder Φ correcting r errors where $r = \left\lfloor \frac{n-k}{2} \right\rfloor$. Select a full rank matrix $M \in \mathbb{F}_{q^m}^{\ell \times n}$. Select a λ-dimensional subspace $\Lambda \subset \mathbb{F}_{q^m}$, seen as $\mathbb{F}_q$-linear space, and select $S^{-1} \in \mathsf{GL}_{n-k}(\Lambda)$ and $P \in \mathbb{F}_{q^m}^{n \times n}$.
Output public key and secret key pair
$\mathsf{pk} = (M, F = MP^{-1}H_1^T[H_2^T]^{-1}S), \mathsf{sk} = (P, H, S, \Phi)$.

$\mathsf{Enc}(\mathsf{pk}, \mathbf{m} \in \mathbb{F}_{q^m}^{\ell})$: Let $\mathbf{m} \in \mathbb{F}_{q^m}^{\ell}$ be the plaintext message to be encrypted. Generate uniformly random vector $\mathbf{e} = (e_1, e_2) \in \mathbb{F}_{q^m}^{2n-k}$ with $\mathrm{wt}_R(e_1) = t_1$, and $\mathrm{wt}_R(e_2) = t_2$, where $e_1 \in \mathbb{F}_{q^m}^{n}$ and $e_2 \in \mathbb{F}_{q^m}^{n-k}$. Compute $\mathbf{m}' = \mathbf{m} + \mathcal{H}(\mathbf{e})$. Compute $c_1 = \mathbf{m}'M + e_1$ and $c_2 = \mathbf{m}'F + e_2$.
Output ciphertext $\mathbf{c} = (c_1, c_2)$.

$\mathsf{Dec}(\mathsf{sk}, c = (c_1, c_2))$: Compute

$$
\begin{aligned}
&c_1 P^{-1} H_1^T - c_2 S^{-1} H_2^T \\
&\quad = \mathbf{m}'MP^{-1}H_1^T + e_1 P^{-1} H_1^T - \mathbf{m}'MP^{-1}H_1^T[H_2^T]^{-1}SS^{-1}H_2^T - e_2 S^{-1} H_2^T \\
&\quad = e_1 P^{-1} H_1^T - e_2 S^{-1} H_2^T \\
&\quad = (e_1 P^{-1}, -e_2 S^{-1}) \begin{bmatrix} H_1^T \\ H_2^T \end{bmatrix}
\end{aligned}
$$

Let $\mathbf{e}' = (e_1 P^{-1}, -e_2 S^{-1})$. Since $\mathrm{wt}_R(\mathbf{e}') \leqslant r$, apply Φ_H to obtain $\mathbf{e}'$.
Compute $e_1 = e_1 P^{-1} P$ and $e_2 = e_2 S^{-1} S$ to obtain $\mathbf{e} = (e_1, e_2)$.
Finally, solve the system $\mathbf{m}'G = c_1 - e_1$ to recover $\mathbf{m} = \mathbf{m}' - \mathcal{H}(e)$.

3.1 Notation

All the notations for specification are given above.

3.2 Specification of REDOG

In the previous version of REDOG which is proposed to KpqC competition round 1, we take the secret key S in some conditions. To avoid the decryption failure, we consider the matrix S^{-1} over some λ-dimensional subspace $\Lambda \subset \mathbb{F}_{q^m}$.

In the document [17], Lange et al. show the key-recovery attack from [14] and show that its applicability is not as general as claimed. The attack is effective only when $P \in \mathsf{GL}_n(\mathbb{F}_q)$, in contrast to the assertion in [14], where it is mentioned for P is an isometry. The key-recovery attack operates on the assumption that given M, the second part of the public key ($F = MP^{-1}H_1^T[H_2^T]^{-1}S$) can be represented as a system of linear equations over $\mathbb{F}_q$ using the coordinates of $H_1' = H_1(P^{-1})^T$ and $H_2' = H_2(S^{-1})^T$. The claim in [14] is that if H_1' and H_2' are Moore matrices, the system becomes overdetermined, reducing the number of unknowns. This assertion is also utilized in the plaintext-recovery attack to show that the public code is a subcode of a Gabidulin code. Consequently, Lange et al. demonstrate that $H_1' = H_1(P^{-1})^T$ is a Moore matrix only if $P \in \mathsf{GL}n(\mathbb{F}_q)$ and not in the general case where P is an isometry. Similarly, $H_2' = H_2(S^{-1})^T$ is a Moore matrix when $S \in \mathsf{GL}n - k$.

We adopt the approach of selecting matrices S and S^{-1} and addressing the associated decoding failure in Sections 7 and 8 [17]. These methods are incorporated with slight modifications to the key generation part of the existing algorithm. As stated in Theorem 8.4, the modified version of REDOG is proven to be correct without encountering any decryption failure.

Throughout this process, opting for a prime value of m during selection serves to prevent a reduction in the number of variables within the system of equations. Consequently, we have adjusted the parameter m from its initially proposed values, opting to designate it as a prime number.

3.3 Selection for (e_1, e_2)

Since the initial observation of the rank and range of error e on July 13, 2023, we have proposed a new method for error selection. However, in the REDOG encryption algorithm based on the slightly modified key generation proposed by [17], the optimal selection of error $e = (e_1, e_2)$ has also been newly suggested. Their document [17] in Section 9 presents an extreme choice method and its consequences (computational cost) for $\mathrm{wt}_R(e_1) = t_1$ and $\mathrm{wt}_R(e_2) = t_2$.

To make the best attacks as hard as possible, they consider attacks starting from the left with (parts of) c_1 and M or from the right with c_2, F, and parts of c_1 and M. The attacks and sub-attacks differ in how many columns they require, depending on the dimension and rank, and they scan the whole range of possible lengths from both sides.

Since $n = \ell + t + 1$, for the t parameter in REDOG, for small choices of $t_1 \leqslant t$ the attack may take a punctured system on c_1 and M to recover m', similar to the attacks which uses the GRS algorithm, or include part of c_2 and F while accepting an error of larger rank including part of t_2. Hence, the search from the left may start with the puncturing of c_1. Once parts of c_2 are included, the rank typically increases by one for each extra position, again because m is much larger than t_1 and t_2, until reaching $t_1 + t_2$, after which the rank does not increase with increasing length.

If $t_1 > t + 1$ parts of c_2 need to be considered in any case, with the corresponding increases in the rank of the error, in turn requiring more positions to deal with the increased rank, typically reaching $t_1 + t_2$ before enough positions are available. Starting from the right, the attacker will always need to include parts from c_1 to even have an invertible system. Hence, the attack is hardest for t_1 maximal in $r \geqslant t_1 + \lambda t_2$ provided that the brute-force attack is excluded. This suggests choosing $t_2 = 1$, $t_1 = r - \lambda$, as then the attacker is forced to decode an unstructured code with an error of rank $t_1 + t_2 = r - \lambda + 1$.

After a computer search, they suggest the parameters for t_1 and t_2 with the given parameters which we suggested for the security of 128, 192, and 256. So we have adopted this approach, set t_1 and t_2 accordingly, and propose new parameters that satisfy each security level.

3.4 Parameter sets

We present our proposed parameters for REDOG in Table 1. We consider M to be an any $(\ell \times n)$ full rank matrix, and S^{-1} to be an $(n-k) \times (n-k)$ circulant matrix. The public key size is $\mathsf{size}_{\mathsf{pk}} = m\ell(n-k)/8$ bytes, the secret key size is $\mathsf{size}_{\mathsf{sk}} = (n^2 + (3n-2k)m)/8$ bytes, and the ciphertext size is $\mathsf{size}_{\mathsf{ct}} = (2n-k)m/8$ bytes.

After the modification of λ and the error weight $t_1 = \mathrm{wt}_R(e_1), t_2 = \mathrm{wt}_R(e_2)$, the costs go higher so we obtained calculation results show that smaller parameters than the previous parameters which we suggested for the KpqC competition round 1 can be used to satisfy a given security level. $(n, k, \ell, m, r, \lambda, t_1, t_2) = (30, 6, 25, 59, 12, 3, 6, 2)$, $(n, k, \ell, m, r, \lambda, t_1, t_2) = (44, 8, 37, 83, 18, 3, 12, 2)$ and $(n, k, \ell, m, r, \lambda, t_1, t_2) = (58, 10, 49, 109, 24, 3, 15, 3)$ parameters satisfy security level of 128, 192 and 256, respectively.

Table 1. Proposed parameters for REDOG

Instance	$(n, k, \ell, q, m, r, \lambda, t_1, t_2)$	$\mathsf{size}_{\mathsf{pk}}$	$\mathsf{size}_{\mathsf{sk}}$	$\mathsf{size}_{\mathsf{ct}}$	Security level
REDOG-1	(30,6,25,2,59,12,3,6,2)	4.17KB	0.65KB	0.38KB	128
REDOG-2	(44,8,37,2,83,18,3,12,2)	13.66KB	1.43KB	0.82KB	192
REDOG-3	(58,10,49,2,109,24,3,15,3)	31.87KB	2.50KB	1.44KB	256

To compare REDOG with other code-based algorithms such as HQC, BIKE, and Classic McEliece, all of which are based on the Hamming metric and advanced to the 4th round of the NIST PQC competition, we display their security level and the corresponding key sizes of these algorithms.

Note that HQC and BIKE algorithms have decryption failure which is a disadvantage although their key sizes are much smaller than REDOG. REDOG does not have a decryption failure. Classic McEliece has no decryption failure but has a large public key size of 1047KB at the 256 bits of security level while REDOG has a much smaller public key size of 31.87 KB. Therefore, REDOG is a strong competitor for HQC, BIKE, and Classic McEliece.

Table 2. Security level and key sizes of HQC [1]

Instance	pk size	sk size	ct size
hqc-128	2,249bytes	40bytes	4,481bytes
hqc-192	4,522bytes	40bytes	9,026bytes
hqc-256	7,245bytes	40bytes	14,469bytes

Table 3. Security level and key sizes of BIKE [3]

Quantity	Size	AES-128	AES-192	AES-256
Private key	$w[log_2(r)]$	2,130bits	2,296bits	4,384bits
Public key	n	20,326bits	43,786bits	65,498bits
Ciphertext	n	20,326bits	43,786bits	65,498bits

Table 4. Parameters, security level and key sizes of Classic McEliece [19]

Variant	n	m	t	$k = n - mt$	pk size	sk size	Security level
mceliece6960119	6960	13	119	5413	1047KB	13.6KB	256
mceliece8192128	8192	13	128	6528	1358KB	13.75KB	256

4 Security

This section covers various aspects related to the security of the REDOG cryptographic system. We describe issues such as the RSD problem, the DRSD problem arising from the configuration of matrices for encryption, and various decoding methods for Rank-metric codes. The presented results provide parameters satisfying security levels of 128, 192, and 256. Additionally, we introduce an alternative method proposed by Lange et al. for setting the rank weight t_1, t_2 of the error vector e_1, e_2.

4.1 Security definition

Problem 1 ([14]) (Rank syndrome decoding (RSD) Problem) Let H be a full rank $(n-k) \times n$ matrix over $\mathbb{F}_{q^m}$, $\mathbf{s} \in \mathbb{F}_{q^m}^{n-k}$, and r is an integer. The Rank Syndrome Decoding problem $\text{RSD}_H(q, m, n, k, r)$ is to determine a vector $x \in \mathbb{F}_{q^m}^{n}$ such that $\text{wt}_R(\mathbf{x}) = r$ and $\mathbf{s} = \mathbf{x}H^T$.

The RSD problem is analogous to the classical syndrome decoding problem in Hamming metric, which was shown to be an NP-complete problem. Gaborit and Zémor (2014) showed that if there were efficient probabilistic algorithms for solving the RSD problem, there exists an efficient probabilistic algorithm to solve the syndrome decoding problem in the Hamming metric.

Problem 2 ([12],[14]) Given a full rank $\ell \times n$ matrix M and a matrix $F = MH_1^T[H_2^T]^{-1}S$ where $[H_1 H_2]$ is a parity-check matrix for a Gabidulin code, and S is an invertible matrix. This problem is to distinguish F from R where R is a random $\ell \times (n-k)$ matrix over $\mathbb{F}_{q^m}$.

Problem 3 ([12],[14]) (Decisional rank syndrome decoding (DRSD) problem) Let H be a full rank $(n-k) \times n$ matrix over $\mathbb{F}_{q^m}$, $\mathbf{s} \in \mathbb{F}_{q^m}^{n-k}$ and r an integer. The Decisional Rank Syndrome Decoding problem $\mathrm{DRSD}_H(q,m,n,k,r)$ is to distinguish the distribution $(H,\mathbf{s})$ where $\mathbf{s} = \mathbf{x}H^T$ and $x \in \mathbb{F}_{q^m}^n$ such that $\mathrm{wt}_R(\mathbf{x}) = r$, from the distribution $(H,\mathbf{y})$ where $\mathbf{y}$ is a random vector in $\mathbb{F}_{q^m}^{n-k}$.

Problem 2 is a form of matrix factorization problem. The random invertible matrix S prevents Overbeck's attack from being used to attack Problem 2. Problem 3 is the decisional version of the RSD problem. Therefore, these two problems were suitable to be candidates for the hard problems with the modified DO.Gab-PKE is based on.

Theorem 2. *Theorem 2 ([12, Theorem 1]) The modified DO.Gab-PKE is IND-CPA secure under the assumptions of Problems 2 and 3.*

4.2 Security strength categories

The below information is in table 1.

To achieve 128 security in our cryptosystem, we need 4.38KB for $\mathsf{size}_{\mathsf{pk}}$, 0.65KB for $\mathsf{size}_{\mathsf{sk}}$, and 0.38KB for $\mathsf{size}_{\mathsf{ct}}$.

To achieve 192 security in our cryptosystem, we need 14.11KB for $\mathsf{size}_{\mathsf{pk}}$, 1.43KB for $\mathsf{size}_{\mathsf{sk}}$, and 0.82KB for $\mathsf{size}_{\mathsf{ct}}$.

To achieve 192 security in our cryptosystem, we need 32.66KB for $\mathsf{size}_{\mathsf{pk}}$, 2.50KB for $\mathsf{size}_{\mathsf{sk}}$, and 1.44KB for $\mathsf{size}_{\mathsf{ct}}$.

4.3 Cost of known attacks

1. $\mathsf{IND-CPA}$ security: REDOG achieves $\mathsf{IND-CPA}$ security. Kim et al. [12] have shown that the modified Do.Gab-PKE achieves $\mathsf{IND-CPA}$ security, and so does REDOG. The only difference is the secret matrix S. In REDOG, $S^{-1} \in \mathsf{GL}_{n-k}(\Lambda)$, distinguishing F from a random R is no longer an easy instance of Problem 2, in Lau et al. [13]. Thus, by Theorem 2 in Lau et al. [13], REDOG achieves $\mathsf{IND-CPA}$ security.

2. Key recovery attack : In the key equation $FS^{-1}H_2^T = MP^{-1}H_1^T$, there are $2(n-k)^2$ unknown variables of quadratic power and $n(n-k)$ unknown linear variables. Even if we rewrite the key equation over $\mathbb{F}_q$, there are a total of $(n-k)^2m + (n-k)m$ unknown variables of quadratic power and nm unknown linear variables. It is generally difficult to solve such equations, i.e., the complexity to solve for the solution is of high exponential power.

3. Our plaintext recovery attack: Rewrite the public key matrix

$$G_{\mathsf{pub}} = [M \mid MP^{-1}H_1^T[H_2^T]^{-1}S] = MP^{-1}[I_n \mid H_1^T[H_2^T]^{-1}] \begin{bmatrix} P & \mathbf{0} \\ \mathbf{0} & S \end{bmatrix}$$

   Although the matrix $[I_n \mid H_1^T[H_2^T]^{-1}]$ is a generator matrix for a Gabidulin code, the right scramble matrix $\begin{bmatrix} P & 0 \\ 0 & S \end{bmatrix}$ does not preserve the Frobenius invariant subspace. This implies that the matrix M is of full rank. Therefore, G_{pub} noes not generate an r-Frobenius weak code. Thus REDOG resists the Frobenius weak attack [4]. We perform simulations of the Frobenius weak attack on REDOG and the simulation result confirms that REDOG is secure against Frobenius weak attack.

4. Message recovery attacks.
   An adversary can try to recover the message by directly attacking the ciphertext. This is now an instance of the Rank Syndrome Decoding(RSD) problem, i.e., the problem of decoding a noisy codeword from a random code. The public code is generated by the $\ell \times (2n-k)$ matrix $(M|F)$ over $\mathbb{F}_{q^m}$. The error vector added to the ciphertext is chosen to have rank t. In this paper, we use the notation ℓ for the dimension and t for the error rank. Also, we denote the length by N. The complexity of algorithms in [6] and [8] also depends on the matrix multiplication exponent which we denote as ω.
   Description and cost of (Combinatorial attacks)-GRS algorithm The GRS [11] algorithm is a combinatorial attack on the rank decoding problem. The idea behind this algorithm is to guess a vector space containing the space spanned by the error vector. In this way, the received vector can be expressed in terms of the basis of the guessed space. The last step is to solve the linear system associated with the syndrome equations. This has complexity

$$\mathcal{O}\left((N-\ell)^3 m^3 q^{\min\{t\lfloor \ell m/N \rfloor, (t-1)\lfloor (\ell+1)m/N \rfloor\}}\right).$$

   The attack AGHT [5] is an improvement over the GRS combinatorial attack. The underlying idea is to guess the space containing the error in a specific way that provides a higher chance of guessing a suitable space. It has complexity

$$\mathcal{O}\left((N-\ell)^3 m^3 q^{t(\ell+1)m/N-m}\right).$$

   Description and cost of (Algebraic attack) The second attack, introduced in [11], which we denote GRS-alg, is an algebraic attack. Under the condition that $l > \lceil ((t+1)(\ell+1)-N-1)/t \rceil$ the decoding problem can be solved in

$$\mathcal{O}(t^3 l^3 q^{t(\lceil ((t+1)(\ell+1)-N-1)/t \rceil)}.$$

5. BBB+ attack.
   The BBB+ attack [6] translates the rank metric decoding problem into a system of multivariate equations and then uses Gröbner-basis methods to find solutions. Much of the analysis is spent on determining the degree of regularity, depending on the length, dimension, and rank of the code and error. If the condition $m\binom{N-\ell-1}{t}+1 \geqslant \binom{N}{t}$ is fulfilled then the problem can be solved in
   $$\mathcal{O}\left(\left(\frac{((m+N)t)^t}{t!}\right)^\omega\right).$$
   If the condition is not satisfied then the complexity of solving the decoding problem becomes
   $$\mathcal{O}\left(\left(\frac{((m+N)t)^{t+1}}{(t+1)!}\right)^\omega\right).$$
   or the same for $t+2$ in place of $t+1$. The authors of [6] use this in their calculations and thus we include this as well.

6. BBC+ attack.
   The paper [8] introduces 3 algorithms BBC+-Overdetermined, BBC+-Hybrid, and BBC+-SupportMinors. They make the use of extended linearization as a technique to compute Grobner bases.
   BBC+-Overdetermined case applies to the overdetermined case, which matches $m\binom{N-\ell-1}{t}+1 \geqslant \binom{N}{t}$, and permits to solve the system in
   $$\mathcal{O}\left(m\binom{N-\ell-1}{t}\binom{N}{t}^{\omega-1}\right).$$
   These costs match matrix computations on a matrix with $m\binom{N-\ell-1}{t}$ rows and $\binom{N}{t}$ columns.
   In the case of an undetermined system, BBC+-Hybrid is a hybrid attack that fixes some of the unknowns in a brute-force manner to produce an overdetermined system in the remaining variables. The costs are testing all possible values for j positions, where j is the smallest non-negative integer such that $m\binom{N-\ell-1}{t}+1 \geqslant \binom{N-j}{t}$, and for each performing the same matrix computations as in BBC on j columns less. This leads to a total complexity of
   $$\mathcal{O}\left(q^{jt}m\binom{N-\ell-1}{t}\binom{N-j}{t}^{\omega-1}\right)$$
   The brute-force part in BBC+-Hybrid quickly becomes the dominating factor. The BBC+-SupportMinors algorithm introduces terms of larger degrees first and then linearizes the system. This consists of multiplying the equations by some homogeneous monomials of degree b to obtain a system of homogeneous equations. However, for the special case of $q=2$ the equations in the system might not be homogeneous. In this case, homogeneous equations coming from smaller values of b are considered. To state the conditions for this and the next algorithms we first introduce some notation from [8].

$$A_b := \sum_{j=1}^{b} \binom{N}{t}\binom{m\ell+1}{j},$$

$$C_b := \sum_{j=1}^{b}\sum_{s=1}^{j} \left((-1)^{s+1}\binom{N}{t+s}\binom{m+s-1}{s}\binom{m\ell+1}{j-s}\right).$$

The degree of the equations formed in BBC+-SupportMinors depends on b, where $0 < b < 2 + t$ is minimal such that $A_b - 1 \leqslant C_b$ if such a b exists. In this case, the problem can be solved with complexity

$$\mathcal{O}((m\ell+1)(t+1)A_b^2).$$

The last two attacks are presented in [8] as the underlying approach has been pointed out to be incorrect in [7]. More precisely, [7] shows that the independence assumptions made in [8] are incorrect.

7. BBB+23 attack.
The Supportinors and MaxMinors modeling in [8] are not as independent as claimed, and [7] introduces a new approach BBB+23 that combines them while keeping independence, at least conjecturally and matched by experiments. They introduce the following notation:

$$\begin{aligned}
\mathcal{N}_b^{\mathbb{F}_q} &= \mathcal{N}_b^{\mathbb{F}_{q^m}} - \mathcal{N}_{b,syz}^{\mathbb{F}_q},\\
\mathcal{N}_b^{\mathbb{F}_{q^m}} &= \sum_{s=1}^{\ell}\binom{N-s}{t}\binom{\ell+b-1-s}{b-1} - \binom{N-\ell-1}{t}\binom{\ell-b-1}{b}\\
\mathcal{N}_{b,syz}^{\mathbb{F}_q} &= (m-1)\sum_{s=1}^{b}(-1)^{(s+1)}\binom{\ell+b-s-1}{b-s}\binom{N-\ell-1}{t+s}, and\\
\mathcal{M}_b^{\mathbb{F}_q} &= \binom{\ell+b-1}{b}\left(\binom{N}{t} - m\binom{N-\ell-1}{t}\right).
\end{aligned}$$

The problem can then be solved by linearization whenever $\mathcal{N}_b^{\mathbb{F}_q} \geqslant \mathcal{M}_b^{\mathbb{F}_q} - 1$. The complexity of solving the system is

$$T(m,N,l,t) = \mathcal{O}\left(m^2\mathcal{N}_b^{\mathbb{F}_q}(\mathcal{M}_b^{\mathbb{F}_q})^{\omega-1}\right).$$

Moreover, [7] introduces a hybrid strategy. Compared to BBC+-Hybrid it randomly picks matrices from $\mathsf{GL}_n(\mathbb{F}_q)$ to compute $\mathbb{F}_q$-linear combinations of the entries of the error vector and applies the same transformation to the generator matrix, hoping to achieve that the last a positions of the error vector are all 0 and then shortening the code while also reducing the dimension.
This technique has complexity

$$\min_{a\geqslant 0}(q^{ta}\cdot T(m, N-a, \ell-a, t))$$

Here, we post about the computational cost for each attack. It is computed by the fixed sage code. The lowest cost showed as blue color.

Table 5. Security level and cost for each parameter of REDOG

Instance	$(n, k, \ell, q, m, r, \lambda, t_1, t_2)$	$AGHT$	GRS	$BBB+$	$BBC+$	$BBB+23$	Security level
REDOG-1	(30,6,25,2,59,12,3,6,2)	198.41	226.01	140.05	144.99	145.79	128
REDOG-2	(44,8,37,2,83,18,3,12,2)	423.23	462.03	229.45	358.13	357.75	192
REDOG-3	(58,10,49,2,109,24,3,15,3)	749.01	800.30	324.24	666.09	672.91	256

4.4 Performance of reference implementation

Lange et al.[17] uploaded their sage code about the computation of the cost for known attacks. But there are some mistakes in the version released on August 9, 2023, so we just fixed those things. In their paper, they write about the C_b as follows:

$$C_b := \sum_{j=1}^{b} \left((-1)^{i+1} \binom{N}{t+s} \binom{m+s-1}{s} \binom{m\ell+1}{j-s} \right).$$

Moreover, at their sage code, they wrote the $(-1)^{i+1}$ part to $(-1)^s + i$, so the computation was wrong. To make this right, we fix $(-1)^{i+1}$ to $(-1)^{s+1}$ which we checked from [8], and edit the sage code also. After doing this process, the computational cost that they posted was changed. So here, we repost the corrected costs for each security. And also, the document [17] was corrected similarly.

After that,

After the fix of λ and the error weight $t_1 = \mathrm{wt}_R(e_1), t_2 = \mathrm{wt}_R(e_2)$, the costs go higher so we obtained calculation results show that smaller parameters than the existing parameters can be used to satisfy a given security level. And we checked about the attacks on [7]. But the cost of BBB+ is the lowest in every case as in the table 5.

5 Performance analysis

The public key size for REDOG is larger than the public key size for the modified DO.Gab-PKE because we have to choose S specifically so that REDOG can be secure. In what follows, we describe the relation between parameters. As the rank of the error is now t instead of r, the error correcting capability r has to increase, increasing the value of $n - k$. Moreover, for H to be a parity-check matrix of a $[2n-k, n]$-Gabidulin code, it is required that $m \geqslant 2n-k$. Moreover, the parameter ℓ is always larger than or equal to $n - k$. As $n - k$ increases, the values for m and ℓ increase. Therefore, the public key size is larger than the public key of the modified DO.Gab-PKE.

5.1 Description of platform

To implement our REDOG cryptosystem, we used the following software and hardware platforms:

· Windows 10 Pro
· Visual Studio 2022
· Intel(R) Core(TM) i5-10500 CPU @ 3.10GZ 3.10GHz
· GTX 960
· 64GB RAM

The performance results of implementing the REDOG cryptosystem using the platform described above are as follows.

Table 6. Performance of REDOG

Instance	$(n, k, \ell, q, m, r, \lambda, t_1, t_2)$	$\mathsf{KeyGen}_{\mathsf{time}}$	$\mathsf{Enc}_{\mathsf{time}}$	$\mathsf{Dec}_{\mathsf{time}}$	Security level
REDOG-1	(30,6,25,2,59,12,3,6,2)	1.6 sec	0.00004 sec	0.270 sec	128
REDOG-2	(44,8,37,2,83,18,3,12,2)	3.5 sec	0.00006 sec	0.565 sec	192
REDOG-3	(58,10,49,2,109,24,3,15,3)	7.2 sec	0.0001 sec	1.160 sec	256

6 Summary or Conclusion

We have enhanced the code-based cryptosystem REDOG, which utilizes the rank metric, presented in the first round of the KpqC competition. In the 4th round of the NIST PQC competition, four algorithms—BIKE, HQC, Classic McEliece, and SIKE—were selected. Among these, the first three algorithms are code-based cryptosystems, all constructed using codes based on the Hamming metric. In contrast, REDOG employs the rank metric instead of the Hamming metric, which can enhance the speed and efficiency of the cryptographic system. Ongoing research into rank metric codes continues, and the principal investigator and several co-authors have conducted extensive research and presentations on code-based cryptosystems utilizing the rank metric in various academic journals and conferences. Comparative analysis reveals that the parameters of REDOG are significantly favorable compared to the aforementioned algorithms. Furthermore, we thoroughly investigated BBB+ and BBC+ attacks, which were not considered when proposing this algorithm for the first round of the KpqC competition, addressing issues presented in that version such as problems with the setting of errors e_1 and e_2 and the selection of secret key S. By appropriately modifying the existing REDOG system and conducting additional calculations, we obtained superior parameters compared to previous proposals. Therefore, we are confident in REDOG's potential as a robust contender for KpqC standardization.

References

1. C. Aguilar Melchor, et al. “Hamming quasi-cyclic (HQC)” NIST PQC Round 2.4 (2018): 13.
2. C. Aguilar-Melchor, A. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.C. Deneuville, P. Gaborit, A. Hauteville, G. Zémor, Ouroboros-R. http://pqc-ourob orosr .org (2017). Accessed 8 Dec 2019.
3. N. Aragon, et al. “BIKE: bit flipping key encapsulation.” (2017).
4. A.-L. Horlemann-Trautmann, K. Marshall, and J. Rosenthal. “Considerations for rank-based cryptosystems.” 2016 IEEE International Symposium on Information Theory (ISIT). Ieee, 2016.
5. N. Aragon, P. Gaborit, A. Hauteville, J.-P. Tillich, A new algorithm for solving the rank syndrome decoding problem. In: Proceedings of IEEE International Symposium on Information Theory (ISIT 2018), pp. 2421–2425 (2018).
6. M. Bardet, P. Briaud, M. Bros, P. Gaborit, V. Neiger, O. Ruatta, and J.-P. Tillich. An algebraic attack on rank metric code-based cryptosystems. In Anne Canteaut and Yuval Ishai, editors, EUROCRYPT 2020, Part III, volume 12107 of LNCS, pages 64–93, Zagreb, Croatia, May 10–14, Springer, Heidelberg, Germany. (2020).
7. M. Bardet, P. Briaud, M. Bros, P. Gaborit, and J.-P. Tillich. Revisiting algebraic attacks on minrank and on the rank decoding problem. Designs, Codes and Cryptography, pages 1–37, 07 (2023).
8. M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R. A. Perlner, D. Smith-Tone, J.-P. Tillich, and J. A. Verbel. Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In Shiho Moriai and Huaxiong Wang, editors, ASIACRYPT 2020, Part I, volume 12491 of LNCS, pages 507–536, Daejeon, South Korea, December 7–11, Springer, Heidelberg, Germany. (2020).
9. E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. Ideals over a non-commutative ring and their application in cryptology. In Workshop on the Theory and Application of Cryptographic Techniques, 482-489. Springer, 1991.
10. P. Gaborit, L. Galvez, A. Hauteville, J.-L. Kim, M.J. Kim, Y.-S. Kim, Dual-Ouroboros: an improvement of the McNie scheme. Adv. Math. Commun. (2019).
11. P. Gaborit, O. Ruatta, and J. Schrek. On the complexity of the rank syndrome decoding problem. IEEE Transactions on Information Theory, 62(2):1006–1019, (2016).
12. J.-L. Kim, Y.-S. Kim, L.E. Galvez, M.J. Kim, A modified Dual-Ouroboros public-key encryption using Gabidulin codes. Appl. Algebra Eng. Commun. Comput. 32, 147—156, (2021).
13. T.S.C. Lau, C.H. Tan, New rank codes based encryption scheme using partial circulant matrices. Des. Codes Cryptogr. 87(12), 2979—2999, (2019).
14. T. S. C. Lau, C. H. Tan, T. F. Prabowo, On the security of the modified Dual-ouroboros PKE using Gabidulin codes, Appl. Algebra Eng. Commun. Comput. 32, 681–699, (2021).
15. P. Loidreau, A Welch–Berlekamp like algorithm for decoding Gabidulin codes. In: Proceedings of the International Workshop on Coding and Cryptography (WCC 2005), pp. 36–45, (2005).
16. P. Loidreau, A new rank metric codes based encryption scheme, 8th International Conference on Post-Quantum Cryptography, PQCrypto 2017, May 2017, Utrecht, France.
17. T. Lange, A. Pellegrini, A. Ravagnani, On the security of REDOG, Cryptology ePrint Archive, Paper 2023/1205 (2023).

18. R. Overbeck, Structural attacks for public key cryptosystems based on Gabidulin codes. J. Cryptol. 21(2), 280–301, (2008).
19. H. Singh, “Code based cryptography: Classic mceliece,” arXiv preprint arXiv:1907.12754 (2019).
20. D. H. Wiedemann. Solving sparse linear equations over finite fields. IEEE Trans. Inf. Theory, 32(1):54–62, (1986).
21. McNie and other cryptosystems, https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization. Accessed 21 Nov 2019.
22. McNie comment from https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1- Submissions, read official comments on McNie dated Dec 24, 2017 and (Dec. 26, 2017).