On the security of REDOG

Tanja Lange, Alex Pellegrini, and Alberto Ravagnani

Eindhoven University of Technology, the Netherlands
tanja@hyperelliptic.org, alex.pellegrini@live.com, a.ravagnani@tue.nl

Abstract. We analyze REDOG, a public-key encryption system submitted to the Korean competition on post-quantum cryptography. REDOG is based on rank-metric codes. We prove its incorrectness and attack its implementation, providing an efficient message recovery attack. Furthermore, we show that the security of REDOG is much lower than claimed. We then proceed to mitigate these issues and provide two approaches to fix the decryption issue, one of which also leads to better security.

Keywords: post-quantum crypto, code-based- crypto, rank-metric codes

1 Introduction

This paper analyzes the security of the REinforced modified Dual-Ouroboros based on Gabidulin codes, REDOG [KHL+22a], a public-key encryption system submitted to KpqC, the Korean competition on post-quantum cryptography. REDOG is a code-based cryptosystem using rank-metric codes, aiming at providing a rank-metric alternative to Hamming-metric code-based cryptosystems.

Rank-metric codes were introduced by Delsarte [Del78] and independently rediscovered by Gabidulin [Gab85] in 1985, who focused on those that are linear over a field extension. Gabidulin, Paramonov, and Tretjakov [GPT91] proposed their use for cryptography in 1991. The GPT system was attacked by Overbeck [Ove05,Ove08] who showed *structural* attacks, permitting recovery of the private key from the public key.

During the mid 2010s new cryptosystems using rank-metric codes were developed such as Ouroboros [DGZ17] and the first round of the NIST competition on post-quantum cryptography saw 5 systems based on rank-metric codes: LAKE [ABD+17a], LOCKER [ABD+17b], McNie [GKK+17], Ouroboros-R [AAB+17a]. RQC [AAB+17b]. For further information about all these systems

Author list in alphabetical order; see https://www.ams.org/profession/leaders/culture/CultureStatement04.pdf. This work was funded in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC 2092 CASA—390781972 "Cyber Security in the Age of Large-Scale Adversaries" and by the Netherlands Organisation for Scientific Research (NWO) under grants OCENW.KLEIN.539 and VI.Vidi.203.045. Date: 2023.11.14. For the full version see [LPR23].

see NIST's Round-1 Submissions page. Gaborit announced an attack weakening McNie and the McNie authors adjusted their parameters. A further attack was published in [LT18] and NIST did not advance McNie into the second round of the competition.

ROLLO, a merger of LAKE, LOCKER and Ouroboros-R, and RQC made it into the the second round but got broken near the end of it by significant advances in the cryptanalysis of rank-metric codes and the MinRank problem in general, see [BBB+20] and [BBC+20]. In their report at the end of round 2 [AASA+20], NIST wrote an encouraging note on rank-metric codes: "Despite the development of algebraic attacks, NIST believes rank-based cryptography should continue to be researched. The rank metric cryptosystems offer a nice alternative to traditional hamming metric codes with comparable bandwidth." (capitalization as in the original).

Kim, Kim, Galvez, and Kim [KKGK21] proposed a new rank-metric system in 2021 which was then analyzed by Lau, Tan, and Prabowo in [LTP21] who also proposed some modifications to the issues they found. REDOG closely resembles the system in [LTP21] and uses the same parameters.

Our contribution In this paper we expose weaknesses of REDOG and show that the system, as described in the documentation, is incorrect. To start with, we prove that REDOG does not decrypt correctly. The documentation and [LTP21] contain an incorrect estimate of the rank of an element which causes the input to the decoding step to have too large rank. The system uses Gabidulin codes [Gab85] which are MRD (Maximum Rank Distance) codes, meaning that vectors with errors of rank larger than half the minimum distance will decode to a different codeword, thus causing incorrect decryption in the REDOG system.

As a second contribution we attack ciphertexts produced by REDOG's reference implementation. We show that we can use techniques from the Hamming metric to obtain a message-recovery attack. This stems from a choice in the implementation which avoids the above-mentioned decryption problem. However, the errors introduced in the ciphertext have a specific shape which allows us to apply basic techniques of Information Set Decoding (ISD) over the Hamming metric to recover the message in seconds.

As a third contribution, we show that, independently of the special choice of error vectors in the implementation, the security of the cryptosystem is lower than the claimed security level. The main effect comes from a group of attacks published in [BBC+20] which the REDOG designers had not taken into account. An smaller effect comes from a systematic scan through all attack parameters.

Finally, we provide two ways to make REDOG's decryption correct. The first is a minimal change to fix the system by changing the space from which some matrix P^{-1} is chosen in a way that differs from the choice in REDOG and avoids the issue mentioned above. However, this still requires choosing much larger parameters to deal with our third contribution. The second way makes a different change to REDOG which improves the resistance to attacks while also fixing the decryption issue. We show that, using this strategy, not only are

REDOG's parameters sufficient to reach any claimed security level, but they provide security abundantly beyond each level, allowing room for an eventual optimization. Note, however, that these estimates are obtained from big-$\mathcal{O}$ complexity estimates, putting all constants to 1 and lower-order terms to 0, and thus underestimate the security.

2 Preliminaries and background notions

This section gives the necessary background on rank-metric codes for the rest of the paper. Let $\{\alpha_1, \ldots, \alpha_m\}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Write $x \in \mathbb{F}_{q^m}$ uniquely as $x = \sum_{i=1}^m X_i \alpha_i$, $X_i \in \mathbb{F}_q$ for all i. So x can be represented as $(X_1, \ldots, X_m) \in \mathbb{F}_q^m$. We will call this the *vector representation* of x. Extend this process to $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{F}_{q^m}^n$ defining a map $\mathsf{Mat} : \mathbb{F}_{q^m}^n \to \mathbb{F}_q^{m \times n}$ by:

$$\mathbf{v} \mapsto \begin{bmatrix} V_{11} & V_{21} & \ldots & V_{n1} \\ V_{12} & V_{22} & \ldots & V_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ V_{1m} & V_{2m} & \ldots & V_{nm} \end{bmatrix}.$$

Definition 2.1. *The rank weight of* $\mathbf{v} \in \mathbb{F}_{q^m}^n$ *is defined as* $\mathsf{wt}_R(\mathbf{v}) := \mathsf{rk}_q(\mathsf{Mat}(\mathbf{v}))$ *and the rank distance between* $\mathbf{v}, \mathbf{w} \in \mathbb{F}_{q^m}^n$ *is* $d_R(\mathbf{v}, \mathbf{w}) := \mathsf{wt}_R(\mathbf{v} - \mathbf{w})$.

Remark 2.2. It can be shown that the rank distance does not depend on the choice of the basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. In particular, the choice of the basis is irrelevant for the results in this document.

When talking about the space spanned by $\mathbf{v} \in \mathbb{F}_{q^m}^n$, denoted as $\langle \mathbf{v} \rangle$, we mean the $\mathbb{F}_q$-subspace of $\mathbb{F}_q^m$ spanned by the columns of $\mathsf{Mat}(\mathbf{v})$.

For completeness, we introduce the Hamming weight and the Hamming distance. These notions will be used in our message recovery attack against REDOG's implementation.

The *Hamming weight* of a vector $\mathbf{v} \in \mathbb{F}_{q^m}^n$ is defined as $\mathsf{wt}_H(\mathbf{v}) := \#\{i \in \{1, \ldots, n\} \mid v_i \neq 0\}$ and the Hamming distance between vectors $\mathbf{v}, \mathbf{w} \in \mathbb{F}_{q^m}^n$ is defined as $d_H(\mathbf{v}, \mathbf{w}) := \mathsf{wt}_H(\mathbf{v} - \mathbf{w})$.

Let $D = d_R$ or $D = d_H$. Then an $[n, k, d]$-code C with respect to D over $\mathbb{F}_{q^m}$ is a k-dimensional $\mathbb{F}_{q^m}$-linear subspace of $\mathbb{F}_{q^m}^n$ with *minimum distance*

$$d := \min_{\mathbf{a}, \mathbf{b} \in C,\, \mathbf{a} \neq \mathbf{b}} D(\mathbf{a}, \mathbf{b})$$

and *correction capability* $\lfloor (d-1)/2 \rfloor$. If $D = d_R$ (resp. $D = d_H$) then the code C is also called a *rank-metric* (resp. *Hamming-metric*) code. All codes in this document are linear over the field extension $\mathbb{F}_{q^m}$.

We say that G is a *generator matrix* of C if its rows span C. We say that H is a *parity check matrix* of C if C is the right-kernel of H.

A very well-known family of rank metric codes are *Gabidulin codes* [Gab85], which have $d = n - k + 1$.

In this paper we can mostly use these codes as a black box, knowing that there is an efficient decoding algorithm using the parity-check matrix of the code and decoding vectors with errors of rank up to $\lfloor (d-1)/2 \rfloor$.

A final definition necessary to understand REDOG is that of isometries.

Definition 2.3. *Consider vectors in* $\mathbb{F}_{q^m}^n$. *An* isometry with respect to the rank metric *is a matrix* $P \in \mathsf{GL}_n(\mathbb{F}_{q^m})$ *satisfying that* $\mathsf{wt}_R(\mathbf{v}P) = \mathsf{wt}_R(\mathbf{v})$ *for any* $\mathbf{v} \in \mathbb{F}_{q^m}^n$.

Obviously matrices $P \in \mathsf{GL}_n(\mathbb{F}_q)$ are isometries as $\mathbb{F}_q$-linear combinations of the coordinates of $\mathbf{v}$ do not increase the rank and the rank does not decrease as P is invertible. The rank does also not change under scalar multiplication by some $\alpha \in \mathbb{F}_{q^m}^*$: $\mathsf{wt}_R(\alpha\mathbf{v}) = \mathsf{wt}_R(\mathbf{v})$. Note that the latter corresponds to multiplication by $P = \alpha I_n$.

Berger [Ber03] showed that any isometry is obtained by composing these two options.

Theorem 2.4. *[Ber03, Theorem 1] The isometry group of* $\mathbb{F}_{q^m}^n$ *for the rank metric is generated by scalar multiplications by elements in* $\mathbb{F}_{q^m}^*$ *and elements of* $\mathsf{GL}_n(\mathbb{F}_q)$. *This group is isomorphic to the product group* $\left(\mathbb{F}_{q^m}^*/\mathbb{F}_q^*\right) \times \mathsf{GL}_n(\mathbb{F}_q)$.

3 System specification

This section introduces the specification of REDOG. We follow the notation of [LTP21], with minor changes.

The system parameters are positive integers $(n, k, \ell, q, m, r, \lambda, t)$, with $\ell < n$ and $\lambda t \leq r \leq \lfloor (n-k)/2 \rfloor$, as well as a hash function $\mathsf{hash} : \mathbb{F}_{q^m}^{2n-k} \to \mathbb{F}_{q^m}^{\ell}$.

KeyGen:

1. Select $H = (H_1 \mid H_2)$, $H_2 \in \mathsf{GL}_{n-k}(\mathbb{F}_{q^m})$, a parity check matrix of a $[2n-k, n]$ Gabidulin code, with syndrome decoder Φ correcting r errors.
2. Select a full rank matrix $M \in \mathbb{F}_{q^m}^{\ell \times n}$ and isometry $P \in \mathbb{F}_{q^m}^{n \times n}$ (w.r.t. the rank metric).
3. Select a λ-dimensional subspace $\Lambda \subset \mathbb{F}_{q^m}$, seen as $\mathbb{F}_q$-linear space, containing 1 and select $S^{-1} \in \mathsf{GL}_{n-k}(\Lambda)$; see Section 4 for the definition.
4. Compute $F = MP^{-1}H_1^T \left(H_2^T\right)^{-1} S$ and publish the public key $\mathsf{pk} = (M, F)$. Store the secret key $\mathsf{sk} = (P, H, S, \Phi)$.

Encrypt ($\mathbf{m} \in \mathbb{F}_{q^m}^{\ell}, \mathsf{pk}$)

1. Generate uniformly random $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2) \in \mathbb{F}_{q^m}^{2n-k}$ with $\mathsf{wt}_R(\mathbf{e}) = t$, $\mathbf{e}_1 \in \mathbb{F}_{q^m}^n$ and $\mathbf{e}_2 \in \mathbb{F}_{q^m}^{n-k}$.
2. Compute $\mathbf{m}' = \mathbf{m} + \mathsf{hash}(\mathbf{e})$.
3. Compute $\mathbf{c}_1 = \mathbf{m}'M + \mathbf{e}_1$ and $\mathbf{c}_2 = \mathbf{m}'F + \mathbf{e}_2$ and send $(\mathbf{c}_1, \mathbf{c}_2)$.

Decrypt $((\mathbf{c}_1, \mathbf{c}_2), \mathsf{sk})$

1. Compute $\mathbf{c}' = \mathbf{c}_1 P^{-1} H_1^T - \mathbf{c}_2 S^{-1} H_2^T = \mathbf{e}' H^T$ where the vector $\mathbf{e}' := (\mathbf{e}_1 P^{-1}, -\mathbf{e}_2 S^{-1})$.
2. Decode $\mathbf{c}'$ using Φ to obtain $\mathbf{e}'$, recover $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$ using P and S.
3. Solve $\mathbf{m}'M = \mathbf{c}_1 - \mathbf{e}_1$. Output $\mathbf{m} = \mathbf{m}' - \mathsf{hash}(\mathbf{e})$.

Suggested parameters We list the suggested parameters of REDOG for 128,192 and 256 bits of security, following [KHL+22a] submitted to KpqC.

Security parameter	$(n, k, \ell, q, m, r, \lambda, t)$
128	$(44, 8, 37, 2, 83, 18, 3, 6)$
192	$(58, 10, 49, 2, 109, 24, 3, 8)$
256	$(72, 12, 61, 2, 135, 30, 3, 10)$

Table 1. Suggested parameters; see [KHL+22a].

4 Incorrectness of decryption

This section shows that decryption typically fails for the version of REDOG specified in [KHL+22a,LTP21]. The novelty of this specification, compared to that introduced in [KKGK21], lies in the selection of the invertible matrix S^{-1} in Step 3, which is selected with the property that $S^{-1} \in \mathsf{GL}_{n-k}(\Lambda)$, where Λ is a λ-dimensional $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$. This method has been first proposed by Loidreau in [Loi17], but it appears to be incorrectly applied in REDOG. Before providing more details about this claim and proving the incorrectness of REDOG's decryption process, we will shed some light on the object $\mathsf{GL}_{n-k}(\Lambda)$. Unlike the notation suggests, this is not a group, but a potentially unstructured subset of $\mathsf{GL}_{n-k}(\mathbb{F}_{q^m})$ defined as follows:

Let $\{1, \alpha_2, \ldots, \alpha_\lambda\} \subset \mathbb{F}_{q^m}$ be a set of elements that are $\mathbb{F}_q$-linearly independent. Let $\Lambda \subset \mathbb{F}_{q^m}$ be the set of $\mathbb{F}_q$-linear combinations of these α_i's. This set forms an $\mathbb{F}_q$-linear vectorspace. Now, $S^{-1} \in \mathsf{GL}_{n-k}(\Lambda)$ is defined to mean that S is an invertible $(n-k) \times (n-k)$ matrix with the property that the entries of S^{-1} are elements of Λ. Note that such an S exists because $\lambda \geq 1$ by assumption. The REDOG documentation [KHL+22a] points out that this does not imply that $S \in \mathsf{GL}_{n-k}(\Lambda)$, hence, despite what the notation may suggest, $\mathsf{GL}_{n-k}(\Lambda)$ is not a group in general.

We continue by giving a proof, and an easy generalization for any q, of [Loi17, Proposition 1].

Proposition 4.1. *Let λ, t, n be positive integers such that $\lambda t \leq n$, $A \in \mathsf{GL}_n(\Lambda)$ where $\Lambda \subset \mathbb{F}_{q^m}$ is a λ-dimensional subspace of $\mathbb{F}_{q^m}$, and $\mathbf{x} \in \mathbb{F}_{q^m}^n$ with $\mathsf{wt}_R(\mathbf{x}) = t$. Then*

$$\mathsf{wt}_R(\mathbf{x}A) \leq \lambda t.$$

Proof. Let Γ be the subspace of $\mathbb{F}_{q^m}$ generated by the entries of $\mathbf{x} = (x_1, \ldots, x_n)$. Since Γ has dimension t, we can write $\Gamma = \langle y_1, \ldots, y_t \rangle$ with $y_i \in \mathbb{F}_{q^m}$. Similarly for Λ, we can write $\Lambda = \langle \alpha_1, \ldots, \alpha_\lambda \rangle$ with $\alpha_i \in \mathbb{F}_{q^m}$. Express $\mathbf{x}A$ as

$$\mathbf{x}A = \left(\sum_{i=1}^{n} x_i A_{i,1}, \ldots, \sum_{i=1}^{n} x_i A_{i,n} \right).$$

Fix $j \in \{1, \dots, n\}$. Then

$$(\mathbf{x}A)_j = \sum_{i=1}^{n} x_i A_{i,j} = \sum_{i=1}^{n} \left(\left(\sum_{h=1}^{t} x_{i,h} y_h \right) \left(\sum_{k=1}^{\lambda} A_{i,j,k} \alpha_k \right) \right),$$

with $x_{i,h}, A_{i,j,k} \in \mathbb{F}_q$. By rearranging the terms we obtain

$$(\mathbf{x}A)_j = \sum_{h=1}^{t} \sum_{k=1}^{\lambda} \left(\sum_{i=1}^{n} x_{i,h} A_{i,j,k} \right) y_h \alpha_k. \tag{1}$$

Therefore each entry of $\mathbf{x}A$ can be expressed as an $\mathbb{F}_q$-linear combination of the λt elements of the form $y_h \alpha_k$. □

We will now show that REDOG typically does not decrypt correctly. In order to do so, we need some preliminary results and tools. The proof of the next lemma uses some tools from combinatorics. It computes the probability that a randomly selected t-tuple of elements of a t-dimensional vector space spans the entire space.

Lemma 4.2. *Let V be a t-dimensional subspace $V \subseteq \mathbb{F}_q^m$ and let $S \in V^s$ be a uniformly random s-tuple of elements of V. The probability $p(q, s, t)$ that $\langle S_i \mid i \in \{1, \dots, s\} \rangle = V$ is 0 if $0 \leq s < t$ and*

$$p(q, s, t) = \sum_{i=0}^{t} \begin{bmatrix} t \\ i \end{bmatrix}_q (-1)^{t-i} q^{s(i-t) + \binom{t-i}{2}} \tag{2}$$

otherwise, where $\begin{bmatrix} t \\ i \end{bmatrix}_q$ is the q-binomial coefficient, counting the number of subspaces of dimension i of $\mathbb{F}_q^t$, and $\binom{a}{b} = 0$ for $a < b$. In particular, this probability does not depend on m or on the choice of V, but only on its dimension.

Proof. Let $(\mathcal{P}, \subseteq)$ be the poset (partially ordered set) of subspaces of $\mathbb{F}_q^m$ ordered by inclusion. Recall that the Möbius function of $\mathcal{P}$, and of any finite poset, is defined, for $A, B \in \mathcal{P}$, as

$$\mu(B, A) = \begin{cases} 1 & \text{if } B = A, \\ -\sum_{C \mid B \subseteq C \subset A} \mu(B, C) & \text{if } B \subset A, \\ 0 & \text{otherwise.} \end{cases}$$

For the poset of subspaces, the Möbius function is computed e.g. in [Sta11, Example 3.10.2] as

$$\mu(B, A) = \begin{cases} (-1)^k q^{\binom{k}{2}} & \text{if } B \subseteq A \text{ and } \dim(A) - \dim(B) = k, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We want to compute the function $f : \mathcal{P} \to \mathbb{N}$ defined as

$$f(A) = \# \left\{ S \in \left(\mathbb{F}_q^m \right)^s \mid \langle S \rangle = A \right\}.$$

Clearly, if $s < \dim A$, there does not exist any s-tuple S spanning A, hence $f(A) = 0$, which gives the first case of (2). We can therefore restrict ourselves to the case $s \geq \dim A$. Define the auxiliary function $g : \mathcal{P} \to \mathbb{N}$ as

$$\begin{aligned} g(A) &= \sum_{B \subseteq A} f(B) \\ &= \#\left\{S \in \left(\mathbb{F}_q^m\right)^s \mid \langle S \rangle \subseteq A\right\} \\ &= |A|^s = q^{s \dim A}. \end{aligned}$$

Then by Möbius inversion we can compute:

$$f(A) = \sum_{B \subseteq A} g(B)\mu(B, A). \tag{4}$$

Splitting the sum over the dimensions, and substituting the values in Equation 3, we can obtain

$$\begin{aligned} f(V) &= \sum_{i=0}^{t} \sum_{U \subseteq V,\, \dim U = i} g(U)\mu(U, V) \\ &= \sum_{i=0}^{t} q^{si}(-1)^{t-i} q^{\binom{t-i}{2}} \sum_{U \subseteq V,\, \dim U = i} 1 \\ &= \sum_{i=0}^{t} \begin{bmatrix} t \\ i \end{bmatrix}_q (-1)^{t-i} q^{si + \binom{t-i}{2}}. \end{aligned}$$

The probability can be computed by dividing $f(V)$ by the number of s-tuples of elements of V, that is, q^{st}. □

Remark 4.3. The probability given in Lemma 4.2 can be interpreted as the ratio of the number of surjective linear maps from $\mathbb{F}_q^s$ onto $\mathbb{F}_q^t$ over the total number of linear maps.

We next compute the probability that by truncating a rank t vector, the rank stays the same.

Theorem 4.4. *Let $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2) \in \mathbb{F}_{q^m}^{2n-k}$, with $\mathbf{e}_1 \in \mathbb{F}_{q^m}^{n}$ and $\mathbf{e}_2 \in \mathbb{F}_{q^m}^{n-k}$, be a uniformly random error with $\mathsf{wt}_R(\mathbf{e}) = t$. Then $\mathsf{wt}_R(\mathbf{e}_1) = t$ and $\mathsf{wt}_R(\mathbf{e}_2) = t$ with probability $p(q, n, t)/p(q, 2n-k, t)$ and $p(q, n-k, t)/p(q, 2n-k, t)$ respectively.*

Proof. By definition, the probability that $\mathsf{wt}_R(\mathbf{e}_1) = t$ is the ratio

$$\pi = \frac{\#\{\mathbf{e} \in \mathbb{F}_{q^m}^{2n-k} \mid \mathsf{wt}_R(\mathbf{e}) = t \text{ and } \mathsf{wt}_R(\mathbf{e}_1) = t\}}{\#\{\mathbf{e} \in \mathbb{F}_{q^m}^{2n-k} \mid \mathsf{wt}_R(\mathbf{e}) = t\}}. \tag{5}$$

We can split the cardinalities above over all the subspaces of $\mathbb{F}_q^m$ of dimension t as follows:

$$\pi = \frac{\sum_{V\subset\mathbb{F}_q^m,\,\dim V=t} \#\{\mathbf{e}\in\mathbb{F}_{q^m}^{2n-k} \mid \langle\mathbf{e}\rangle = \langle\mathbf{e}_1\rangle = V\}}{\sum_{V\subset\mathbb{F}_q^m,\,\dim V=t} \#\{\mathbf{e}\in\mathbb{F}_{q^m}^{2n-k} \mid \langle\mathbf{e}\rangle = V\}}. \tag{6}$$

It is not hard to prove that the summands in (4) are independent of the space V. Therefore

$$\pi = \frac{\#\{\mathbf{e}\in\mathbb{F}_{q^m}^{2n-k} \mid \langle\mathbf{e}\rangle = \langle\mathbf{e}_1\rangle = V\}}{\#\{\mathbf{e}\in\mathbb{F}_{q^m}^{2n-k} \mid \langle\mathbf{e}\rangle = V\}} = \frac{\#\{\mathbf{e}_1\in\mathbb{F}_{q^m}^{n} \mid \langle\mathbf{e}_1\rangle = V\}\, q^{t(n-k)}}{\#\{\mathbf{e}\in\mathbb{F}_{q^m}^{2n-k} \mid \langle\mathbf{e}\rangle = V\}},$$

where V is any subspace of $\mathbb{F}_q^m$ of dimension t. By applying Lemma 4.2 we then get

$$\pi = \frac{p(q,n,t)\, q^{nt} q^{t(n-k)}}{p(q,2n-k,t)\, q^{(2n-k)t}} = \frac{p(q,n,t)}{p(q,2n-k,t)},$$

as claimed. The probability for $\mathbf{e}_2$ can be computed with the same arguments as for $\mathbf{e}_1$. □

Remark 4.5. In the context of a REDOG instance, the data q, n and t is fixed, hence, for the sake of reading simplicity, we denote the probability given in Theorem 4.4 by

$$\bar{p}(r,t) = \frac{p(q,r,t)}{p(q,2n-k,t)}.$$

Example 4.6. Consider the suggested parameters of REDOG for 128 bits of security from Table 1. Using SageMath [S+21] we computed the probability that $\mathsf{wt}_R(\mathbf{e}_1) = t$, that is $\bar{p}(44,6) = 0.999999999996419$, and the probability that $\mathsf{wt}_R(\mathbf{e}_2) = t$, that is $\bar{p}(36,6) = 0.999999999083229$.

We are ready to state the following theorem, which directly implies that REDOG's decryption process fails with extremely high probability.

Theorem 4.7. *Let (n,k,q,m,λ,t) be integers with $k<n<m$ and $\lambda t \le m$. Let $\Lambda \subset \mathbb{F}_{q^m}$ be a λ-dimensional subspace of $\mathbb{F}_{q^m}$ and $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$ as in Theorem 4.4. Let $P \in \mathbb{F}_{q^m}^{n\times n}$ be a random isometry matrix (w.r.t. the rank metric) and $S^{-1} \in \mathsf{GL}_{n-k}(\Lambda)$. Then $\mathbf{e}' := \left(\mathbf{e}_1 P^{-1}, -\mathbf{e}_2 S^{-1}\right)$ has rank weight $\mathsf{wt}_R(\mathbf{e}') \ge \lambda t + 1$ with probability bounded from below by*

$$p_{\mathsf{fail}}(n,k,q,m,\lambda,t) := \bar{p}(n,t)\,\bar{p}(n-k,\lambda t)\,\bar{p}(n-k,t)\left(1 - \begin{bmatrix}\lambda t\\ t\end{bmatrix}_q \Big/ \begin{bmatrix}m\\ t\end{bmatrix}_q\right).$$

Proof. By Theorem 2.4, the isometry P is of the form $\alpha\bar{P}$ for $\alpha\in\mathbb{F}_{q^m}^*$ and $\bar{P}\in\mathsf{GL}_n(\mathbb{F}_q)$, where $q^m \gg q$ and thus typically $\alpha \notin \mathbb{F}_q$. Because of the multiplication by α^{-1}, we can assume that the linear transformation induced by P^{-1} takes a t-dimensional subvectorspace of $\mathbb{F}_q^m$ to a random t-dimensional subspace. Similarly we assume that S^{-1} sends a t-dimensional subspace of $\mathbb{F}_q^m$ to a random subspace of dimension at most λt, by Proposition 4.1. We get the lower bound on the failure probability by showing the following:

1. $\mathsf{wt}_R(\mathbf{e}_1 P^{-1}) = t$ with probability $\bar{p}(n,t)$;
2. $\mathsf{wt}_R(-\mathbf{e}_2 S^{-1}) = \lambda t$ with probability $\bar{p}(n-k,t)\bar{p}(n-k,\lambda t)$;
3. under the conditions in (1) and (2), $\langle \mathbf{e}_1 P^{-1}\rangle \not\subset \langle -\mathbf{e}_2 S^{-1}\rangle$ with probability $1 - \begin{bmatrix} \lambda t \\ t \end{bmatrix}_q / \begin{bmatrix} m \\ t \end{bmatrix}_q$.

Note that (1) follows directly from Theorem 4.4 and the fact that P is an isometry of the space w.r.t the rank metric.

Likewise, $\mathsf{wt}_R(-\mathbf{e}_2) = t$ with probability $\bar{p}(n-k,t)$. The proof of Proposition 4.1 shows that for $\mathbf{e}_2$ with $\mathsf{wt}_R(-\mathbf{e}_2) = t$ we have that $-\mathbf{e}_2 S^{-1}$ is contained in a λt-dimensional subspace of $\mathbb{F}_q^m$. Again by Theorem 4.4 we obtain that $\langle -\mathbf{e}_2 S^{-1}\rangle$ spans the entire space with probability $\bar{p}(n-k,\lambda t)$, proving (2).

To prove (3) we will compute the opposite, i.e. the probability that $\langle \mathbf{e}_1 P^{-1}\rangle$ is a subspace of $\langle -\mathbf{e}_2 S^{-1}\rangle$. As mentioned at the beginning of the proof, we treat $\langle \mathbf{e}_1 P^{-1}\rangle$ as a random t-dimensional subspace of $\mathbb{F}_{q^m}$. Thus we can compute this probability as the ratio between the number of t-dimensional subspaces of $\langle -\mathbf{e}_2 S^{-1}\rangle$ and of $\mathbb{F}_q^m$, that is, $\begin{bmatrix} \lambda t \\ t \end{bmatrix}_q / \begin{bmatrix} m \\ t \end{bmatrix}_q$.

Combining the probabilities and observing that $(1-3)$ imply $\mathsf{wt}_R(\mathbf{e}') \geq \lambda t+1$ gives the result. □

Remark 4.8. There are more ways to get $\mathsf{wt}_R(\mathbf{e}') \geq \lambda t + 1$ by relaxing the first two requirements in the proof of Theorem 4.7 and studying the dimension of the union in the third, but p_{fail} is large enough for the parameters in REDOG to prove the point.

Remark 4.9. The proof of property (3) relies on $\mathbf{e}_1 P^{-1}$ being a random subspace of dimension t. We note that for $\alpha \in \mathbb{F}_q$ we have $\langle \mathbf{e}_1\rangle = \langle \mathbf{e}_1 P^{-1}\rangle \subset \langle \mathbf{e}_2 S^{-1}\rangle$ for $S^{-1} \in \mathsf{GL}_{n-k}(\Lambda)$ and $1 \in \Lambda$. The latter constraint is stated in [KHL$^+$22a] and [LTP21] and it is possible that the authors were not aware of the full generality of isometries. See also the full version [LPR23] for further observations on [LTP21] which are consistent with this misconception.

Recall that the decoder Φ can only correct errors up to rank weight $r = \lambda t$. By Theorem 4.7 we have that $\mathbf{e}'$ has rank weight $\geq \lambda t + 1$, hence the following corollary.

Corollary 4.10. *Let $(n,k,\ell,q,m,r,\lambda,t)$ be the parameters of a instance of REDOG with $r = \lambda t$. Then REDOG will produce decryption failures with probability at least $p_{\mathsf{fail}}(n,k,q,m,\lambda,t)$.*

Note that a $[2n-k,n]$ Gabidulin code has minimum distance $d_R = 2n-k-n+1 = n-k+1$ and can thus correct at most $\lfloor (n-k)/2\rfloor$ errors and that all instances of REDOG in Table 1 satisfy $\lfloor (n-k)/2\rfloor = r = \lambda t$.

Example 4.11. As in Example 4.6, consider the suggested parameters for 128 bits of security. Then Theorem 4.7 states that $\mathsf{wt}_R(\mathbf{e}') \geq 19$ with probability at least $p_{\mathsf{fail}}(44,8,2,83,3,6) = \bar{p}(44,8)\bar{p}(36,6)\bar{p}(36,18)\left(1 - \begin{bmatrix} 18 \\ 6 \end{bmatrix}_2 / \begin{bmatrix} 83 \\ 6 \end{bmatrix}_2\right) =$ 0.999996184401789.

Table 2 reports the value of p_{fail} for each set of security parameters given in Table 1. This shows that REDOG's decryption process fails almost always.

Security parameter	p_{fail}
128	0.999996184401789
192	0.999999940394453
256	0.999999999068677

Table 2. Value of decryption failure probability p_{fail} per suggested parameters.

5 Message recovery attack on REDOG's implementation

Theorem 4.7 and the numerical examples show that, with probability almost 1, REDOG will fail decrypting. However, the probability is not exactly 1 and there exist some choices of $\mathbf{e}$ for which decryption still succeeds. One extreme way to avoid decryption failures, chosen in the refenrence implementation of REDOG, is to build errors as follows:

Algorithm 5.1 *(REDOG's error generator)*

```
1. Pick $\beta_1, \ldots, \beta_t \in \mathbb{F}_{q^m}$ being $\mathbb{F}_q$-linearly independent.
2. Pick random permutation $\pi$ on $2n-k$ symbols.
3. Set $\mathbf{e}_{\mathsf{init}} = (\beta_1, \ldots, \beta_t, 0, \ldots, 0) \in \mathbb{F}_{q^m}^{2n-k}$. Output $\mathbf{e} = \pi(\mathbf{e}_{\mathsf{init}})$.
```

Error vectors in REDOG's reference implementation[1], whose performance is analyzed in [KHL+22b], are generated in an equivalent way to Algorithm 5.1. Indeed, $\mathbf{e}'$ has rank weight $\mathsf{wt}_R(\mathbf{e}') = (\mathbf{e}_1 P^{-1}, -\mathbf{e}_2 S^{-1}) \leq \lambda t$ and can therefore be decoded using Φ.

Remark 5.2. Algorithm 5.1 produces an error vector $\mathbf{e}$ such that $\mathsf{wt}_H(\mathbf{e}) = \mathsf{wt}_R(\mathbf{e}) = t$ as only t coordinates of $\mathbf{e}$ are nonzero.

We are ready to give the description of an efficient message recovery algorithm.

Algorithm 5.3 *(Message recovery attack)*
Input: *REDOG's public key* pk *and a REDOG's ciphertext* $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) =$ *Encrypt*$(\mathbf{m}, \mathsf{pk})$ *generated by the reference implementation.*
Output: $\mathbf{m}$

```
1. Let $C'$ be the linear $[2n-k, \ell]$-code in the Hamming metric generated by $G = (\mathsf{pk}_1 \mid \mathsf{pk}_2)$. Put $f = 0$.
2. While $f = 0$:
   (a) Randomly select $\ell$ columns of $G$ to form the matrix $A$. Let $\mathbf{c}_A$ be the matching positions in $\mathbf{c}$.
```

[1] https://www.kpqc.or.kr/images/zip/REDOG.zip

```
    (b) If $A$ is invertible
        i. Compute $B = A^{-1}$ and $\bar{\mathbf{m}} = \mathbf{c}_A B$.
        ii. Compute $\bar{\mathbf{c}}_1 = \bar{\mathbf{m}}\mathsf{pk}_1$.
        iii. If $\mathsf{wt}_H(\mathbf{c}_1 - \bar{\mathbf{c}}_1) = t_1 \leq t$
            A. Compute $\bar{\mathbf{c}}_2 = \bar{\mathbf{m}}\mathsf{pk}_2$.
            B. If $\mathsf{wt}_H(\mathbf{c}_2 - \bar{\mathbf{c}}_2) = t - t_1$
               Put $\mathbf{m}' = \bar{\mathbf{m}}, \mathbf{e} = (\mathbf{c}_1, \mathbf{c}_2) - (\bar{\mathbf{c}}_1, \bar{\mathbf{c}}_2)$ and $f = 1$.
3. Compute $\mathbf{m} = \mathbf{m}' - \mathsf{hash}(\mathbf{e})$.
```

The inner loop is Prange's information-set decoding algorithm [Pra62] in the generator-matrix form with early aborts. If the chosen ℓ positions are not all error free then $\bar{\mathbf{m}}$ equals $\mathbf{m}$ with one or more rows of B added to it. Then $\bar{\mathbf{m}}\mathsf{pk}_1$ will be random vector and thus differ from $\mathbf{c}_1$ in more than t positions. If the initial check succeeds there is a high chance of the second condition succceeding as well leading to $\mathbf{e}$ with $\mathsf{wt}_H(\mathbf{e}) = t$.

We now analyze the success probability of each iteration of the inner loop of Algorithm 5.3. The field $\mathbb{F}_{q^m}$ is large, hence A very likely to be invertible. The algorithm succeeds if the ℓ positions forming A are chosen outside the positions where $\mathbf{e}$ has non-zero entries. This happens with probability $\binom{2n-k-t}{\ell}\binom{2n-k}{\ell}$.

Each trial costs the inversion of an $\ell \times \ell$ matrix and up to three matrix-vector products, where the vector has length ℓ and the matrices have ℓ, n, and $n-k$ columns respectively, in addition to minor costs of two vector differences and two weight computations.

We implemented the attack in Algorithm 5.3 in Sagemath 9.5; see online for the code. We perform faster early aborts, testing $\bar{\mathbf{m}}$ on only $t+3$ columns of pk_1. The probability that a coordinate matches between $\mathbf{c}_1$ and $\bar{\mathbf{c}}_1$ for $\bar{\mathbf{m}} \neq \mathbf{m}$ is q^{-m} and thus negligible for large m. Hence, most candidate vectors $\bar{\mathbf{m}}$ are discared after $(t+3)\ell^2$ multiplications in $\mathbb{F}_{q^m}$. Running the attack on a Linux Mint virtual machine we broke the KAT ciphertexts included in the submssion package for all the proposed parameters. We also generated a bunch of ciphertexts corresponding to randomly chosen public keys and messages and measured the average running time of our algorithm.

As can be seen from Table 3, the attack on the reference implementation succeeds in few steps and is very fast to execute for all parameter sets.

Security parameter	$\log_2(\mathsf{Prob})$	Time_{KAT} (sec.)	$\mathrm{Time}_{100}(sec.)$
128	-5.62325179726894	$\sim$ 8.01	$\sim$ 9.17
192	-7.51182199577027	$\sim$ 108.13	$\sim$ 112
256	-9.40052710879827	$\sim$ 167.91	$\sim$ 133.43

Table 3. Prob is the probability of success of one iteration of the inner loop of Algorithm 5.3. Time_{KAT} is the average timing of message recovery attack over entries in the KAT file (30 for 128 bits, 15 for 192 bits, 13 for 256 bits). Time_{100} is the average timing of message recovery attack over 100 ciphertext generated by REDOG's encryption.

6 Recomputing attacks costs

In this section we deal with the computation of complexities of general attacks against cryptosystems relying on the rank decoding problem. We noticed that the official REDOG submission [KHL+22a], as well as [LTP21] do not consider attack algorithms proposed in [BBC+20] and [BBB+23]

Our computations are reported in Table 4 which shows that parameters suggested for REDOG provide significantly less security than expected. The tables also confirm that the parameters do provide the claimed security under attacks prior to [BBC+20] when using a realistic exponent for matrix multiplication. Note that the computations in these tables ignore all constants and lower-order terms in the big-$\mathcal{O}$ complexities. This is in line with how the authors of the attack algorithms use their results to determine the security of other systems, but typically constants are positive and large. We apply the same to [BBB+23] although their magma code makes different choices.

Overview of rank decoding attacks Recall that the public code is generated by the $\ell \times 2n - k$ matrix $(M \mid F)$ over $\mathbb{F}_{q^m}$. The error vector added to the ciphertext is chosen to have rank t. In the description of the attacks we will give formulas for the costs using the notation of this paper, i.e., the dimension is ℓ and the error has rank t; we denote the length by N for reasons that will become clear later. The complexity of algorithms also depends on the matrix multiplication exponent ω.

The GRS [GRS16] algorithm is a combinatorial attack on the rank decoding problem. The idea behind this algorithm is to guess a vectorspace 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 to the syndrome equations. This has complexity

$$\mathcal{O}\left((N-\ell)^{\omega} m^{\omega} q^{\min\{t\lfloor \ell m/N\rfloor,(t-1)\lfloor(\ell+1)m/N\rfloor\}}\right). \tag{7}$$

Note that we use ω here while the result originally was stated with exponent 3. These matrices are not expected to be particularly sparse but should be large enough for fast matrix multiplication algorithms to apply. The same applies to the next formulas.

The second attack, introduced in [GRS16], which we denote GRS-alg, is an algebraic attack. Under the condition that $\ell > \lceil((t+1)(\ell+1)-N-1)/t\rceil$ the decoding problem can be solved in

$$\mathcal{O}\left(t^{\omega}\ell^{\omega}q^{t(\lceil((t+1)(\ell+1)-N-1)/t\rceil)}\right). \tag{8}$$

The attack AGHT [AGHT18] 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 higher chance of guessing a suitable space. It has complexity

$$\mathcal{O}\left((N-\ell)^{\omega} m^{\omega} q^{t(\ell+1)m/N-m}\right). \tag{9}$$

The BBB+ attack [BBB+20] 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 $m\binom{N-\ell-1}{t}+1 \geq \binom{N}{t}$ then the problem can be solved in

$$\mathcal{O}\left(\left(\frac{((m+N)t)^t}{t!}\right)^{\omega}\right). \tag{10}$$

If the condition is not satisfied then the complexity of solving the decoding problem becomes

$$\mathcal{O}\left(\left(\frac{((m+N)t)^{t+1}}{(t+1)!}\right)^{\omega}\right) \tag{11}$$

or the same for $t+2$ in place of $t+1$. The authors of [BBB+20] use (11) in their calculations and thus we include that as well.

The BBC+-Overdetermined,BBC+-Hybrid and BBC+-SupportMinors improvements that will follow are all introduced in [BBC+20]. They make explicit the use of extended linearization as a technique to compute Gröbner bases. For solving the rank-decoding problem it is not necessary to determine the full Gröbner basis but to find a solution to this system of equations. Extended linearization introduces new variables to turn a multivariate quadratic system into a linear system. The algorithms and complexity estimates differ in how large the resulting systems are and whether they are overdetermined or not, dependent on the system parameters.

BBC+-Overdetermined applies to the overdetermined case, which matches $m\binom{N-\ell-1}{t}+1 \geq \binom{N}{t}$, and permits to solve the system in

$$\mathcal{O}\left(m\binom{N-\ell-1}{t}\binom{N}{t}^{\omega-1}\right). \tag{12}$$

In case of an undetermined system, BBC+-Hybrid fixes some of the unknowns in a brute-force manner to produce to 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 $m\binom{N-\ell-1}{t}+1 \geq \binom{N-j}{t}$, and for each performing the same matrix computations as in BBC on j columns less. This leads to a total complexity of

$$\mathcal{O}\left(q^{jt}m\binom{N-\ell-1}{t}\binom{N-j}{t}^{\omega-1}\right). \tag{13}$$

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 in multiplying the equations by some homogeneous monomials of degree b so as 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. Let $A_b = \sum_{j=1}^{b} \binom{N}{t}\binom{m\ell+1}{j}$. The degree of the equations formed in BBC+-SupportMinors depends on b, where $0 < b < 2+t$ is minimal such that $a_b - 1 \leq \sum_{j=1}^{b} \sum_{s=1}^{j} \left((-1)^{s+1}\binom{N}{t+s}\binom{m+s-1}{s}\binom{m\ell+1}{j-s}\right)$ if such a b exists. In this case the problem can be solved with complexity

$$\mathcal{O}\left((m\ell+1)(t+1)A_b^2\right). \tag{14}$$

We do not report the last two attacks presented in [BBC+20] as the underlying approach has been pointed out to be incorrect in [BBB+23]. More precisely, [BBB+23] show that the independence assumptions made in [BBC+20] are incorrect. The SupportMinors and MaxMinors modelings in [BBC+20] are not as independent as claimed, and [BBB+23] introduces a new approach that combines them while keeping independence, at least conjecturally and matched by experiments. They again multiply by monomials of degree up to $b-1$ but a relevant difference is that the equations from the SupportMinors system are kept over $\mathbb{F}_{q^m}$. They introduce the following notation:

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

and put $\mathcal{N}_b^{\mathbb{F}_q} = \mathcal{N}_b^{\mathbb{F}_{q^m}} - \mathcal{N}_{b,syz}^{\mathbb{F}_q}$.

The problem can then be solved by linearization whenever $\mathcal{N}_b^{\mathbb{F}_q} \geq \mathcal{M}_b^{\mathbb{F}_q} - 1$. The complexity of solving the system is $T(m,N,\ell,t) = \mathcal{O}\left(\mathcal{N}_b^{\mathbb{F}_q}\left(\mathcal{M}_b^{\mathbb{F}_q}\right)^{\omega-1}\right)$.

Moreover, [BBB+23] introduce a hybrid strategy. Compared to BBC+-Hybrid it randomly picks matrices from $\mathsf{GL}_N(\mathbb{F}_q)$ to randomly 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

$$\min_{a\geq 0}\left(q^{ta}\cdot T(m,N-a,\ell-a,t)\right). \tag{15}$$

6.1 Lowering the attack costs beyond the formulas stated

The combinatorial attacks GRS and AGHT perform best for longer codes, however, algebraic attacks that turn each column into a new variable perform best with fewer variables. For each attack strategy we search for the best number of columns that we should consider in order to obtain the cheapest cost of a successful break of REDOG. This is why we presented the above formulas using N rather than the full code length $2n - k$. The conditions given above determine the minimum length required relative to dimension and rank of the error.

We then evaluate the costs for each algorithm for each choice of length $N = \ell + t + i$, for every value of $i = 0, 1, \ldots, 2n - k - \ell - t$ satisfying the conditions of the attacks. Figure 1 shows the different behaviour of the algorithms for fixed ℓ and t and increasing i. The jump in the BBB+ plot is at the transition between the two formulas.

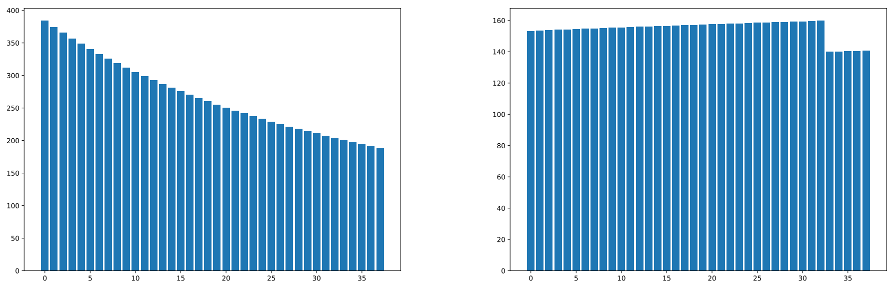

Fig. 1. Plots showing the $\log_2$ of the costs for AGHT and BBB+ for the parameters at the 128–bit security level for different choices of code length.

We point out that [BBC+20] also considered decreasing the length of the code for the case of overdetermined systems, see [BBC+20, Section 4.2] on puncturing the code in the case of "super"-overdetermined systems. We perform a systematic scan for all algorithms as an attacker will use the best possible attack.

The recomputed values We computed complexity costs for all the attacks introduced in the previous subsection, taking into consideration two values of matrix multiplication exponent, namely $\omega = 2.807$ and $\omega = 2.37$. For each possible length $N+i$ for $N = \ell+t$ and $i = 0, 1, \ldots, 2n-k-\ell-t$ we computed the costs for each attack strategy, keeping the lowest value per strategy. For the two cases of BBB+ and the three strategies described for the BBC+-* algorithms, we selected the best complexity among them. For the sake of completeness, we report the value of i in Table 4 as well and the value of a for [BBB+23]. All the values are stated as the $\log_2$ of the costs resulting from the complexity formulas. The lowest costs of the best algorithm are stated in blue. Note the above-mentioned caveats regarding evaluating big-$\mathcal{O}$ estimates for concrete parameters.

As shown in the tables, suggested parameters of REDOG for 128 and 192 levels of security do not resist BBC+ attack and Mixed-attack for any choice of ω, and BBB+ for $\omega = 2.37$. Suggested parameters for level 256 resist all

Algorithm	Formula	128 level			192 level			256 level		
		$\omega = 2.807$	$\omega = 2.37$	i	$\omega = 2.807$	$\omega = 2.37$	i	$\omega = 2.807$	$\omega = 2.37$	i
GRS [GRS16]	7	228.03	-	36	392.30	-	48	604.07	-	60
GRS-alg [GRS16]	8	207.88	-	36	368.18	-	48	595.97	-	60
AGHT [AGHT18]	9	186.68	-	37	337.69	-	49	536.22	-	61
BBB+ [BBB$^+$20]	10 & 11	140.06	118.25	33	210.26	150	0	269.03	227.15	0
BBC+ [BBC$^+$20]	12 – 14	77.83	65.73	33	175.72	159.57	48	337.92	318.01	61
Mixed [BBB$^+$23]	15	80.94	68.61	32	166.67	149.49	49	347.38	311.77	61

Table 4. Values of the $\log_2$ of attack costs for REDOG's suggested parameters for all security level (see Table 1).

attacks except BBB+ for $\omega = 2.37$. In Section 8 we propose a solution to the decryption failures that also boosts the security of REDOG.

7 Solving decryption failures

The core of REDOG's decryption failures is given by point (3) of the proof of Theorem 4.7. Indeed, the crucial step for showing decoding failure of the decoder Φ, is that $\langle \mathbf{e}_1 P^{-1} \rangle \not\subset \langle -\mathbf{e}_2 S^{-1} \rangle$.

In order to solve the issue of decryption failures in REDOG, we propose an alternative that keeps the random choice of an error vector $\mathbf{e}$ with $\mathsf{wt}_R(\mathbf{e}) = t$ and changes the public key. The idea is to retain the method introduced in [Loi17], but also to make sure that $\mathsf{wt}_R(\mathbf{e}') \leq \lambda t$. We suggest to pick $P^{-1} \in \mathsf{GL}_n(\Lambda)$ randomly instead of it being an isometry of the space $\mathbb{F}_{q^m}^n$.

The proof of the next result is an adaptation of the proof of Proposition 4.1.

Proposition 7.1. *Let $\Lambda \subset \mathbb{F}_{q^m}$ be a λ-dimensional subspace of $\mathbb{F}_{q^m}$ and $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$ a random vector with $\mathsf{wt}_R(\mathbf{e}) = t$ with $\mathbf{e}_1 \in \mathbb{F}_{q^m}^n$ and $\mathbf{e}_2 \in \mathbb{F}_{q^m}^{n-k}$. Let $S^{-1} \in \mathsf{GL}_{n-k}(\Lambda)$ and $P^{-1} \in \mathsf{GL}_n(\Lambda)$. Then $\langle \mathbf{e}_1 P^{-1}, -\mathbf{e}_2 S^{-1} \rangle \subseteq V$ for some λt-dimensional $\mathbb{F}_q$-linear vectorspace V.*

Proof. Let $\Gamma = \langle \mathbf{e} \rangle$ be the $\mathbb{F}_q$-linear subspace of $\mathbb{F}_{q^m}$ generated by $\mathbf{e}$. As before we can write $\Gamma = \langle y_1, \ldots, y_t \rangle$. Write also $\Lambda = \langle \alpha_1, \ldots, \alpha_\lambda \rangle$. As in the proof of Proposition 4.1 we can express the j-th coordinate of $\mathbf{e}_1 P^{-1}$ as a linear combination of the λt elements $y_h \alpha_k$ for $h = 1, \ldots, t$ and $k = 1, \ldots, \lambda$ as $(\mathbf{e}_1 P^{-1})_j = \sum_{h=1}^{t} \sum_{k=1}^{\lambda} c_{h,k} y_h \alpha_k$. The same can be done for each coordinate of $-\mathbf{e}_2 S^{-1}$. Hence both subspaces are contained in the space $V = \langle y_h \alpha_k \rangle$ generated by these λt elements. □

Corollary 7.2. *Let $\mathbf{e}' = \left(\mathbf{e}_1 P^{-1}, -\mathbf{e}_2 S^{-1}\right)$ with $\mathbf{e}, P^{-1}$ and S^{-1} as in Proposition 7.1. Then $\mathsf{wt}_R(\mathbf{e}') \leq \lambda t$.*

The only change to the specification of REDOG is in the KeyGen algorithm in Step 3; encryption and decryption remain unchanged as in Section 3. Here is KeyGen for the updated version of REDOG with no decryption failures.

1. Select $H = (H_1 \mid H_2)$, $H_2 \in \mathsf{GL}_{n-k}(\mathbb{F}_{q^m})$, a parity check matrix of a $[2n-k, n]$ Gabidulin code, with syndrome decoder Φ correcting r errors.
2. Select a full rank matrix $M \in \mathbb{F}_{q^m}^{\ell \times n}$.
3. 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 $P^{-1} \in \mathsf{GL}_n(\Lambda)$.
4. Compute $F = MP^{-1}H_1^T \left(H_2^T\right)^{-1} S$ and publish the public key $\mathsf{pk} = (M, F)$. Store the secret key $\mathsf{sk} = (P, H, S, \Phi)$.

Theorem 7.3. *The updated version of REDOG is correct.*

Proof. The correctness of the updated version of REDOG follows from the correctness of the original version, except for decryption correctness, which is proven by Corollary 7.2. □

8 Solving decryption failures and boosting security

Our second idea of how to deal with REDOG not decrypting correctly is to change how $\mathbf{e}$ is sampled. While the approach in Section 7 works and preserves all considerations regarding parameter sizes, in Section 6 we have shown that these are too small to offer security against the best known attacks. The approach in this section provides a functioning system and increases the security offered by the parameters.

Recall that the public key is $(M \mid F)$, where M has dimension $\ell \times n$ and F has dimension $\ell \times (n-k)$ and both, M and F, have full rank. The relative sizes in REDOG are such that $n - k = \ell - 1$, so F is just one column short of being square, and $n = \ell + t + 1$. The parameters are chosen so that the decryption step can decode errors of rank up to r, while encryption in REDOG adds only an error vector of rank t with $r \geq t\lambda$. All parameter sets have $\lambda = 3$ and $r = \lambda t = (n-k)/2$.

Encryption is computed as $\mathbf{c} = \mathbf{m}'(M \mid F) + \mathbf{e}$, for $\mathbf{m}' \in \mathbb{F}_{q^m}^{\ell}$. Decryption requires decoding in the Gabidulin code for error $(\mathbf{e}_1 P^{-1}, -\mathbf{e}_2 S^{-1})$, where P is an isometry and $S^{-1} \in \mathsf{GL}_{n-k}(\Lambda)$. We have shown in Theorem 4.7 that this $\mathbf{e}'$ typically has rank larger than r, which causes incorrect decoding, for REDOG's choice of $\mathbf{e}$ with $\mathsf{wt}_R(\mathbf{e}) = t$. Where we proposed changing the definition of P in the previous section to reach a system which has minimal changes compared to REDOG, we now suggest changing the way that $\mathbf{e}$ is chosen.

In particular, we redefine $\mathbf{e}$ to have different rank on the first n positions and the last $n-k$ positions. Let $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$ with $\mathsf{wt}_R(\mathbf{e}_1) = t_1$ and $\mathsf{wt}_R(\mathbf{e}_2) = t_2$. This can be achieved by sampling t_1 random elements from $\mathbb{F}_{q^m}$, testing that this achieves rank t_1 and taking the n positions in $\mathbf{e}_1$ as random $\mathbb{F}_q$-linear combinations of these t_1 elements. Because m is significantly larger than t_1, this finds an $\mathbf{e}_1$ of rank t_1 on first try with high probability. Similarly, we pick t_2 random elements from $\mathbb{F}_{q^m}$ and use their $\mathbb{F}_q$-linear combinations for $\mathbf{e}_2$.

We keep P being an isometry and $S^{-1} \in \mathsf{GL}_{n-k}(\Lambda)$ as in REDOG. Then the decoding step needs to find an error of rank $t_1 + \lambda t_2$, namely $\mathbf{e}_1 P^{-1}$ on the first

n positions and $\mathbf{e}_2 S^{-1}$ on the last $n-k$ positions. This will succeed if

$$r \geq t_1 + \lambda t_2. \tag{16}$$

Hence, we can consider different splits of r to maximize security.

Considerations for extreme choices of t_1 and t_2 As already explained in Section 6.1, the attacker can consider parts of $\mathbf{c}_1$ and $\mathbf{c}_2$, for example, the extreme choice of $t_1 = 0$ would mean that $\mathbf{c}_1$ is a codeword in the code generated by M and thus $\mathbf{m}'$ would be trivially recoverable from $\mathbf{c}_1 = \mathbf{m}'M$ by computing the inverse of an $\ell \times \ell$ submatrix of M. Because $\mathbb{F}_{q^m}$ is large, almost any choice of submatrix will be invertible.

The other extreme choice, $t_2 = 0$, does not cause such an obvious attack as for the REDOG parameters F has one column fewer than it has rows, meaning that $\mathbf{c}_2 = \mathbf{m}'F$ cannot be solved for $\mathbf{m}'$. Hence, at least one position of $\mathbf{c}_1$ needs to be included, but that means that we do not have a codeword in the code generated by that column of M and F but a codeword plus an error of rank 1. However, a brute-force attack on this system still succeeds with cost q^m as follows:

Let $\bar{F} = (M_i|F)$ be the square matrix obtained from taking M_i, the i-th column of M, for a choice of i that makes $\bar{F}$ invertible. Most choices of i will succeed. Let $\bar{\mathbf{c}} = (c_{1i}, \mathbf{c}_2)$, the i-th coordinate of $\mathbf{c}_1$ followed by $\mathbf{c}_2$.

For each $a \in \mathbb{F}_{q^m}$ compute $\bar{\mathbf{m}} = (\bar{\mathbf{c}} - (a, 0, 0, \ldots, 0))\bar{F}^{-1}$. Then compute $\bar{\mathbf{e}} = \mathbf{c} - \bar{\mathbf{m}}(M \mid F)$ and check if $\mathsf{wt}_R(\bar{\mathbf{e}}_1) = t_1$. If so put $\mathbf{m}' = \bar{\mathbf{m}}$ and $\mathbf{e} = \bar{\mathbf{e}}$.

The matrix operations in this attack are cheap and can be made even cheaper by observing that $\bar{\mathbf{m}} = \bar{\mathbf{c}}\bar{F}^{-1} - a\mathbf{f}$, for $\mathbf{f}$ the first *row* of $\bar{F}^{-1}$, and $\bar{\mathbf{e}} = \mathbf{c} - (\bar{\mathbf{c}}\bar{F}^{-1})(M \mid F) + a\mathbf{f}(M \mid F)$, where everything including $\mathbf{f}(M \mid F) \in \mathbb{F}_{q^m}^{2n-k}$ is fixed and can be computed once per target $\mathbf{c}$. Note also that only the $\mathbf{c}_1$ and $\mathbf{e}_1$ parts need to be computed as by construction $\mathbf{e}_2 = 0$. This leaves just n multiplications and additions in $\mathbb{F}_{q^m}$ and the rank computation for each choice of a. The search over $a \in \mathbb{F}_{q^m}$ is thus the main cost for a complexity of q^m. For all parameters of REDOG this is less than the desired security.

Generalizations of the brute-force attack For $t_1 = 1$, a brute-force attack needs to search over all $a \in \mathbb{F}_{q^m}$, up to scaling by $\mathbb{F}_q$-elements, and over all choices of error patterns, where each position of the error is a random $\mathbb{F}_q$-multiple of a. We need ℓ positions from $\mathbf{c}_1 = \mathbf{m}'M + \mathbf{e}_1$ to compute a candidate $\bar{\mathbf{m}}'$ as in the attack on $t_1 = 0$. Hence, for each $a \in \mathbb{F}_{q^m}$ we need to try at most the q^ℓ patterns for those ℓ positions of $\mathbf{e}_1$ for a cost of $(q^m - 1)q^\ell/(q-1)$. For the REDOG parameters, $q = 2$ and $m + \ell$ is significantly smaller than the security level. Hence, $t_1 = 1$ is also a bad choice.

Starting at $t_1 = 2$, when there are two elements $a, b \in \mathbb{F}_{q^m}$ and error patterns need to consider random $\mathbb{F}_q$-linear combinations of these two elements, the attack costs of $(q^m-1)(q^m-2)q^{2\ell}/(2(q-1)^2)$ grow beyond the more advanced attacks considered in Section 6.1.

Lemma 8.1. *In general, the brute-force attack on the left side takes*

$$\binom{q^m-1}{t_1} q^{t_1\ell}/(q-1)^{t_1}$$

steps.

Proof. The error vector on the left, $\mathbf{e}_1$, has rank t_1, this means that there are t_1 elements $a_1, a_2, \ldots, a_{t_1} \in \mathbb{F}_{q^m}$ which are $\mathbb{F}_q$-linearly independent. There are $\binom{q^m-1}{t_1}/(q-1)^{t_1}$ such choices up to $\mathbb{F}_q$ factors.

Each of the ℓ positions takes a random $\mathbb{F}_q$-linear combination. For a fixed choice of the a_i there are $q^{t_1\ell}$ choices for these linear combinations. Combining these quantities gives the result. □

Similarly, for $t_2 = 1$ the brute-force attack is no longer competitive, yet less clearly so than for $t_1 = 2$ because a and b appear in separate parts. There are q^m candidate choices for e_{1i} and $(q^m-1)q^{\ell-1}/(q-1)$ candidates for $\mathbf{e}_2$. For $q = 2$ this amounts to roughly $2^{2m+\ell-1}$ and $2m+\ell-1$ is larger than the security level for all parameters in REDOG.

Lemma 8.2. *In general, the brute-force attack on the right side takes*

$$q^m \binom{q^m-1}{t_2} q^{t_2(\ell-1)}/(q-1)^{t_2}$$

steps.

Proof. There are q^m choices for e_{1i}. The result follows by the same arguments as for Lemma 8.1, and taking into account that $\mathbf{e}_2$ has length $\ell - 1$. □

We do not consider other combinations of columns from the left and right as those would lead to higher ranks than these two options. Depending on the sizes of t_1 and t_2, Lemma 8.1 or 8.2 gives the better result, but apart from extreme choices these costs are very high.

Finding good choices of t_1 and t_2 We now turn to the more sophisticated attacks and try to find optimal splits of the decoding budget r between t_1 and t_2 satisfying (16), to $r \geq t_1 + \lambda t_2$. to make the best attacks as hard as possible. For any such choice, we consider attacks starting from the left with (parts of) $\mathbf{c}_1$ and M or from the right with $\mathbf{c}_2, F$, and parts of $\mathbf{c}_1$ and M. The attacks and sub-attacks differ in how many columns they require, depending on the dimension and rank, and we 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 $t_1 \leq t$ the attack may take a punctured system on $\mathbf{c}_1$ and M to recover $\mathbf{m}'$, similar to the attacks considered in Section 6, or include part of $\mathbf{c}_2$ and F, while accepting an error of larger rank including part of t_2. Hence, the search from the left may start with puncturing of $\mathbf{c}_1$. Once parts of $\mathbf{c}_2$ are included, the rank typically

parameter set	best attack	$\log_2(\text{cost})$	$N+i$	t_1	t_2	m	n	k	ℓ
128-bit	brute-force	320.00	-	12	2	83	44	8	37
192-bit	BBB+	458.25	61	15	3	109	58	10	49
256-bit	BBB+	628.20	75	21	3	135	72	12	61

Table 5. Best parameter choices and achieved security for $\omega = 2.807$, using the original values for ℓ, k, m, and n and splitting the decoding capacity r according to $r \geq t_1 + \lambda t_2$.

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 $\mathbf{c}_2$ 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 $\mathbf{c}_1$ to even have an invertible system. Hence, the attack is hardest for t_1 maximal in (16) 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 $t_1 + t_2 = r - \lambda + 1$.

A computer search, evaluating all attacks considered in Section 6 for all choices of $t_2 \in \{1, 2, \ldots, r/\lambda - 1\}$ and considering both directions as starting points for the attacker confirms that $t_2 = 1$ is optimal. See online for the Sage code used for the search. The original parameters choices for REDOG then provide the attack costs in Table 5.

This means that this second idea solves decryption failures and takes the parameters of REDOG to a safe level of strength. Actually our optimized choice of t_1 and t_2 allows enough margin to shrink the other system parameters.

Note that, as pointed out before, these computations use big-$\mathcal{O}$ complexity estimates and put all constants to 1 and lower-order terms to 0. This is in line with how estimates are presented in the papers introducing BBB+ [BBB$^+$20] and BBC+ [BBC$^+$20] but typically underestimates the security.

Remark 8.3. After we developed this idea but before posting it, the REDOG authors informed us that they fixed the decryption issue in a manner similar to the approach in this section, namely by having different ranks for $\mathbf{e}_1$ and $\mathbf{e}_2$. Their choice of $t_1 = r/2$ and $t_2 = r/(2\lambda)$ satisfies $r \geq t_1 + \lambda t_2$. but provides less security against attacks. The Sage script gives the results in Table 6 as a byproduct of computing the costs for all values of t_2.

9 Conclusions and further considerations

In this paper we showed several issues with the REDOG proposal but also some ways to repair it. One other issue is that REDOG has rather large keys for a rank-metric-based system. A strategy used by many systems in the NIST post-quantum competition, is to generate parts of the secret and public keys from seeds and storing or transmitting those seeds instead of the matrices they generated. Implementations written in C always need to define ways to take the output of a random-number generator and this strategy includes the use of a fixed such generator into the KeyGen, encryption, and decryption steps. For REDOG, this approach permits to reduce the size of the secret key sk and, at the same time, moderately shrink the size of the public key pk.

Let $f : \{0,1\}^{256} \to \{0,1\}^*$ be such a generator, where $\{0,1\}^*$ indicates that the output length is arbitrary, in a use of f the output length N must be specified. Most recent proposals use SHAKE-256 or SHAKE-512. The idea is to pick a random 256-bit seed s and initialize f with this seed, the output bits of $f(s)$ are then used in place of the regular outputs of the random-number generator to construct elements of the public or secret key. This method is beneficial if s is much smaller than the key element it replaces. The downside is that any use of that key element then incurs the costs of recomputing that element from s.

As one of the more interesting cases, we show how to build the isometry P form $f(s)$ for some seed s. Let $(n, k, \ell, q, m, \lambda)$ denote the same quantities as in REDOG.

Example 9.1. Let $N = (n^2 + m)\lceil\log_2(q)\rceil + 256$ and let $\{\alpha_1, \ldots, \alpha_m\}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Choose a random seed s and produce the N-bit string $f(s)$. Use the first $n^2\lceil\log_2(q)\rceil$ bits of $f(s)$ to determine n^2 elements in $\mathbb{F}_q$ and build an $n \times n$ matrix Q with these elements. The matrix Q is invertible with probability roughly 0.29. If this is not the case, use the last 256 bits of the output as a new seed s', discard s, and repeat the above with $f(s')$ (an average of 3 trials produces an invertible matrix).

Once an invertible Q has been constructed, use the middle $m\lceil\log_2 q\rceil$ bits of $f(s)$ to define m coefficients in $\mathbb{F}_q$ and to determine an element $\gamma \in \mathbb{F}_{q^m}$ as the $\mathbb{F}_q$-linear combination of the α_i. Then compute $P = \gamma Q$ which, by Theorem 2.4 is an isometry for the rank metric.

As a second example we show how to select S.

Example 9.2. We first observe that $\mathbb{F}_{q^m}$ is a large finite field, so any choice of λ elements for $\lambda \ll m$ will be $\mathbb{F}_q$-linearly independent with overwhelming probability. Using $N = (m + (n-k)^2)\lambda\lceil\log_2(q)\rceil$ we can determine λ random

Intended security in bits	128	192	256
Achieved security in bits ($\omega = 2.807$)	271.75	384.03	500.50
Number of columns $(N+i)$ ($\omega = 2.807$)	46	61	76
Achieved security in bits ($\omega = 2.37$)	229.45	324.24	422.58
Number of columns $(N+i)$ ($\omega = 2.37$)	46	61	76

Table 6. Results for the modified parameter for REDOG using $t_1 = r/2$ and $t_2 = r/(2\lambda)$. The stated costs are achieved by BBB+ at length $N + i$.

elements from $\mathbb{F}_{q^m}$ which define the subspace $\Lambda \subset \mathbb{F}_{q^m}$. We then define the $(n-k)^2$ entries of $S^{-1} \in \mathsf{GL}_{n-k}(\Lambda)$ as $\mathbb{F}_q$-linear combinations over those λ elements, using the next $(n-k)^2\lambda\lceil\log_2 q\rceil$ bits. The resulting matrix is almost certainly invertible and permits computing $S = (S^{-1})^{-1}$.

Similar strategies can be applied to compute the matrices M, H_1 and H_2. Let $s_P, s_S, s_M, s_{H_1}, s_{H_2}$ be the seeds corresponding to the matrices P, S, M, H_1 and H_2, respectively. Then we can set $\mathsf{sk} = (s_P, s_S, s_{H_1}, s_{H_2})$ and $\mathsf{pk} = (s_M, F)$ where $F = MP^{-1}H_1^T\left(H_2^T\right)^{-1}S$. This approach cannot be used to compress F as it depends on the other matrices. In this way we reduced the private key size of REDOG to 1024 bits and public key of size of REDOG to $256+\ell(n-k)m\lceil\log_2(q)\rceil$. For the 128-bit-security level, we obtain a secret key size of 0.13 KB compared to the original 1.45 KB and a public key size of $13,85$ KB, compared to the original $14,25$KB (which was obtained by choosing M to be a circulant matrix) at the expense of having to recompute the matrices from their seeds when needed. Given that matrix inversion over $\mathbb{F}_{q^m}$ is not fast, implementations may prefer to include S and S^{-1} in sk and use seeds for the other matrices. To save even more space, it is possible to replace $s_P, s_S, s_M, s_{H_1}, s_{H_2}$ by a single seed s and generating those five seeds as a call to $f(s)$. The public key then includes the derived value s_M but the secret key consists only of s. Note that in that case each non-invertible Q will be generated for each run expanding the secret seed, before finding the Q and P that were used in computing pk. In summary, this strategy provides a tradeoff between size and computing time.

References

AAB+17a. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loic Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Phillipe Gaborit, Adrien Hauteville, and Gilles Zémor. Ouroboros-R. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions.

AAB+17b. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loic Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Phillipe Gaborit, and Gilles Zémor. RQC. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions.

AASA+20. Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone. Status report on the second round of the NIST post-quantum cryptography standardization process. NIST IR 8309, 2020. https://doi.org/10.6028/NIST.IR.8309.

ABD+17a. Nicolas Aragon, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Adrien Hauteville, Olivier Ruatta, Jean-Pierre Tillich, and Gilles Zémor. LAKE. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions.

ABD$^+$17b. Nicolas Aragon, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Adrien Hauteville, Olivier Ruatta, Jean-Pierre Tillich, and Gilles Zémor. LOCKER. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions.

AGHT18. Nicolas Aragon, Philippe Gaborit, Adrien Hauteville, and Jean-Pierre Tillich. A new algorithm for solving the rank syndrome decoding problem. In *ISIT*, pages 2421–2425. IEEE, 2018.

BBB$^+$20. Magali Bardet, Pierre Briaud, Maxime Bros, Philippe Gaborit, Vincent Neiger, Olivier Ruatta, and Jean-Pierre 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, 2020. Springer, Heidelberg, Germany.

BBB$^+$23. Magali Bardet, Pierre Briaud, Maxime Bros, Philippe Gaborit, and Jean-Pierre Tillich. Revisiting algebraic attacks on minrank and on the rank decoding problem. *Designs, Codes and Cryptography*, pages 1–37, 07 2023.

BBC$^+$20. Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray A. Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier A. Verbel. Improvements of algebraic attacks for solving the rank decoding and Min-Rank 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, 2020. Springer, Heidelberg, Germany.

Ber03. Thierry P. Berger. Isometries for rank distance and permutation group of Gabidulin codes. *IEEE Trans. Inf. Theory*, 49(11):3016–3019, 2003.

Del78. Philippe Delsarte. Bilinear forms over a finite field, with applications to coding theory. *Journal of Combinatorial Theory, Series A*, 25(3):226–241, 1978.

DGZ17. Jean-Christophe Deneuville, Philippe Gaborit, and Gilles Zémor. Ouroboros: A simple, secure and efficient key exchange protocol based on coding theory. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017*, pages 18–34, Utrecht, The Netherlands, June 26–28, 2017. Springer, Heidelberg, Germany.

Gab85. Ernst M. Gabidulin. Theory of codes with maximum rank distance. *Problems of Information Transmission*, 21(1):1–12, 1985.

GKK$^+$17. Lucky Galvez, Jon-Lark Kim, Myeong Jae Kim, Young-Sik Kim, and Nari Lee. McNie. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions.

GPT91. Ernst M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. Ideals over a non-commutative ring and thier applications in cryptology. In Donald W. Davies, editor, *EUROCRYPT'91*, volume 547 of *LNCS*, pages 482–489, Brighton, UK, April 8–11, 1991. Springer, Heidelberg, Germany.

GRS16. Philippe Gaborit, Olivier Ruatta, and Julien Schrek. On the complexity of the rank syndrome decoding problem. *IEEE Transactions on Information Theory*, 62(2):1006–1019, 2016.

KHL$^+$22a. Jon-Lark Kim, Jihoon Hong, Terry Shue Chien Lau, YounJae Lim, Chik How Tan, Theo Fanuela Prabowo, and Byung-Sun Won. REDOG. Submission to KpqC Round 1, 2022.

KHL$^+$22b. Jon-Lark Kim, Jihoon Hong, Terry Shue Chien Lau, YounJae Lim, Chik How Tan, Theo Fanuela Prabowo, and Byung-Sun Won. REDOG and its performance analysis. Cryptology ePrint Archive, Report 2022/1663, 2022. https://eprint.iacr.org/2022/1663.

KKGK21. Jon-Lark Kim, Young-Sik Kim, Lucky Erap Galvez, and Myeong Jae Kim. A modified Dual-Ouroboros public-key encryption using Gabidulin codes. *Appl. Algebra Eng. Commun. Comput.*, 32(2):147–156, 2021.

Loi17. Pierre Loidreau. A new rank metric codes based encryption scheme. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017*, pages 3–17, Utrecht, The Netherlands, June 26–28, 2017. Springer, Heidelberg, Germany.

LPR23. Tanja Lange, Alex Pellegrini, and Alberto Ravagnani. On the security of REDOG. Cryptology ePrint Archive, Paper 2023/1205, 2023. https://eprint.iacr.org/2023/1205.

LT18. Terry Shue Chien Lau and Chik How Tan. Key recovery attack on McNie based on low rank parity check codes and its reparation. In Atsuo Inomata and Kan Yasuda, editors, *IWSEC 18*, volume 11049 of *LNCS*, pages 19–34, Sendai, Japan, September 3–5, 2018. Springer, Heidelberg, Germany.

LTP21. Terry Shue Chien Lau, Chik How Tan, and Theo Fanuela Prabowo. On the security of the modified Dual-Ouroboros PKE using Gabidulin codes. *Appl. Algebra Eng. Commun. Comput.*, 32(6):681–699, 2021.

Ove05. Raphael Overbeck. A new structural attack for GPT and variants. In *Mycrypt*, volume 3715 of *Lecture Notes in Computer Science*, pages 50–63. Springer, 2005.

Ove08. R. Overbeck. Structural attacks for public key cryptosystems based on Gabidulin codes. *Journal of Cryptology*, 21(2):280–301, April 2008.

Pra62. E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

S$^+$21. W. A. Stein et al. *Sage Mathematics Software (Version (9.3))*. The Sage Development Team, 2021. http://www.sagemath.org.

Sta11. Richard P. Stanley. *Enumerative Combinatorics*, volume 1 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 2 edition, 2011.