The Blockwise Rank Syndrome Learning problem and its applications to cryptography

Nicolas Aragon[1], Pierre Briaud[2,3], Victor Dyseryn[1], Philippe Gaborit[1], and Adrien Vinçotte[1]

[1] XLIM, Université de Limoges, France
[2] Inria Paris, France
[3] Sorbonne Université, France

Abstract. Recently the notion of blockwise error in a context of rank based cryptography has been introduced in [30]. This notion of error, very close to the notion sum-rank metric [26], permits, by decreasing the weight of the decoded error, to greatly improve parameters for the LRPC and RQC cryptographic schemes. A little before the multi-syndromes approach introduced for LRPC and RQC schemes in [3, 17] had also allowed to considerably decrease parameters sizes for LRPC and RQC schemes, through in particular the introduction of Augmented Gabidulin codes.

In the present paper we show that the two previous approaches (blockwise errors and multi-syndromes) can be combined in a unique approach which leads to very efficient generalized RQC and LRPC schemes. In order to do so, we introduce a new problem, the Blockwise Rank Support Learning problem, which consists of guessing the support of the errors when several syndromes are given in input, with blockwise structured errors. The new schemes we introduce have very interesting features since for 128 bits security they permit to obtain generalized schemes for which the sum of public key and ciphertext is only 1.4 kB for the generalized RQC scheme and 1.7 kB for the generalized LRPC scheme. The new approach proposed in this paper permits to reach a 40% gain in terms of parameters size when compared to previous results [17, 30], obtaining even better results in terms of size than for the KYBER scheme whose total sum is 1.5 kB.

Besides the description of theses new schemes the paper provides new attacks for the l-RD problem introduced in [30], in particular these new attacks permit to cryptanalyze all blockwise LRPC parameters proposed in [30] (with an improvement of more than 40bits in the case of structural attacks). We also describe combinatorial attacks and algebraic attacks, in the spirit of the recent paper [17], for the new Blockwise Rank Support Learning problem we introduce.

1 Introduction and previous works

Background on rank metric code-based cryptography. Classical code-based cryptography relies on the Hamming distance but it is also possible to use another metric: the rank metric. This metric introduced in 1985 by Gabidulin [18] is very different from the Hamming distance. In recent years, the rank metric has received very strong attention from the coding community because of its relevance to network coding. Moreover, this metric can also be used for cryptography. Indeed it is possible to construct rank-analogues of Reed-Solomon codes: the Gabidulin codes. Gabidulin codes inspired early cryptosystems, like the GPT cryposystem ([19]), but they turned out to be inherently vulnerable because of the very strong structure of the underlying codes. More recently, by considering an approach similar to NTRU [23](and also MDPC codes [25]), constructing a very efficient cryptosystem based on weakly structured rank codes was shown to be possible, the LRPC cryptosystem [21]. Overall the main interest of rank-metric based cryptography is that the complexity of the best known attack grows very fast with size of parameters: unlike (Hamming) code-based or lattice-based cryptography, it is possible to obtain a cryptosystem based on *a general instance of the rank decoding problem* with size only a few thousands bytes, when such sizes of parameters can only be obtained with additional structure (quasi-cyclic for instance) for code-based or lattice based cryptography. At the 2017 NIST standardization process several schemes based on rank metric were proposed (LAKE, LOCKER, OUROBOROS-R and RQC), for the second round the three schemes LAKE, LOCKER and OUROBOROS-R were merged in the ROLLO 2nd round submission and the RQC submission remained as an independent submission. Eventually due to incertitudes brought by algebraic attacks [13] which attacked NIST proposed parameters for rank metric, the schemes did not reach the Third round of the NIST standardization, meanwhile the overall process permitted to give a new audience for the potentiality of rank-based cryptosystems. The Loidreau cryptosystem [24] and its recent improvement [8] are another example of rank-based cryptosystem. In this paper we focus on the LRPC and RQC cryptosystems.

Historical evolution of the LRPC cryptosystem. The main point which permits to obtain small size parameters for the LRPC cryptosystem is their decoding algorithm. In the original 2014 version of the cryptosystem [21], the Decoding Failure Rate (DFR) is related to the block size n of the code, which is a major drawback if one intents to reach a very low DFR as expected to obtain IND-CCA2 security. The adopted approach for LRPC was either to consider a cryptosystem with high DFR (of order 2^{-30}, as in the LAKE cryptosystem), or considering a very low DFR but at a cost of a high block size n, which leads to very high parameters (as in the LOCKER cryptosystem). Overall, even if the LAKE parameters were very appealing (public key $\simeq$ 600 bytes) the high DFR remained a strong limitation, and on the contrary obtaining a very low DFR implied very high parameters (4 kB) for LOCKER which made the scheme less competitive than its high DFR counterpart. Another possibility to decrease the DFR was proposed in [9] but relies on having a bigger m (the dimension of the extension field), which overall is too expensive. If one excepts the introduction of Ideal LRPC for the second round of NIST standardization process for ROLLO, which permits to increase the number of choices for the block size of LRPC, there was not any major breakthrough for LRPC until the introduction in 2022 [3] of the multiple syndromes approach: this approach, based on the Rank Support Learning problem, permits to consider several syndromes. It has a strong impact on parameters since it permits to increase the number of considered syndromes and hence the overall decoding capacity of the code. This approach did not really change the high DFR approach, but had a strong impact on the very low DFR approach which reached a size (pk+ct) of 2.4 kB, a strong improvement compared to the previous 4 kB. In practice the multiple syndrome approach permits to consider a decoding capacity potentially close to the rank Gilbert-Varshamov which has a double impact on parameters: first the attacks become more expensive in complexity, and approaching the RGV bound is a parameter area for which algebraic attack are less efficient and have a complexity similar to combinatorial attacks. The previously cited paper [3] also allows to build unstructured LRPC variations of the scheme with very low parameters of 7 kB, which beats best unstructured lattices schemes. At last the paper also introduced the extended multiple syndromes (xMS) approach in which at a cost of a slower decoding algorithm it is possible to decode LRPC codes with smaller m, the key point to obtain smaller parameters. Very recently another approach was proposed in [30]. This approach uses blockwise errors to increase the decoding capacity of the LRPC codes: it permits to reach smaller parameters but not as small as the multiple syndrome approach, mainly because the classical LRPC approach relies on large block size to reach very low DFR.

Historical evolution of the RQC cryptosystem. The RQC cryptosystem was submitted to the 2017 NIST standardization process and in [1], prepublished in 2016. It is also in the scope of the 2010 Gaborit-Aguilar patent [4]. The scheme is an equivalent in rank metric of the HQC scheme submitted to the NIST standardization process. The security of the protocol can be reduced to the security of random instances, but it comes at a cost of two parts ciphertexts, which naturally implies a bigger parameters size. The main strong feature of the RQC protocol is the fact that thanks to the Gabidulin decoder, it is possible to obtain a zero DFR, avoiding potential DFR existential drawbacks. In practice RQC parameters were rather large and after algebraic attacks of 2019 [13] reached 5.6 kB (for 128b security) for public key + ciphertext size. There are two reasons for this. First the fact that the weight of the decoded error increases quadratically, which induces a bigger block size n and hence a bigger m which for Gabidulin codes has to be greater than n. Second the fact that for the RQC scheme the security of the code is reduced to attacking a $[3n, n]$ code rather than a $[2n, n]$ code (as for LRPC), which significantly impacts the complexity of attacks. Overall, although in itself the zero DFR is an attractive feature, the parameters size is less appealing. After the RQC NIST submission several improvements were proposed. First in 2019 a notion of non-homogeneous error was proposed for the second round submission of RQC, this approach with a common error support for the first 2n coordinates and a different support the last n length block, was a way to counter the costly $[3n, n]$ reduction. At last, recently in [14] the notion of multiple syndromes was also extended to the RQC cryptosystem, this approach as for LRPC is very interesting in itself but is even more efficient with the Augmented Gabidulin codes also introduced in [14]. The Augmented Gabidulin codes correspond to Gabidulin codes with additional zero positions, in practice it permits to mitigate the condition $n \leq m$, it comes at a cost of obtaining a non zero DFR but with quadratic negative exponent which makes the approach very efficient, since it practice it permits to decrease m but with a similar decoding capacity and for a very low DFR. This approach combined with the multiple syndromes approach and the non-homogeneous errors, permits to reach a 2.7 kB parameters size. It also permits to reach low parameters for the unstructured case (see [14] for details).

Recent results and introduction of blockwise rank errors for rank codes for LRPC and RQC schemes Very recently in [30], the authors introduced the notion of rank blockwise errors, which permits to decrease the weight of decoded errors. The main idea of this approach is to consider words formed of blocks of respective length $n_1, ..., n_l$ with each block being associated to a given error e_i of rank r_i with support E_i such that the supports E_i do not intersect. In the case of $l = 2$ it permits to get an error to decode for LRPC of smaller weight $r_1.d_1 + r_2.d_2$ rather than $r.d$ in the case of classical LRPC. In fact to give a general idea one exchanges the complexity of searching for an error of weight $2r$ and length $2n$ by the complexity of searching for a blockwise error of weight (r, r) associated to two blocks of length n. If one considers $r = d$ and $r_1 = r_2 = d_1 = d_2 = \frac{r}{2}$, the classical LRPC approach with homogeneous errors gives an error of weight $r.d = r^2$ when for the blockwise case the error has weight $r_1.d_1 + r_2.d_2 = \frac{r^2}{2}$. Having to decode of smaller weight can have a strong impact for decoding. In their paper [30] the authors then generalize previously known attacks in their blockwise rank error case (both for combinatorial and algebraic attacks) following recent results on non-homogeneous errors. In practice they show that in certain cases there can be an advantage in considering the blockwise approach rather than the classical homogeneous approach. The approach is especially interesting for the RQC scheme for which they propose parameters with size 2.5 kB (public key + ciphertext), and a little less for the ILRPC case where with high DFR 2^{-30} where their parameters are 15% smaller than ROLLO-I (ex-LAKE) (but in fact as we will explain later their proposed parameters can be broken). Overall the approach they propose is very interesting and completely develop the potential of rank metric.

Blockwise rank errors: why this new structured error structure is completely suited for rank metric based cryptography The notion of rank error is a well known notion, historically this metric benefits from strange properties. Indeed suppose one wants to solve the RSD problem $H.e^t = s$ (for e a codeword of $\mathbb{F}_{q^m}^n$ of weight r and H a random $(n-k) \times n$ matrix), the error e can be seen as an $m \times n$ matrix and in practice the complexity of best attacks becomes linear whenever n becomes large enough. This property is directly related to the notion of support of the error: when the error has a larger length the support of the error does not change, this peculiar property leads to the fact that it is easily possible to construct simple codes which can decode up to the rank Gilbert-Varshamov bound [20]. Notice that this type of feature is not present for Hamming or Euclidean distance for instance. This property also explains why a straightforward adaptation of the Learning Parity with Noise (LPN) or Learning With Errors (LWE) problem does not work for rank metric, since at some point after a quadratic number of given syndromes it is possible to polynomially solve the system. A way to obtain an equivalent approach for LPN or LWE in rank metric is proposed in [16] in which rather than adding errors with always the same support one adds fixed length block errors with different error supports. This Learning with Rank Errors (LRE) approach permits to get an equivalent notion to LPN and LWE. The previous LRE approach is very close to the approach proposed in [30] and is also closely related to the sum-rank approach. Also the non homogeneous approach of [14] can be seen as a particular case of blockwise rank errors. In practice the rank blockwise error approach permits to efficiently counter the attack in which for a given m one dramatically increases the length n of the code. Concretely the best combinatorial attacks have a complexity with roughly an exponent in krm/n, the blockwise structured error support counters the m/n effect so that the best attacks essentially remains in kr for the exponent and typically this type of structured error is especially resistant for $[\ell n, n]$ codes with blocks of size n and $m = n$. This type of parameters corresponds very well to ideal LRPC and RQC schemes for which the main attacks corresponds precisely to this case. However the case of unstructured scheme when m is larger than n (and n is small) does not permit to benefit from the advantage of this blockwise structure and hence does not seem to reach any improvement. Moreover the blockwise structure, as explained in [30] permits to decrease the weight of error to decode for LRPC and RQC. This point of view leans in the direction that the blockwise rank error approach is the natural approach to consider for rank metric since first it naturally permits to get smaller error weight to decode and since, second, it is naturally resilient to the very long length attack approach which necessary leads to polynomial attacks. In particular this approach is especially efficient for RQC since it permits to counter the $[3n, n]$ attack which strongly impacts parameters. This explains why RQC parameters of [30] are rather small. In practice this block size approach is especially interesting for the case where the main attack arises for $m << n$, which is precisely the case of ideal LRPC and RQC.

Contributions In this paper we combine the two previous approaches : multiple syndromes (together with Augmented Gabidulin codes) and blockwise errors for LRPC and RQC schemes. This new combined approach is especially efficient for the RQC scheme for which it permits to obtain parameters of size 1.4 kB (public key + ciphertext) for 128 bit security, since the blockwise approach counters the $[3n, n]$ security reduction. However the approach in the case of LRPC codes combined with the xMS approach of [3] also remains interesting with a 1.7 kB size. These results are really a big step compared to previous results with a 40% decrease in terms of parameters size, giving parameters even smaller than KYBER parameters (1.5 kB), it is the first time that one gets so small parameters in rank metric (and codes in general) along with very small DFR.

Besides these main results the contributions are the following:

- We define a new problem : the Blockwise Rank Syndrome Learning problem which permits to design new generalized LRPC and RQC schemes using multiple syndromes and blockwise rank error approaches. We generalize the xMS approach of [3] for the case of rank block errors.

- We give new attacks for the ℓ-RD blockwise error problem, in particular we break all parameters of [30] for their LRPC variations. Notice that it does not alter the confidence we can have in the scheme, since parameters can be increased to thwart this attack.

- We give generalized combinatorial and algebraic attacks for the new Blockwise Rank Syndrome Learning problem.

- We revisit some combinatorial and algebraic attacks described in [30].

Organisation of the paper Section 1 gives a general overview of the situation for LRPC and RQC schemes and also gives a perspective on the blockwise rank error approach. Section 2 gives the general background on rank metric and cryptographic schemes. Section 3 describes the new blockwise RSL problem together with the generalization of the XMS approach in the case of blockwise rank errors. Section 4 gives a description of our new generalized RQC and LRPC schemes. Section 5 and 6 gives details for combinatorial and algebraic attacks for the problem we consider , but also revisit some complexities of [30]. Section 7 describes the cryptanalyze of LRPC parameters of [30]. At lest Section 8 describes new parameters with our new approach and compares to other schemes.

2 Preliminaries and background

2.1 Background on rank metric

Definition 1 (Rank metric over $\mathbb{F}_{q^m}^n$). *Let $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{F}_{q^m}^n$ and let $(b_1, \dots, b_m) \in \mathbb{F}_{q^m}^m$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Each coordinate x_j is associated to a vector of $\mathbb{F}_q^m$ in this basis: $x_j = \sum_{i=1}^m m_{ij} b_i$. The $m \times n$ matrix associated to $\boldsymbol{x}$ is given by $\boldsymbol{M}(\boldsymbol{x}) = (m_{ij})_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant n}}$.*

The rank weight $\|\boldsymbol{x}\|$ of $\boldsymbol{x}$ is defined as the rank of $\boldsymbol{M}(\boldsymbol{x})$. This definition does not depend on the choice of the basis. The associated distance $d(\boldsymbol{x}, \boldsymbol{y})$ between elements $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{F}_{q^m}^n$ is defined by $d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|$.

The support of $\boldsymbol{x}$, denoted $\mathsf{Supp}(\boldsymbol{x})$, is the $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$ generated by the coordinates of $\boldsymbol{x}$: $\mathsf{Supp}(\boldsymbol{x}) \overset{def}{=} \langle x_1, \dots, x_n \rangle$. We have $\dim \mathsf{Supp}(\boldsymbol{x}) = \|\boldsymbol{x}\|$.

Definition 2 **($\mathbb{F}_{q^m}$-linear code).** *An $\mathbb{F}_{q^m}$-linear code $\mathcal{C}$ of dimension k and length n is a subspace of dimension k of $\mathbb{F}_{q^m}^n$ seen as a rank metric space. The notation $[n,k]_{q^m}$ is used to denote its parameters.*

The code $\mathcal{C}$ can be represented by two equivalent ways:

- *by a generator matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k\times n}$. Each row of $\boldsymbol{G}$ is an element of a basis of $\mathcal{C}$,*

$$\mathcal{C} = \{\boldsymbol{x}\boldsymbol{G}, \boldsymbol{x} \in \mathbb{F}_{q^m}^k\}.$$

- *by a parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$. Each row of $\boldsymbol{H}$ determines a parity-check equation verified by the elements of $\mathcal{C}$:*

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{F}_{q^m}^n : \boldsymbol{H}\boldsymbol{x}^t = \boldsymbol{0}\}.$$

We say that $\boldsymbol{G}$ (respectively $\boldsymbol{H}$) is under systematic form if and only if it is of the form $(\boldsymbol{I}_k|\boldsymbol{A})$ (respectively $(\boldsymbol{I}_{n-k}|\boldsymbol{B})$).

2.2 Difficult problems in rank metric

We begin by recall the two main equivalent problems of decoding in rank metric:

Definition 3 **(RD Problem).** *On input $(\boldsymbol{G}, \boldsymbol{y}) \in \mathbb{F}_{q^m}^{k\times n} \times \mathbb{F}_{q^m}^n$, the Rank Syndrome Decoding problem $\mathsf{RD}(n,k,r)$ asks to compute $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ such that $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{G} + \boldsymbol{e}$ and $\|e\| \leq r$.*

Definition 4 **(RSD Problem).** *On input $(\boldsymbol{H}, \boldsymbol{s}) \in \mathbb{F}_{q^m}^{(n-k)\times n} \times \mathbb{F}_{q^m}^{n-k}$, the Rank Syndrome Decoding problem $\mathsf{RSD}(n,k,r)$ asks to compute $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ such that $\boldsymbol{H}\boldsymbol{e}^t = \boldsymbol{s}^t$ and $\|e\| \leq r$.*

It is proven in [28] that the Syndrome Decoding problem in the Hamming metric, which is a well-known NP-hard problem, is probabilistically reduced to the RSD problem. The following problem was introduced in [20] and generalizes the Rank Syndrome Decoding problem: instead of having only one syndrome, several syndromes which shares the same support are given as input.

Definition 5 **(RSL Problem).** *On input $(\boldsymbol{H}, \boldsymbol{S}) \in \mathbb{F}_{q^m}^{(n-k)\times n} \times \mathbb{F}_{q^m}^{\ell\times(n-k)}$, the Rank Support Learning Problem $\mathsf{RSL}(n,k,r,\ell)$ asks to compute a subspace E of $\mathbb{F}_{q^m}$ of dimension r, such that there exists a matrix $\boldsymbol{V} \in E^{\ell\times n}$ such that $\boldsymbol{H}\boldsymbol{V}^t = \boldsymbol{S}^t$*

The security of the RSL problem is similar to the RSD problem for a small number of syndromes. A detailed analysis of the difficulty of solving this problem can be found in [3].

2.3 Ideal codes

Let $P \in \mathbb{F}_q[X]$ an irreducible polynomial of degree n. We define the intern product of two vectors $\mathbf{x}, \mathbf{y}$ in $\mathbb{F}_{q^m}^n$ as:

$$\mathbf{x}\mathbf{y} \overset{\text{def}}{=} \mathbf{X}(X)\mathbf{Y}(X) \mod P$$

where $\mathbf{X}(X) = \sum_{i=0}^{k-1} x_i X^i$ and $\mathbf{Y}(X) = \sum_{i=0}^{k-1} y_i X^i$. It can be seen as a matrix-vector product by the so-called ideal matrix generated by $\mathbf{x}$ and P.

Definition 6 **(Ideal matrix).** *Let $P \in \mathbb{F}_q[X]$ a polynomial of degree n and $\boldsymbol{v} \in \mathbb{F}_{q^m}^n$. The ideal matrix generated by $\boldsymbol{v}$ and P, noted $\mathcal{IM}_P(\boldsymbol{v})$ (or $\mathcal{IM}(\boldsymbol{v})$ if there is no ambiguity on the value of P), is an $n \times n$ whose coefficients belong to $\mathbb{F}_{q^m}$, such that:*

$$\mathcal{IM}_P(\boldsymbol{v}) = \begin{pmatrix} \boldsymbol{v}(X) \mod P \\ X\boldsymbol{v}(X) \mod P \\ \vdots \\ X^{k-1}\boldsymbol{v}(X) \mod P \end{pmatrix}.$$

One can see that $\mathbf{u} \cdot \mathbf{v} = \mathbf{u}\mathcal{IM}(\mathbf{v}) = \mathbf{v}\mathcal{IM}(\mathbf{u}) = \mathbf{v} \cdot \mathbf{u}$. An ideal code $\mathcal{C}$ of parameters $[sn, tn]_{q^m}$ is an $\mathbb{F}_{q^m}$-linear code which admits a generator matrix made of $s \times t$ ideal matrix blocks. A crucial point is that if $P \in \mathbb{F}_q[X]$ is irreducible and n and m are prime, then $\mathcal{C}$ admits a systematic generator matrix made of ideal block [1]. Hereafter, we only consider the case $t = 1$.

Definition 7 (Ideal codes). *Let $P(X) \in \mathbb{F}_q[X]$ be a polynomial of degree n, and $\boldsymbol{g}_i \in \mathbb{F}_{q^m}^n$ for $i \in \{1, ..., s-1\}$. We call the $[sn, n]_{q^m}$ ideal code $\mathcal{C}$ of generators $(\boldsymbol{g}_1, ..., \boldsymbol{g}_{s-1})$ the code with generator matrix $\boldsymbol{G} = \big(\boldsymbol{I}_n \; \mathcal{IM}(\boldsymbol{g}_1) \; ... \; \mathcal{IM}(\boldsymbol{g}_{s-1})\big) \in \mathbb{F}_{q^m}^{n \times sn}$. Equivalently, $\mathcal{C}$ admits a parity check matrix of the form*

$$\boldsymbol{H} = \begin{pmatrix} & \mathcal{IM}(\boldsymbol{h}_1) \\ \boldsymbol{I}_{n(s-1)} & \vdots \\ & \mathcal{IM}(\boldsymbol{h}_{s-1}) \end{pmatrix}.$$

We can therefore define the RSD and RSL problem adapted to ideal codes.

Definition 8 (IRSD Problem). *Let $\boldsymbol{H}$ the parity check matrix of a $[sn, n]$ ideal code. On input $(\boldsymbol{H}, \boldsymbol{s}) \in \mathbb{F}_{q^m}^{(s-1)n \times sn} \times \mathbb{F}_{q^m}^{(s-1)n}$, the Ideal Rank Support Decoding Problem $\mathsf{IRSD}(n, s, r)$ asks to compute a subspace E of $\mathbb{F}_{q^m}$ of dimension r, such that there exists a matrix $\boldsymbol{e} \in E^n$ such that $\boldsymbol{H}\boldsymbol{e}^t = \boldsymbol{s}^t$*

Definition 9 (IRSL Problem). *Let $\boldsymbol{H}$ the parity check matrix of a $[sn, n]$ ideal code. On input $(\boldsymbol{H}, \boldsymbol{S}) \in \mathbb{F}_{q^m}^{(s-1)n \times sn} \times \mathbb{F}_{q^m}^{N \times (s-1)n}$, the Ideal Rank Support Learning Problem $\mathsf{IRSL}(n, s, r, N)$ asks to compute a subspace E of $\mathbb{F}_{q^m}$ of dimension r, such that there exists a matrix $\boldsymbol{V} \in E^{N \times n}$ such that $\boldsymbol{H}\boldsymbol{V}^t = \boldsymbol{S}^t$*

2.4 LRPC codes

LRPC codes, introduced by [21], are well suited codes for cryptography thanks to their strong decoding capacity and their weak algebraic structure.

Definition 10 (LRPC code). *Let $\boldsymbol{H} = (h_{i,j})_{(i,j) \in \{1,...,n-k\} \times \{1,...,n\}} \in \mathbb{F}_{q^m}^{(n-k) \times n}$ a full-rank matrix, whose coordinates generate an $\mathbb{F}_q$-vectorial space $F = \langle h_{i,j} \rangle$ of small dimension d. Let $\mathcal{C}$ the code with parity check matrix $\boldsymbol{H}$. By definition, $\mathcal{C}$ is an $[n, k]_{q^m}$ LRPC code of dual weight d. Such a matrix $\boldsymbol{H}$ is called a homogeneous matrix of weight d and support F.*

We clearly see that it is possible to combine the two ideas above to obtain codes whose parity check matrix has the two properties on the same time, by imposing to the set of polynomials $(\mathbf{h}_i)_i$ to have its coefficents belonging to a vectorial space of small dimension. This is the type of code that we use later in our new schemes.

Definition 11 (Ideal-LRPC code). *An Ideal-LRPC code is both an Ideal code and an LRPC code.*

Notice that the IRSL problem isn't easier if $\mathbf{H}$ is a low-rank parity check matrix than in the general case of a random Ideal Parity check matrix. Let F the vectorial space generated by coefficient of an low-rank parity check matrix. Then there exists an efficient algorithm Rank Support Recover (RSR) that takes in input F, the syndrome associated to the error and its rank, and recovers the support of the error [21]. Let $P \in \mathbb{F}_q[X]$ an irreducible polynomial of degree n. $\oplus$ denotes here the bitwise XOR. The LOCKER Public Key Encryption scheme, presented in Figure 1, was introduced in [7]. Its security relies on the difficulty to solve the IRSD problem, whose the parity check matrix in instance is $\big(\mathbf{1} \; \mathbf{h}\big)$.

A Key Encapsulation Mechanism KEM = ($\mathsf{KeyGen}, \mathsf{Encap}, \mathsf{Decap}$) is a triple of probabilistic algorithms together with a key space $\mathcal{K}$. The key generation algorithm KeyGen generates a pair of public and secret keys $(\mathsf{pk}, \mathsf{sk})$. The encapsulation algorithm Encap uses the public key pk to produce an encapsulation c and a key $K \in \mathcal{K}$. Finally Decap, using the secret key sk and an encapsulation c, recovers the key $K \in \mathcal{K}$, or fails and returns $\perp$.

```
KeyGen($1^\lambda$):
    - Sample uniformly $(\mathbf{x},\mathbf{y}) \xleftarrow{\$} \mathcal{S}_d^{2n}(\mathbb{F}_{q^m})$.
    - Compute $\mathbf{h} = \mathbf{x} \cdot \mathbf{y}^{-1} \mod P$
    - Output $\mathsf{pk} = \mathbf{h}$ and $\mathsf{sk} = (\mathbf{x},\mathbf{y})$

Encrypt($\mathsf{pk}, \mathbf{m}$):
    - Sample uniformly at random $(\mathbf{e}_1,\mathbf{e}_2) \xleftarrow{\$} \mathcal{S}_r^{2n}(\mathbb{F}_{q^m})$
    - Compute $E = \mathsf{Supp}(\mathbf{e}_1,\mathbf{e}_2)$ and $cipher = \mathbf{m} \oplus \mathsf{H}(E)$
    - Compute $\mathbf{c} = \mathbf{e}_1 + \mathbf{e}_2 \cdot \mathbf{h}$ and output $\mathsf{ct} = (cipher, \mathbf{c})$.

Decrypt($\mathsf{sk}, \mathsf{ct}$):
    - Compute $\mathbf{s} = \mathbf{xc}$, set $F = \mathsf{Supp}(\mathbf{x},\mathbf{y})$ and retrieve $E = \mathsf{RSR}(F,\mathbf{s},r)$
    - Output $\mathbf{m} = cipher \oplus \mathsf{H}(E)$
```

Fig. 1: Description of the LOCKER scheme

The encapsulation scheme also needs a hash function H, modeled as a random oracle. The KEM scheme shown in Figure 2 has been introduced in [3], and exploits several syndromes given for decoding. Its security relies on RSL problem.

```
KeyGen($1^\lambda$):
    - Choose uniformly at random a subspace $F$ of $\mathbb{F}_{q^m}$ of dimension $d$.
    - Sample $\mathbf{U} = (\mathbf{A}|\mathbf{B}) \xleftarrow{\$} F^{(n-k)\times n}$.
    - Output $\mathbf{H} = (\mathbf{I}_{n-k}|\mathbf{A}^{-1}\mathbf{B})$ the systematic form of $\mathbf{U}$.

Encap($\mathbf{H}$):
    - Choose uniformly at random $E$ of dimension $r$.
    - Sample uniformly $\mathbf{V} \xleftarrow{\$} E^{n\times N}$
    - Output $\mathbf{C} = \mathbf{HV}$
    - Define $K = \mathsf{H}(E)$

Decap($\mathbf{C}, \mathbf{U}$):
    - Compute $\mathbf{S} = \mathbf{AC}$
    - Recover $E \leftarrow \mathsf{RSR}(F, \mathbf{S}, r)$
    - Return $K = \mathsf{H}(E)$ or $\perp$ if RSR failed.
```

Fig. 2: Algorithms KeyGen, Encap and Decap of the Key Encapsulation Mechanism ILRPC-MS

2.5 Augmented Gabidulin codes

Augmented Gabidulin codes have been introduced in [17]. They are an improvement of Gabidulin codes, which can be seen as an analog of Reed-Solomon codes in rank metric, where standard polynomials are replaced by q-polynomials. The main idea behind these codes is to add a sequence of zeros at the end of the Gabidulin codes; by doing this, one directly gets elements of the support of the error, which correspond to *support erasure* in a rank metric context. More precisely, *support erasures* are defined as a subspace of the vector space spanned by the coordinates of the error, i.e. the support of the error.

We will only recall here the basic definitions and properties, the interested reader can found more details and proofs in [17].

Definition 12 (Augmented Gabidulin codes). *Let $(k, n, n', m) \in \mathbb{N}^4$ such that $k \leq n' < m < n$. Let $\boldsymbol{g} = (g_1, \dots, g_{n'})$ be an $\mathbb{F}_q$- linearly independent family of n' elements of $\mathbb{F}_{q^m}$ and let $\overline{\boldsymbol{g}}$ be the vector of* *length n which is equal to $\boldsymbol{g}$ padded with $n-n'$ extra zeros on the right. The* Augmented Gabidulin code $\mathcal{G}^{+}_{\boldsymbol{g}}(n,n',k,m)$ *is the code of parameters $[n,k]_{q^m}$ defined by:*

$$\mathcal{G}^{+}_{\overline{\boldsymbol{g}}}(n,n',k,m) := \left\{P(\overline{\boldsymbol{g}}),\ \deg_q(P) < k\right\},$$

where $P(\overline{\boldsymbol{g}}) := (P(g_1),\ldots,P(g_{n'}),0,\ldots,0)$.

Proposition 1 (Decoding capacity of Augmented Gabidulin codes). *Let $\mathcal{G}^{+}_{\overline{\boldsymbol{g}}}(n,n',k,m)$ be an augmented Gabidulin code, and let $\varepsilon \in \{1,2,\ldots,\min(n-n',n'-k)\}$ be the dimension of the vector space generated by the support erasures. Then, $\mathcal{G}^{+}_{\overline{\boldsymbol{g}}}(n,n',k,m)$ can uniquely decode an error of rank weight up to*

$$t := \left\lfloor \frac{n'-k+\varepsilon}{2} \right\rfloor.$$

Proposition 2 (Decoding Algorithm for Augmented Gabidulin codes). *Let $\mathcal{G}^{+}_{\overline{\boldsymbol{g}}}(n,n',k,m)$ be an augmented Gabidulin code, and let $\varepsilon \in \{1,2,\ldots,\min(n-n',n'-k)\}$ be the dimension of the vector space generated by the support erasures. This code benefits from an efficient decoding algorithm correcting errors of rank weight up to $\delta := \left\lfloor \frac{n'-k+\varepsilon}{2} \right\rfloor$ with a decryption failure rate (DFR) of:*

$$q^{\delta(n'-n)} \sum_{i=1}^{\epsilon} \prod_{j=0}^{i-1} \frac{(q^{\delta}-q^{j})(q^{n-n'}-q^{j})}{q^{i}-q^{j}}$$

2.6 RQC-MS-AG scheme

An encryption scheme presented later is an improvement of the RQC-MS-AG (RQC - Multi Syndrome - Augmented Gabidulin) Public Key Encryption (PKE) scheme, introduced in [17], that we present here. We briefly recall the two initial Multi-RQC-AG schemes (there are two: the standard scheme and the other with non homogeneous errors), for which we need the following notations:

$$\mathcal{S}^{n}_{\omega,1}(\mathbb{F}_{q^m}) = \{\mathbf{x} \in \mathbb{F}^{n}_{q^m} \mid \|x\| = \omega, 1 \in \mathsf{Supp}(\mathbf{x})\}$$

$$\begin{aligned}\mathcal{S}^{3n}_{(\omega_1,\omega_2)}(\mathbb{F}_{q^m}) =\{&\mathbf{x} = (\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3) \in \mathbb{F}^{3n}_{q^m} \mid \\ &\|(\mathbf{x}_1,\mathbf{x}_3)\| = \omega_1, \\ &\|\mathbf{x}_2\| = \omega_1+\omega_2, \\ &\mathsf{Supp}(\mathbf{x}_1,\mathbf{x}_3) \subset \mathsf{Supp}(\mathbf{x}_2)\}\end{aligned}$$

Let n_1 and n_2 two integers, and let $P \in \mathbb{F}_q[X]$ an irreducible polynomial of degree n_2. Two types of codes are used in the RQC scheme: a first random $[2n_2,n_2]_{q^m}$ ideal code with parity check matrix $\begin{pmatrix}\mathbf{1}\ \mathbf{h}\end{pmatrix}$ which permits to ensure the security of the scheme, and a second public code which permits to code and decode the ciphertext. Augmented Gabidulin codes are used as public decryption codes, thanks to their high capacity of decoding. The resulting schemes can be found in Figure 3 and Figure 4.

2.7 Blockwise errors and related problems

The block-wise errors have been recently introduced in [30], where they exploit a particular structure of the error to increase the capacity of decoding.

Definition 13 (Blockwise ℓ-error). *Let $\boldsymbol{n} = (n_1,...,n_\ell) \in \mathbb{N}^\ell$ and $\boldsymbol{r} = (r_1,...,r_\ell) \in \mathbb{N}^\ell$ vectors of integers, and $n = \sum_{i=1}^{\ell} n_i$. We say that an error $\boldsymbol{e} \in \mathbb{F}^{n}_{q^m}$ is an ℓ-error with parameters $\boldsymbol{n}$ and $\boldsymbol{r}$ if it is the concatenation of ℓ vectors $\boldsymbol{e} = (\boldsymbol{e}_1,...,\boldsymbol{e}_\ell)$ such that:*

- *For all $i \in \{1,...,\ell\}$, the vector $\boldsymbol{e}_i \in \mathbb{F}^{n_i}_{q^m}$ has rank-weight r_i,*

```
$\mathsf{KeyGen}(1^\lambda)$:
    - Sample uniformly $\mathbf{h} \xleftarrow{\$} \mathbb{F}_{q^m}^n$, $\mathbf{g} \xleftarrow{\$} \mathcal{S}_{n'}^{n'}(\mathbb{F}_{q^m})$ and $(\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathcal{S}_{\omega,1}^{2n_2}(\mathbb{F}_{q^m})$.
    - Compute $\mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y} \mod P$
    - Output $\mathsf{pk} = (\mathbf{g}, \mathbf{h}, \mathbf{s})$ and $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$

$\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m})$:
    - Compute the generator matrix $\mathbf{G} \in \mathbb{F}_{q^m}^{k \times n_1 n_2}$ of the code $\mathcal{G}_{\mathbf{g}}^{+}(n_1 n_2, n', k, m)$
    - Sample uniformly at random $(\mathbf{R}_1, \mathbf{E}, \mathbf{R}_2) \xleftarrow{\$} \mathcal{S}_{\omega_1,\omega_2}^{n_2 \times 3n_1}(\mathbb{F}_{q^m})$
    - Compute $\mathbf{U} = \mathbf{R}_1 + \mathbf{H} \cdot \mathbf{R}_2 \mod P$ and $\mathbf{V} = \mathbf{mG} + \mathbf{s} \cdot \mathbf{R}_2 + \mathbf{E}$
    - Output $\mathbf{C} = (\mathbf{U}, \mathbf{V})$

$\mathsf{Decrypt}(\mathsf{sk}, \mathbf{C})$:
    - Output $\mathcal{G}_{\mathbf{g}}^{+}.\mathsf{Decode}(\mathsf{Unfold}(\mathbf{V} - \mathbf{y} \cdot \mathbf{U}))$
```

Fig. 3: Description of the RQC-AG-MS scheme

```
$\mathsf{KeyGen}(1^\lambda)$:
    - Sample uniformly $\mathbf{H} \xleftarrow{\$} \mathbb{F}_{q^m}^{n \times n}$, $\mathbf{g} \xleftarrow{\$} \mathcal{S}_{n'}^{n'}(\mathbb{F}_{q^m})$ and $(\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathcal{S}_{\omega,1}^{n \times 2n_1}(\mathbb{F}_{q^m})$.
    - Compute $\mathbf{S} = \mathbf{X} + \mathbf{HY}$
    - Output $\mathsf{pk} = (\mathbf{g}, \mathbf{H}, \mathbf{S})$ and $\mathsf{sk} = (\mathbf{X}, \mathbf{Y})$

$\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m})$:
    - Compute the generator matrix $\mathbf{G} \in \mathbb{F}_{q^m}^{k \times n_1 n_2}$ of the code $\mathcal{G}_{\mathbf{g}}^{+}(n_1 n_2, n', k, m)$
    - Sample uniformly at random $(\mathbf{R}_1, \mathbf{E}, \mathbf{R}_2) \xleftarrow{\$} \mathcal{S}_{\omega_1,\omega_2}^{n_2 \times (n, n_1, n)}(\mathbb{F}_{q^m})$
    - Compute $\mathbf{U} = \mathbf{R}_1 + \mathbf{R}_2\mathbf{H}$ and $\mathbf{V} = \mathsf{Fold}(\mathbf{mG}) + \mathbf{R}_2 \cdot \mathbf{S} + \mathbf{E}$
    - Output $\mathbf{C} = (\mathbf{U}, \mathbf{V})$

$\mathsf{Decrypt}(\mathsf{sk}, \mathbf{C})$:
    - Output $\mathcal{G}_{\mathbf{g}}^{+}.\mathsf{Decode}(\mathsf{Unfold}(\mathbf{V} - \mathbf{UY}))$
```

Fig. 4: Description of the RQC-AG-MS scheme with non-homogeneous errors

– *For all* $i \neq j$, $\mathsf{Supp}(\boldsymbol{e}_i) \cap \mathsf{Supp}(\boldsymbol{e}_j) = \{0\}$

We denote $S_{\mathbf{r}}^{\mathbf{n}}$ as the set of blockwise errors with parameters $\mathbf{n}$ and $\mathbf{r}$. For an integer N and vectors $\mathbf{n}$ and $\mathbf{r}$, we can similarly define $S_{\mathbf{r}}^{N \times \mathbf{n}}$, the set of matrices of size $N \times n_i$ whose elements are block matrices $\{\mathbf{M} = (\mathbf{M}_1, ..., \mathbf{M}_\ell)\}$ such that $\mathbf{M}_i \in \mathbb{F}_{q^m}^{N \times n_i}$ and all its entries lying in a subspace of dimension r_i.

Let $\mathcal{V} = (\mathcal{V}_i)_{i \in \{1,...,\ell\}}$ a finite sequence of subspaces of $\mathbb{F}_{q^m}$ such that $\dim \mathcal{V}_i = r_i$ and for all $i \neq j$: $\mathcal{V}_i \cap \mathcal{V}_j = \{0\}$. We denote $S_{\mathbf{r}}^{\mathbf{n}}(\mathcal{V})$ the set of vectors of the form $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_\ell)$, such that for all $i \in \{1, ..., \ell\}$, the coefficients of $\mathbf{x}_i$ belongs to $\mathcal{V}_i$. Similarly, we denote as $S_{\mathbf{r}}^{N \times \mathbf{n}}(\mathcal{V})$ the set of matrices of size $N \times n$ whose lines belong to $S_{\mathbf{r}}^{\mathbf{n}}(\mathcal{V})$.

We can naturally define a generalization of the RD and IRSD problems by considering only blockwise errors.

Definition 14 ($\ell - \mathsf{RD}$ problem). *Let* $\boldsymbol{n} = (n_1, ..., n_\ell) \in \mathbb{N}^\ell$ *and* $\boldsymbol{r} = (r_1, ..., r_\ell) \in \mathbb{N}^\ell$ *vectors of integers, and* $n = \sum_{i=1}^{\ell} n_i$. *Given* $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k \times n}$ *of full rank,* $\boldsymbol{y} = \boldsymbol{xG} + \boldsymbol{e}$ *where* $\boldsymbol{x}$ *is uniformly sampled from* $\mathbb{F}_{q^m}^k$ *and* $\boldsymbol{e} \in S_{\boldsymbol{r}}^{\boldsymbol{n}}$, *the Blockwise Rank Decoding problem* $\mathsf{RD}(\boldsymbol{n}, k, \boldsymbol{r}, m)$ *asks to find* $\boldsymbol{x}$ *and* $\boldsymbol{e}$.

Definition 15 ($\ell - \mathsf{IRSD}$ problem). *Let* $\boldsymbol{n} = (n_1, ..., n_\ell) \in \mathbb{N}^\ell$ *and* $\boldsymbol{r} = (r_1, ..., r_\ell) \in \mathbb{N}^\ell$ *vectors of integers, and* $n = \sum_{i=1}^{\ell} n_i$. *Let* $\boldsymbol{H}$ *the parity check matrix of a* $[sn, n]$ *ideal code. On input* $(\boldsymbol{H}, \boldsymbol{s})$ *where* $\boldsymbol{s}^t = \boldsymbol{He}^t$ *and* $\boldsymbol{e} \in S_{\boldsymbol{r}}^{\boldsymbol{n}}$, *the Blockwise Ideal Rank Syndrome Decoding problem* $\mathsf{IRSD}(\boldsymbol{n}, k, \boldsymbol{r}, m)$ *asks to find* $\boldsymbol{e}$.

An improved version of LOCKER based on the 2-IRSD problem have been proposed in [30]. The scheme is presented in Figure 5. The algorithm denoted RSR is similar to the previous defined in Figure 1, but suitable for decoding blockwise errors.

```
$\mathsf{KeyGen}(1^\lambda)$:
    - Sample uniformly $(\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathcal{S}^{(n_1,n_2)}_{(d_1,d_2)}(\mathbb{F}_{q^m})$.
    - Compute $\mathbf{h} = \mathbf{x} \cdot \mathbf{y}^{-1} \mod P$
    - Output $\mathsf{pk} = \mathbf{h}$ and $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$

$\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m})$:
    - Sample uniformly at random $(\mathbf{e}_1, \mathbf{e}_2) \xleftarrow{\$} \mathcal{S}^{(n_1,n_2)}_{(r_1,r_2)}(\mathbb{F}_{q^m})$
    - Compute $E_1 = \mathsf{Supp}(\mathbf{e}_1)$, $E_2 = \mathsf{Supp}(\mathbf{e}_2)$, $E = E_1 + E_2$ and $cipher = \mathbf{m} \oplus \mathsf{H}(E)$
    - Compute $\mathbf{c} = \mathbf{e}_1 + \mathbf{e}_2 \cdot \mathbf{h}$ and output $\mathsf{ct} = (cipher, \mathbf{c})$.

$\mathsf{Decrypt}(\mathsf{sk}, \mathsf{ct})$:
    - Compute $\mathbf{xc} = \mathbf{xe}_1 + \mathbf{ye}_2$, and retrieve $(E_1, E_2) = \mathsf{RSR}((\mathbf{x}, \mathbf{y}), \mathbf{xc}, r_1, r_2)$
    - Compute $E = E_1 + E_2$, and output $\mathbf{m} = cipher \oplus \mathsf{H}(E)$
```

Fig. 5: Description of the Blockwise LOCKER scheme

3 ℓ-LRPC codes and their decoding

In this section we study multiple decoding algorithms for ℓ-LRPC codes, in particular, we focus on the decoding of ℓ-errors. First we present the decoding algorithm as well as the analysis of the decoding failure rate (DFR) from [30], adapted to decode multiple syndromes. We then generalize this algorithm using the techniques from [3].

3.1 New problems related to blockwise errors

We introduce in this subsection new problems of decoding when several syndromes associated to errors with the same blockwise support are given in input. The security of our new schemes is conditioned by the hardness of these new problems.

Definition 16 (ℓ-RSL problem). *Given $(\boldsymbol{H}, \boldsymbol{H}\boldsymbol{E}^t)$, where $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ is of full rank, and $\boldsymbol{E} = (\boldsymbol{E}_1, ..., \boldsymbol{E}_\ell) \in \mathbb{F}_{q^m}^{N\times n}$ is a block matrix such that for all $i \in \{1, ..., \ell\}$, $\boldsymbol{E}_i \in \mathbb{F}_{q^m}^{N\times n_i}$ has all its entries lying in a subspace $\mathcal{V}_i$ of dimension r_i, the Blockwise - Rank Support Learning Problem ℓ-$\mathsf{RSL}(m, \boldsymbol{n}, \boldsymbol{r}, k, N)$ asks to find the set of subspaces $(\mathcal{V}_i)_{i\in\{1,...,\ell\}}$.*

We can also define a variant of this problem in the case of an ideal code $[sn, n]_{q^m}$, whose systematic matrix is defined by a set of polynomials $(\mathbf{h}_i)_{i\in\{1,...,s-1\}}$. We consider only the particular case of s-errors whose the s blocks have the same length n.

Definition 17 (s-IRSL problem). *Let $\boldsymbol{H}$ the parity check matrix of a $[sn, n]_{q^m}$ ideal code. Let $\boldsymbol{r} = (r_1, ..., r_s) \in \mathbb{N}^s$ and $\boldsymbol{n} = (n, ...n) \in \mathbb{N}^s$. On input $(\boldsymbol{H}, \boldsymbol{S}) \in \mathbb{F}_{q^m}^{(s-1)n\times sn} \times \mathbb{F}_{q^m}^{N\times(s-1)n}$, the Blockwise - Ideal - Rank Support Learning Problem $\mathsf{IRSL}(s, n, r, N)$ asks to compute a set of s subspaces $E = (E_1, ..., E_s)$ in $\mathbb{F}_{q^m}$ such that $\dim E_i = r_i$, $E_i \cap E_j = \{0\}$ for $i \neq j$, and such that there exists a matrix $\boldsymbol{V} \in S_{\boldsymbol{r}}^{N\times\boldsymbol{n}}(E)$ such that $\boldsymbol{H}\boldsymbol{V}^t = \boldsymbol{S}^t$*

3.2 Decoding algorithm and multiple syndromes

We recall the definition of ℓ-LRPC codes from [30]:

Definition 18. *Let $\boldsymbol{H} = (\boldsymbol{H}_1, \ldots, \boldsymbol{H}_\ell)$ be an $(n-k) \times n$ matrix over $\mathbb{F}_{q^m}$ such that:*

- *The coefficients of the submatrix $\boldsymbol{H}_i \in \mathbb{F}_{q^m}^{(n-k)\times n_i}$ generate an $\mathbb{F}_q$-subspace F_i of $\mathbb{F}_{q^m}$ of small dimension d_i,*
- *The support of all these submatrices are mutually disjoint: $F_i \cap F_j = \emptyset$ for all $i \neq j$.*

Let $\mathcal{C}$ be the code with parity-check matrix $\boldsymbol{H}$. By definition $\mathcal{C}$ is an ℓ-LRPC code.

The decoding algorithm of ℓ-LRPC codes is described Algorithm 1. We refer the reader to [9, 30] for the proofs of correctness of this decoding algorithm.

Algorithm 1 Decoding algorithm of ℓ-LRPC codes for ℓ-errors

```
Input: A collection of syndromes $(\mathbf{s}_1, \ldots, \mathbf{s}_N) \in \mathbb{F}_{q^m}^{(n-k)\times N}$ and the parity-check matrix $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times k}$
Output: The $\ell$-error $\mathbf{e}$, or error
  Compute the syndrome space $S = \langle s_{1,1}, \ldots s_{N,n-k} \rangle$
  Let $\{F_{i1}, \ldots F_{id_i}\}$ be a basis of $F_i$ for all $i$
  Compute $S_{ij} = F_{ij}^{-1} S$ for all $i \in \{1, \ldots, \ell\}$ and $j \in \{1, \ldots, d_i\}$
  Compute $E_i = \bigcap_{j=1}^{d_i} S_{ij}$
  if $dim(E_i) \neq r_i$ for any $i$ then
      return error
  else
      Recover $E = \sum_{i=1}^{\ell} E_i$
      Solve the linear system $\mathbf{He} = \mathbf{s}$ with $\mathbf{e} \in E^n$ as unknown
      return $\mathbf{e}$
```

Algorithm 1 has a non-null DFR. There are two cases that makes the algorithm fail:

1. The dimension of the syndrome space S is lower than the dimension of the whole product space $\sum_{i=1}^{\ell} E_i F_i$,
2. There exists i such that $E_i \supsetneq \bigcap_{j=1}^{d_i} S_{ij}$.

An upper bound of the decoding failure rate is given in Theorem 1.

Theorem 1. *Let $\mu = \sum_{i=1}^{\ell} r_i d_i$ and let N be the number of given syndromes. Under the assumption that each s_{ij} behaves like a random element of $\sum_{i=1}^{\ell} E_i F_i$, the decoding failure probability of ℓ-LRPC codes is bounded by:*

$$q^{-(N(n-k)-\mu)} + \sum_{i=1}^{\ell} q^{-(d_i-1)(m-\mu)+\mu-r_i}$$

To prove this theorem we use the following result from [7]:

Proposition 3. *Let r and d be two integers. Let E be a fixed subspace of dimension r and let $R_i, 1 \leqslant i \leqslant d$, be d independently chosen random subspaces if dimension rd containing the subspace E. The probability that $dim \bigcap_{i=0}^{d} R_i > r$ is bounded from above by:*

$$q^{rd-r}\left(\frac{q^{rd}-q^r}{q^m}\right)^{d-1} \approx q^{-(d-1)(m-rd-r)}$$

Proof. First we study the probability that $dim(S) < dim(\sum_{i=1}^{\ell} E_iF_i)$.

Each s_{ij} is an element of the product space $P = \sum_{i=1}^{\ell} E_iF_i$. We can thus write the syndromes $(\mathbf{s}_1, \dots, \mathbf{s}_N)$ as an $N(n-k) \times \mu$ matrix by unfolding each s_{ij} in a basis of P. By assumption, each s_{ij} behaves like a random element of $\sum_{i=1}^{\ell} E_iF_i$, thus the probability that $dim(S) < dim(P)$ is equal to the probability that a random $N(n-k) \times \mu$ matrix is not full rank, which is not more than $q^{-(N(n-k)-\mu)}$ (see [9] for a proof of this upper bound).

The second case which leads to a decoding failure is the case where there exists i such that $E_i \supsetneq \bigcap_{j=1}^{d_i} S_{ij}$.

From Proposition 3 we have that for each $1 \leqslant i \leqslant \ell$, the probability that $E_i \supsetneq \bigcap_{j=1}^{d_i} S_{ij}$ can be upper bounded by $q^{-(d_i-1)(m-\mu-r_i)}$. We need to recover E_i for all $1 \leqslant i \leqslant \ell$, hence the result.

3.3 Extended decoding algorithm

Using the techniques from [3], we extend Algorithm 1 to reduce its DFR. The resulting algorithm is Algorithm 2.

Algorithm 2 Decoding algorithm of ℓ-LRPC codes for ℓ-errors

```
Input: A collection of syndromes $(\mathbf{s}_1, \dots, \mathbf{s}_N) \in \mathbb{F}_{q^m}^{(n-k)\times N}$, the parity-check matrix $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times k}$ and an algorithm parameter $c$
Output: The $\ell$-error $\mathbf{e}$, or error
  Compute the syndrome space $S = \langle s_{1,1}, \dots s_{N,n-k} \rangle$
  Let $\{F_{i1}, \dots F_{id_i}\}$ be a basis of $F_i$ for all $i$
  Compute $S_{ij} = F_{ij}^{-1}S$ for all $i \in \{1, \dots, \ell\}$ and $j \in \{1, \dots, d_i\}$
  Compute $E_i = \bigcap_{j=1}^{d_i} S_{ij}$
  if $dim(E_i) > r_i + c$ for any $i$ then
      return error
  else
      $E' = \sum_{i=1}^{\ell} E_i$
      Solve the linear system $\mathbf{He} = \mathbf{s}$ with $\mathbf{e} \in E'^n$ as unknown
      return $\mathbf{e}$
```

Correctness of the Algorithm 2. The linear system $\mathbf{He} = \mathbf{s}$ has $(n-k)m$ equations in $\mathbb{F}_q$ and $\sum_{i=1}^{\ell} n_i(r_i+c)$ unknowns. As long as the system has more equations than unknowns, then it will have a unique solution with overwhelming probability. The rest of the algorithm works the same way as Algorithm 1.

Theorem 2. *Let $\mu = \sum_{i=1} r_id_i$ and let N be the number of given syndromes. Under the assumption that each s_{ij} behaves like a random element of $\sum_{i=1}^{\ell} E_iF_i$, the decoding failure probability (DFR) of the extended decoding algorithm for ℓ-LRPC codes is bounded by:*

$$q^{-(N(n-k)-\mu)} + \sum_{i=1}^{\ell} q^{\frac{1}{\phi(q^{-1})}(c+1)(\mu-r_i-(c+1)+(d_i-1)(\mu-m))}$$

Where ϕ is the Euler function given by:

$$\phi(x) = \prod_{k=1}^{\inf}(1-x^k) for \|x\| < 1$$

Proof. The probability that the dimension of the syndrome space is lower than the dimension of the product space is the same as for Algorithm 1.

For each $1 \leqslant i \leqslant \ell$, we want to compute the probability that $dim(\bigcap_{j=1}^{d_i} S_{ij}) > r_i + c$. From [3, Proposition 3] we have:

$$P(dim(\bigcap_{j=1} d_i S_{ij}) > r_i + c) \leqslant q^{\frac{1}{\phi(q^{-1})}(c+1)(\mu-r_i-(c+1)+(d_i-1)(\mu-m))}$$

Hence the result.

4 New cryptographic schemes based on support learning problem with blockwise errors

4.1 RQC-MS-AG scheme with blockwise errors

In this subsection, we improve the RQC-MS with Augmented Gabidulin codes by sample 2-errors and 3-errors rather than unstructured errors.

Let n_1 and n_2 positive integers, and $P \in \mathbb{F}_{q^m}[X]$ an irreducible polynomial of degree n_1. For a vector $\mathbf{v} \in \mathbb{F}_{q^m}^{n_2}$ and a matrix $\mathbf{M} \in \mathbb{F}_{q^m}^{n_2 \times n_1}$ whose columns are denoted $\mathbf{m}_i$ for $i \in \{1, ..., n_1\}$, we define a generalization of product of two vectors with:

$$\mathbf{v} \cdot \mathbf{M} = \left((\mathbf{v}\mathbf{m}_1)^t, ..., (\mathbf{v}\mathbf{m}_{n_1})^t\right)$$

Let $\mathbf{v} = (\mathbf{v}_1, ..., \mathbf{v}_{n_1}) \in \mathbb{F}_{q^m}^n$ with every $\mathbf{v}_i \in \mathbb{F}_{q^m}^{n_2}$ for all $i \in \{1, ..., n_1\}$. The procedure Fold turns the vector $\mathbf{v}$ into a matrix $\mathsf{Fold}(\mathbf{v}) = \left(\mathbf{v}_1^t, ..., \mathbf{v}_{n_1}^t\right) \in \mathbb{F}_{q^m}^{n_2 \times n_1}$.

Protocol. Let $\mathbf{G}$ the generator matrix of an Augmented Gabidulin $[n_1 n_2, k]_{q^m}$, that can correct up to $\delta = \left\lfloor \frac{m-k+\varepsilon}{2} \right\rfloor$ errors. There exists an efficient algorithm Decode that allows to decode noisy words $\mathbf{x} \in \mathbb{F}_{q^m}^n$. The scheme uses two other codes: a random ideal code $[2n_1, n_1]_{q^m}$ with parity check matrix $\begin{pmatrix}\mathbf{1}\ \mathbf{h}\end{pmatrix}$, and a random ideal code $[3n_1, n_1]_{q^m}$ with parity check matrix $\begin{pmatrix}\mathbf{1}\ \mathbf{0}\ \mathbf{h} \\ \mathbf{0}\ \mathbf{1}\ \mathbf{s}\end{pmatrix}$. A description of the resulting scheme can be found in Figure 6.

Proposition 4. *The decryption algorithm is valid as long as:*

$$\|\mathsf{Unfold}(\boldsymbol{x} \cdot \boldsymbol{R}_2 - \boldsymbol{y} \cdot \boldsymbol{R}_1 + \boldsymbol{E})\| \leq \delta$$

Proof. The correctness of the scheme follows from:

$$\begin{aligned}\mathbf{V} - \mathbf{y} \cdot \mathbf{U} &= \mathsf{Fold}(\mathbf{m}\mathbf{G}) + (\mathbf{x} + \mathbf{h}\mathbf{y}) \cdot \mathbf{R}_2 + \mathbf{E} - \mathbf{y} \cdot (\mathbf{R}_1 + \mathbf{h}\mathbf{R}_2) \\ &= \mathsf{Fold}(\mathbf{m}\mathbf{G}) + \mathbf{x} \cdot \mathbf{R}_2 - \mathbf{y} \cdot \mathbf{R}_1 + \mathbf{E}\end{aligned}$$

```
$\mathsf{KeyGen}(1^\lambda)$:
  - Sample randomly from the seed $\lambda$: $\mathbf{g} \xleftarrow{\$} \mathcal{S}_m^m$, $\mathbf{h} \xleftarrow{\$} \mathbb{F}_{q^m}^{n_1}$ and $(\mathbf{x}, \mathbf{y}) \xleftarrow{\$} S_{(r_\mathbf{x}, r_\mathbf{y})}^{(n_1, n_1)}$
  - Compute $\mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y} \mod P$
  - Output $\mathsf{pk} = (\mathbf{g}, \mathbf{h}, \mathbf{s})$ and $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$

$\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m})$:
  - Compute the generator matrix $\mathbf{G} \in \mathbb{F}_{q^m}^{k \times n_1 n_2}$ of the code $\mathcal{G}_\mathbf{g}^+(n_1 n_2, m, k, m)$
  - Sample $(\mathbf{R}_1, \mathbf{R}_2, \mathbf{E}) \xleftarrow{\$} S_{(r_1, r_2, r_\mathbf{e})}^{n_2 \times (n_1, n_1, n_1)}$
  - Compute $\mathbf{U} = \mathbf{R}_1 + \mathbf{h} \cdot \mathbf{R}_2$ and $\mathbf{V} = \mathsf{Fold}(\mathbf{mG}) + \mathbf{s} \cdot \mathbf{R}_2 + \mathbf{E}$
  - Output $\mathbf{C} = (\mathbf{U}, \mathbf{V})$

$\mathsf{Decrypt}(\mathsf{sk}, \mathbf{C})$:
  - Output $\mathsf{Decode}(\mathsf{Unfold}(\mathbf{V} - \mathbf{UY}))$
```

Fig. 6: Algorithms KeyGen, Encrypt and Decrypt of the RQC-MS-AG scheme with blockwise errors

Therefore:

$$\mathsf{Unfold}(\mathbf{V} - \mathbf{y} \cdot \mathbf{U}) = \mathbf{mG} + \mathsf{Unfold}(\mathbf{x} \cdot \mathbf{R}_2 - \mathbf{y} \cdot \mathbf{R}_1 + \mathbf{E}) \in \mathbb{F}_{q^m}^n$$

It does mean that the algorithm Decode will output $\mathbf{m}$ as long as:

$$\|\mathsf{Unfold}(\mathbf{x} \cdot \mathbf{R}_2 - \mathbf{y} \cdot \mathbf{R}_1 + \mathbf{E})\| \leq \delta$$

4.2 ILRPC-MS with blockwise errors

Our new scheme presented in figure 7 is an adaptation of the one in [3] with ℓ-errors, in the case of an ideal code generated by a polynomial $\mathbf{h} = \mathbf{x}^{-1}\mathbf{y}$, where $\mathbf{x}, \mathbf{y} \in \mathbb{F}_{q^m}[X]$. Decode is an algorithm that allows to decode LRPC codes in case of blockwise errors, for example Algorithm 1. Note that this algorithm output the error vector rather than its support, but finding the full vector error or its support are equivalent problems. By default when the extended decoding algorithm is not used the system is denoted by ILRPC-Block-MS, in the case where the extended algorithm of previous section is used with parameter c, the scheme is denoted by ILRPC-Block-XMS(r+c).

```
$\mathsf{KeyGen}(1^\lambda)$:
  - Choose uniformly at random two subspaces $F_1$ and $F_2$ of $\mathbb{F}_{q^m}$ of respective dimensions $d_1$ and $d_2$.
  - Sample a couple of polynomials whose coefficients belong to $F$: $(\mathbf{x}, \mathbf{y}) \xleftarrow{\$} F^n \times F^n$.
  - Compute $\mathbf{h} = \mathbf{x}^{-1}\mathbf{y}$
  - Output $\mathsf{pk} = \mathbf{h}$ and $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$

$\mathsf{Encap}(\mathsf{pk})$:
  - Choose uniformly at random $\mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2)$ such that $\dim \mathcal{V}_i = r_i$ and $\mathcal{V}_1 \cap \mathcal{V}_2 = \{0\}$
  - Sample uniformly $N$ polynomials whose coefficients belong to $S_\mathbf{r}^{(n,n)}(\mathcal{V})$: $(\mathbf{e}_1, ..., \mathbf{e}_N) \xleftarrow{\$} (S_\mathbf{r}^{(n,n)}(\mathcal{V}))^N$
  - Write each vector $\mathbf{e}_i$ as concatenation of $\mathbf{e}_{i,1}$ and $\mathbf{e}_{i,2}$, i.e. $\mathbf{e}_i = (\mathbf{e}_{i,1} | \mathbf{e}_{i,2})$
  - Compute $\mathbf{c}_i = \mathbf{e}_{i,1} + \mathbf{e}_{i,2}\mathbf{h}$
  - Define $K = \mathsf{H}(\mathcal{V})$ and output $\mathbf{c} = (\mathbf{c}_1, ..., \mathbf{c}_N)$

$\mathsf{Decap}(\mathsf{sk}, \mathbf{c})$:
  - Compute $\mathbf{S} = (\mathbf{xc}_1, ..., \mathbf{xc}_N)$
  - Recover $\mathcal{V} \leftarrow \mathsf{Decode}(F, \mathbf{S}, \mathbf{r})$
  - Return $K = \mathsf{H}(\mathcal{V})$ or $\perp$ if the Decode algorithm failed.
```

Fig. 7: Algorithms KeyGen, Encap and Decap of the Key Encapsulation Mechanism ILRPC-Block-MS

5 Combinatorial attacks

In this section, we present combinatorial attacks against three difficult problems adapted to blockwise errors:

1. For the $\ell - \mathsf{RD}$ problem, we present an adaptation of the AGHT attack, different from [30], as well as a new attack called *Shortening and Truncating*. We compare these attacks on a specific parameter case;
2. For the $\ell - \mathsf{RSL}$ problem
3. A structural attack against ℓ-LRPC codes.

5.1 Combinatorial attacks against ℓ-RD

To study the complexity of solving the ℓ-RD problem with combinatorial attacks, we will adapt and derive the new complexity of the attacks from [10, 22, 27] to the case of ℓ-errors. Most of the times, we present results in a simplified situation where $k = n$, $n_1 = \cdots = n_\ell = n$, $r_1 \leq r_2 \leq \cdots \leq r_\ell$.

These attacks are similar to what was presented in [29], although it does not require the support to be disjoint. Another difference is that we take advantage of simplified situations as explained in the previous paragraph.

5.1.1 The Ourivski-Johansonn attack As presented in [30], the complexity of the OJ attack when $k = n$, $n_1 = \cdots = n_\ell = n$, $r_1 \leq r_2 \leq \cdots \leq r_\ell$ is

$$\mathcal{O}((m(r-1) + (n - r_1))^\omega q^{(r_1-1)(n-r_1)+r_\ell}) \tag{1}$$

5.1.2 The AGHT attack

The general idea of the AGHT attack from [10] is to first compute a parity-check matrix $\mathbf{H}'$ of the code $\mathcal{C}'$ which is generated by a parity-check matrix of the code $\mathcal{C}$ and $\mathbf{x}$, where $\mathbf{x}$ is a solution to the equation $\mathbf{H}\mathbf{x}^\intercal = \mathbf{s}^\intercal$. Then if $\mathbf{e}$ is solution to the ℓ-RD problem, any $\mathbf{e}' = \alpha\mathbf{e}, \alpha \in \mathbb{F}_{q^m}$, is solution to the system:

$$\begin{aligned} \mathbf{H}'\mathbf{e}'^\intercal =& 0 \\ \|\mathbf{e}'\| =& r \end{aligned} \tag{2}$$

The strategy to solve this system is as follows:

- Randomly sample F, a subspace of $\mathbb{F}_{q^m}$ of dimension t,
- Solve the system $\mathbf{H}'\mathbf{e}'^\intercal = 0$ with $\mathbf{e}' \in F^n$. This system has $m(n-k-1)$ equations and nt unknowns in $\mathbb{F}_q$.

Complexity of the attack.

The value t is chosen as $t = \lfloor \frac{m(n-k-1)}{n} \rfloor = m - \lceil \frac{m(k+1)}{n} \rceil$ in order to have more equations than unknowns. This way if for any $\alpha E = \alpha \mathsf{Supp}(\mathbf{e})$ we have $\alpha E \subset F$, we will obtain a codeword of weight r of $\mathcal{C}'$ and from that we can recover the solution $\mathbf{e}$ to the ℓ-RD problem. We refer the reader to [10] for more details about this step. The probability that $\alpha E \subset F$ can be approximated by:

$$\frac{q^m - 1}{q - 1} \frac{\begin{bmatrix} t \\ r \end{bmatrix}_q}{\begin{bmatrix} m \\ r \end{bmatrix}_q}$$

Which gives a total complexity of:

$$\mathcal{O}((n-k)^3m^3q^{r\lceil\frac{(k+1)m}{n}\rceil-m})$$

Adaptation to ℓ-errors

In order to adapt this algorithm to the case of ℓ-errors, we will sample ℓ different vector spaces F_i of dimension t_i, and the algorithm will succeed if $\exists\alpha$ such that $\forall i, \alpha E_i \subset F_i$. Using the same techniques as in [10] this probability can be approximated by:

$$\frac{q^m-1}{q-1}\prod_{i=1}^{\ell} q^{-r_i(m-t_i)}$$

Which gives a total complexity of:

$$\mathcal{O}((n-k)^3m^3q^{-m+\sum\limits_{i=1}^{\ell} r_i(m-t_i)}) \tag{3}$$

In the following we restrict ourselves to the case where $\forall i, n_i = \frac{n}{\ell}$.

The total complexity depends on the choice of t_is. First we must choose these values such that $\sum\limits_{i=1}^{\ell} t_in_i \leqslant m - \lceil\frac{m(k+1)}{n}\rceil$ for the system to have more equations than unknowns, and $t_i > r_i$ for having a non-zero probability that $E_i \subset F_i$. Then there are two cases:

1. All of the r_is are equal. In this case the choice of the t_is does not change the complexity, and the complexity is the same for ℓ-errors and an error of weight r.
2. The r_is are not equal. In this case the optimal strategy is to try to make perfect guesses for the smaller r_is (i.e choosing $t_i = r_i$) in order to have the highest possible value for the t_i corresponding to the highest r_i.

The more the r_is are different, the bigger the advantage of specifically targeting ℓ-errors instead of errors of weight r.

Comparison with [30].

In [30, Section 3.3], the authors propose an adaptation of the AGHT attack to the case of ℓ-errors. We claim their adaptation misestimates the complexity of ℓ-AGHT attack. We give below two arguments to support our assertion.

First, in the demonstration of their Lemma 3.5 (cf. [30, Appendix C.1]), they seem to imply that the number of subspaces of $\mathbb{F}_{q^m}$ of dimension t_2 disjoint from a fixed E_1 is exactly equal to the number of subspaces of $\mathbb{F}_{q^m}/E_1$ of dimension t_1, which is not the case. In particular, in their $\ell = 2$ example, they guess a subspace F_2 in $\mathbb{F}_{q^m}/E_1$, but in order to perform the rest of the attack, this F_2 needs to be lifted in $\mathbb{F}_{q^m}$ into a $\widehat{F}_2$. Even though F_2 contains E_2/E_1, it is not guaranteed that $\widehat{F}_2$ will contain E_2, as it depends on the choice of the representatives for the lifting.

Second, as we understood their attack, sampling F_ℓ requires a correct guess for each $E_1, \ldots, E_{\ell-1}$. Therefore $F_1, \ldots, F_{\ell-1}$ play no role in the attack, which sounds somewhat strange.

5.1.3 Hybrid shortening and truncating attack

This new attack is an hybrid between Ourivski-Johansonn and other attacks against the plain RD problem. The attack consists of reducing the problem to solving the same problem in a code with smaller dimension (shortening), and then considering only the part of the code associated to error coordinates belonging to vectorial space of dimension r_1 (truncating). Then, we obtain a Rank Decoding problem instance with a

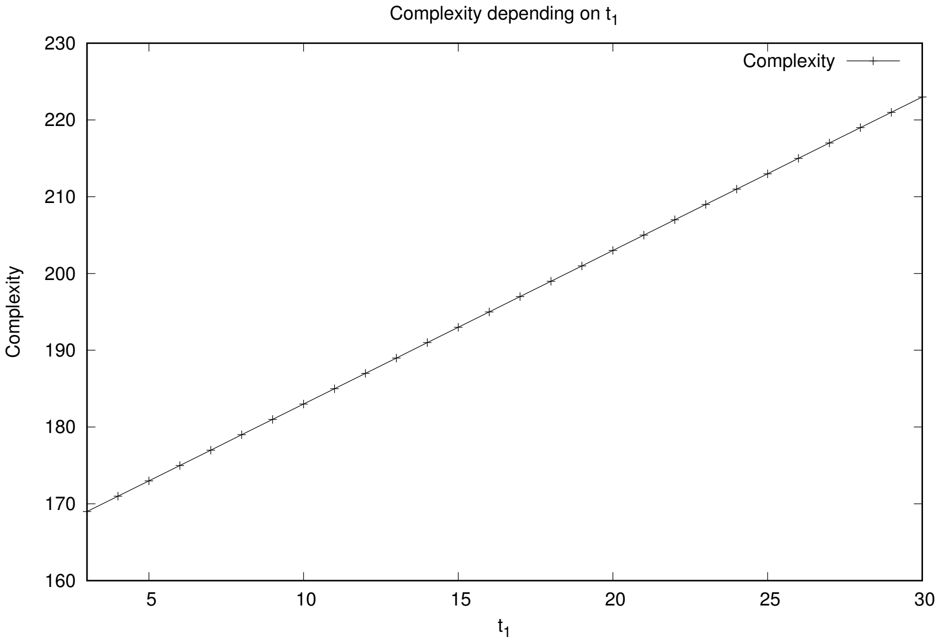

Fig. 8: Complexity of the AGHT algorithm adapted to ℓ-errors for parameters $m = 61, n = 134, k = 67, \mathbf{r} = (3, 5), l = 2$ and different values of t_1. In this case $t_2 = 60 - t_1$.

homogeneous error of smaller dimension. It is related to the hybrid attack presented in [14, Section 5.5], with the difference that the truncating part was previously unpublished.

To simplify the analysis, let us present an attack of the 2-RD problem in a code $\mathcal{C}$ of size $[2n, n]$: let $\mathbf{G} \in \mathbb{F}_{q^m}^{n\times 2n}$ the generator matrix of $\mathcal{C}$, an error $\mathbf{e} \in S_{(r_1,r_2)}^{(n,n)}$ with $(r_1, r_2) \in \mathbb{N}^2$. We reduce the problem to the resolution of a homogeneous RD problem, in a code with smaller parameters. Let $\mathbf{y} = \mathbf{x}\mathbf{G} + \mathbf{e}$ with $\mathbf{x} \in \mathbb{F}_{q^m}^n$.

We can perform $\mathbb{F}_q$-linear combinations on coordinates of $\mathbf{e}_1$, in order to obtain 0 in the first t_1 coordinates. In other words, it is possible to apply a matrix $\mathbf{P}$ with t_1 unknowns in $\mathbb{F}_q$ such that $\mathbf{eP}$ is $(0...0 \,|\, \mathbf{e}_1' \,|\, \mathbf{e}_2)$.

The attacker can then apply the same operations on the syndrome, and gets

$$\mathbf{y}' = \mathbf{y}\mathbf{P} = \mathbf{x}\mathbf{G}' + \mathbf{e}'$$

with $\mathbf{G}' = \mathbf{GP}$. Without loss of generality, the matrix $\mathbf{G}$ can be in a semi-systematic form

$$\mathbf{G}' = \left(\begin{array}{c|c} I_t & * \\ \hline 0 & * \end{array}\right)$$

Operations on the columns can then be performed to cancel to top-right block of $\mathbf{G}'$, i.e. there exists an invertible matrix $\mathbf{Q}$ such that

$$\mathbf{G}'\mathbf{Q} = \left(\begin{array}{c|c} I_t & 0 \\ \hline 0 & \mathbf{A} \end{array}\right)$$

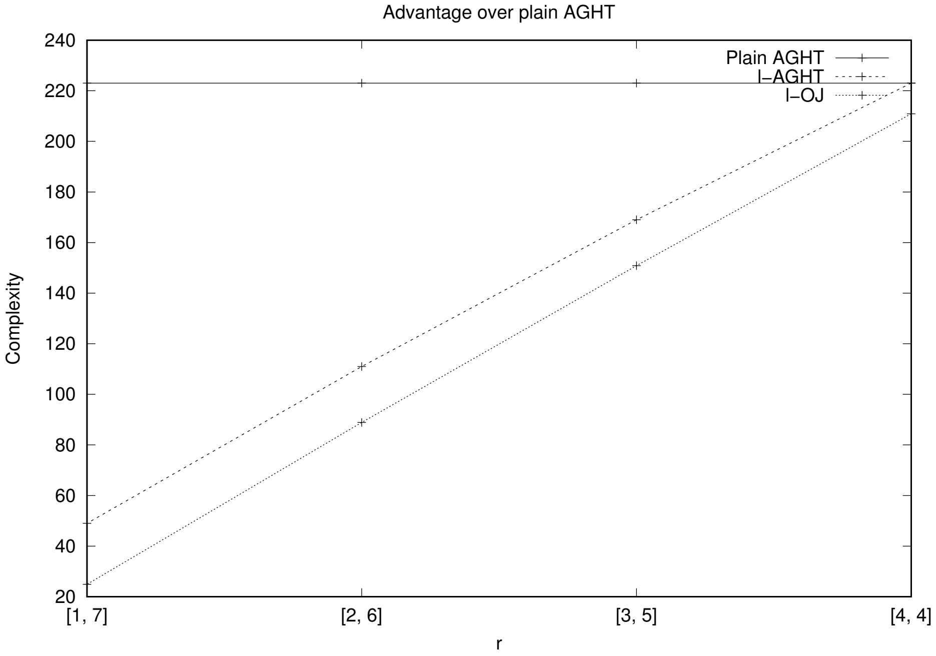

Fig. 9: Complexities of the AGHT algorithm targeting an error of rank r (plain) and adapted to ℓ-errors for parameters $m = 61, n = 134, k = 67$ and different values of $\mathbf{r}$.

Because the error $\mathbf{e}'$ has its first t coordinates set to 0, $\mathbf{e}'\mathbf{Q} = \mathbf{e}'$ hence by writing:

$$\begin{aligned} \mathbf{y}'', &\text{ the } n \text{ rightmost coordinates of } \mathbf{y}'\mathbf{Q} \\ \mathbf{x}'', &\text{ the } n-t \text{ rightmost coordinates of } \mathbf{x} \\ \mathbf{G}'', &\text{ the } n \text{ rightmost columns of } \mathbf{A} \end{aligned}$$

we get

$$\mathbf{y}'' = \mathbf{x}''\mathbf{G}'' + \mathbf{e}_2$$

which is an instance of the RD problem in a code of parameters $[n, n - t_1, r_2]$. The cost of transforming the initial instance in this reduced instance is $q^{r_1 t_1}$ (for finding the correct matrix $\mathbf{P}$) times n^2 (for calculating the matrix $\mathbf{Q}$).

By symmetry, another variant of the attack consists in canceling t_2 coordinates in the rightmost part of the error of weight r_2, and then solving an RD instance in a code with parameters $[n, n - t_2, r_1]$.

In the above explanation, the attacker *truncates* until obtaining a plain RD instance. Another possibility is to truncate only $t_1 \leq u_1 < n$ columns of $\mathbf{G}''$, yielding a 2-RD instance $(n - u_1, n)$ with weights (r_1, r_2).

We can then deduce the following proposition:

Proposition 5. *The complexity of solving the 2-*RD *problem in a code of size (n, n) by the Shortening and Truncating attack is estimated as:*

$$n^2 \cdot \min_{\substack{1\leq t_1\leq n\\ 1\leq t_2\leq n\\ t_1\leq u_1\leq n\\ t_2\leq u_2\leq n}} \left(q^{r_1t_1} \times \mathcal{T}_{2-\mathsf{RD}}([n-u_1,n],n-t_1,[r_1,r_2],m), q^{r_2t_2} \times \mathcal{T}_{2-\mathsf{RD}}([n,n-u_2],n-t_2,[r_1,r_2],m)\right) \quad (4)$$

where $\mathcal{T}_{2-\mathsf{RD}}(\boldsymbol{n},k,\boldsymbol{r},m)$ is the complexity of the best algorithm for solving an instance of $2-\mathsf{RD}(\boldsymbol{n},k,\boldsymbol{r},m)$ problem.

5.2 Combinatorial attacks against ℓ-RSL

In [17], a new combinatorial attack against the plain RSL problem was presented. It gave a new polynomial bound, i.e. a number of syndromes above which the RSL problem becomes polynomial, as well as an improved combinatorial complexity in the exponential regime adapted to the plain RSL.

Complexity of combinatorial attacks against plain RSL (from [17])

$$\begin{cases} \text{polynomial} & \text{when } N \geq kr\frac{m}{m-r} \\ \mathcal{O}\left(q^{r\left(m-\left\lfloor\frac{m(n-k)-\ell}{n-a}\right\rfloor\right)}\right) \text{ with } a=\left\lfloor\frac{N}{r}\right\rfloor & \text{when } N < kr\frac{m}{m-r} \end{cases}$$

The principle of the attack consists in exploiting the fact that there exists an $\mathbb{F}_q$-linear combination of the errors with $a = \lfloor\frac{N}{r}\rfloor$ zeros. In other words there exist scalars $(\lambda_1,\ldots,\lambda_\ell) \in \mathbb{F}_q^\ell$ and $\widetilde{\mathbf{e}} \in E^{n-a}$ such that

$$(\mathbf{0}\,|\,\widetilde{\mathbf{e}}) = \sum_{i=1}^{\ell} \lambda_i \mathbf{e}_i$$

The zeros in the error creates a reduced instance of RSD problem, but the λ_i add new unknowns to the system. Overall, when the number of equations is larger than the number of unknowns, the attacks becomes polynomial, and in the other case, an adaptation of AGHT attack is devised and gives the above complexity.

This combinatorial can be adapted to the ℓ-RSL problem and gains in efficiency due to the fact that the number a of zeros obtained by the linear combination of the errors is increased in the case of ℓ-RSL. Indeed, blockwise error coordinates belongs to a smaller subspace and can be canceled with fewer $\mathbb{F}_q$-scalars. In order to simplify the following analysis, we place ourselves in the typical case where $n_1 = n_2 = \cdots = n_\ell = n$, $r_1 \leq \cdots \leq r_\ell$ (i.e. $r_1 = \min_i(r_i)$) and the number of syndromes is reasonably small $N \leq nr_1$, which will always be the case for typical cryptographic applications.

Complexity of combinatorial attacks against ℓ-RSL
(when $n_1 = \cdots = n_\ell = n$, $r_1 = \min_i(r_i)$ and $N \leq nr_1$)

$$\mathcal{O}\left(q^{r\left(m-\left\lfloor\frac{m(n-k)-N}{n-a}\right\rfloor\right)}\right) \text{ with } a = \left\lfloor\frac{N}{r_1}\right\rfloor$$

We prove in the following proposition the above complexity. We don't provide a polynomial bound because for most parameters, it will be attained for $N > nr_1$, and in that case, the number of zeros obtained by the linear combination will overflow the first block n_1. As it would greatly complexify the analysis and that such a big number of syndromes is usually not used for cryptographic parameters, we prefer not to mention this case.

Proposition 6. *When $n_1 = n_2 = \cdots = n_\ell = n$, $r_1 \leq \cdots \leq r_\ell$ (i.e. $r_1 = \min_i(r_i)$), and $N \leq n_1 r_1$, the combinatorial attack against ℓ-$\mathsf{RSL}(m, \boldsymbol{n}, \boldsymbol{r}, k, N)$ has a complexity of*

$$\mathcal{O}(q^{r(m - \lfloor \frac{(n-k)m-N}{n-a} \rfloor)})$$

with $a = \lfloor \frac{N}{r_1} \rfloor$.

Proof. As discussed earlier, there exists an $\mathbb{F}_q$-linear combination of the errors with $a = \left\lfloor \frac{N}{r_1} \right\rfloor$ zeros.

Similarly to [17], by setting $\widetilde{\mathbf{H}} = \mathbf{H}_{*,[a+1,n]}$, an attacker can solve the linear system of $(n-k)m$ equations over $\mathbb{F}_q$

$$\widetilde{\mathbf{e}}\widetilde{\mathbf{H}}^T = \sum_{i=1}^{\ell} \lambda_i \mathbf{y}_i \tag{5}$$

whose $(n-a)m + N$ unknowns are the coordinates of $\widetilde{\mathbf{e}}$ and the λ_i.

The attacker then picks a vector space V of dimension $\widetilde{r} \geq r$ hoping that $Supp(\widetilde{e}) \subset V$. This reduces the number of unknowns in Eq. 5 to $(n-a)\widetilde{r} + N$ while the number of equations is still $(n-k)m$.

The complexity is given by the inverse of the probability that $Supp(\widetilde{e}) \subset V$, which can be calculated like before and is equal to $\mathcal{O}(q^{r(m-\widetilde{r})})$. The optimal complexity is obtained for the highest value of $\widetilde{r}$ has more equations than unknowns, $(n-a)\widetilde{r} + N = (n-k)m$, i.e. $\widetilde{r} = \lfloor \frac{(n-k)m-N}{n-a} \rfloor$, which finalizes the proof. □

5.3 A structural attack against 2-LRPC codes

It is also possible to consider structural attacks, by exploiting a possible particular structure of the code to recover the secret key $\mathbf{H}$. For example: in the case of an 2-LRPC code.

Proposition 7. *The complexity of recovering the structure of a 2-LRPC code $\mathcal{C}$ of size (n, n) by the Shortening and Truncating attack is estimated as:*

$$n^2 \cdot \min_{\substack{1 \leq t_1 \leq n \\ 1 \leq t_2 \leq n \\ t_1 + \lfloor n/d_1 \rfloor \leq u_1 \leq n \\ t_2 + \lfloor n/d_2 \rfloor \leq u_2 \leq n}} \Big(q^{r_1 t_1} \times \mathcal{T}_{2-\mathsf{RD}}([n-u_1, n], n - t_1 - \lfloor \frac{n}{d_1} \rfloor, [r_1, r_2], m), \quad q^{r_2 t_2} \times \mathcal{T}_{2-\mathsf{RD}}([n, n-u_2], n - t_2 - \lfloor \frac{n}{d_2} \rfloor, [r_1, r_2], m) \Big) \tag{6}$$

Proof. We explain using the attack described in [22] why we can reduce it to a subcode of $\mathcal{C}$ with smaller parameters.

Let $\mathbf{H} \in \mathbb{F}_{q^m}^{n \times 2n}$ the parity check matrix of $\mathcal{C}$. We can define the matrix as $\mathbf{H} = (\mathbf{H}_1 \mathbf{H}_2)$, where $\mathbf{H}_1, \mathbf{H}_2 \in \mathbb{F}_{q^m}^{n \times n}$ and $\mathbf{H}_1$ (resp. $\mathbf{H}_2$) has its coefficients belong to the same subspace F_1 (resp. F_2, disjoint to F_1) of dimension d_1 (resp. d_2).

Let $\mathcal{D}$ the dual code of $\mathcal{C}$, whose $\mathbf{H} = (\mathbf{H}_1 \mathbf{H}_2)$ is a generator matrix. We denote by $(H_i)_{i \in \{1,\ldots,n\}}$ the rows of $\mathbf{H}$, and we consider a word $\mathbf{x} \in \mathcal{D}$ obtained from linear combination in $\mathbb{F}_q$: $\mathbf{x} = \sum_{i=1}^{n} a_i H_i$, with $a_i \in \mathbb{F}_q$. Consider the block $\mathbf{H}_2$, whose coefficients belong to F_2. Since F_2 has dimension d_2, choose d_2 variables a_i correctly allows to put to 0 a coordinate of $\mathbf{x}$. Since there are n variables a_i, one can put to 0 with a good probability $\lfloor n/d_2 \rfloor$ coefficients of $\mathbf{x}$. Therefore, the dual code $\mathcal{C}^\perp$ contains with a good probability a word $\mathbf{x} = (\mathbf{x}_1 \mathbf{x}_2)$, whose the coefficients of $\mathbf{x}_1$ belongs to F_1 and the $\lfloor n/d_2 \rfloor$ first coordinates of $\mathbf{x}_2$ are equal to zero (without loss of generality). Then, the attacker can perform the Shortening and Truncating attack on $\mathcal{D}$, knowing that the dimension of the code has already been reduced. □

6 Algebraic attacks on ℓ-RD and ℓ-RSL

The algebraic attacks on ℓ-RD proposed in [30] are an adaptation of the known techniques for RD [13–15] by taking advantage of the block structure. They do not exploit the fact that the supports are pairwise disjoint. In addition to them, we should also mention the algebraic attack on RSL of [Bardet-Briaud]

6.1 MaxMinors attack

As in the most recent combinatorial attacks, RD is reduced to the problem of finding a weight r codeword in the code $\mathcal{C}_{\mathbf{y}} \overset{def}{=} \mathcal{C} \oplus \langle \mathbf{y} \rangle_{\mathbb{F}_{q^m}}$. The error vector satisfies the equation

$$\mathbf{e}\mathbf{H}_{\mathbf{y}}^{\mathsf{T}} = \mathbf{0},$$

where $\mathbf{H}_{\mathbf{y}} \in \mathbb{F}_{q^m}^{(n-k-1)\times n}$ is a systematic parity-check matrix for $\mathcal{C}_{\mathbf{y}}$. Similarly to [27], we then express $\mathbf{M}(\mathbf{e}) \in \mathbb{F}_q^{m\times n}$ as a product $\mathbf{SC}$, where $\mathbf{S} \in \mathbb{F}_q^{m\times r}$ and $\mathbf{C} \in \mathbb{F}_q^{r\times n}$ are the support and coefficient matrices respectively. Finally, the matrix $\mathbf{SCH}_{\mathbf{y}}^{\mathsf{T}} \in \mathbb{F}_{q^m}^{r\times(n-k-1)}$ is not full-rank because $\boldsymbol{\beta}\mathbf{SCH}_{\mathbf{y}}^{\mathsf{T}} = \mathbf{0}$.

Modeling 1 (MaxMinors) *Let $\boldsymbol{H}_{\boldsymbol{y}} \in \mathbb{F}_{q^m}^{(n-k-1)\times n}$ be a systematic parity-check matrix for $\mathcal{C}_{\boldsymbol{y}} = \mathcal{C} \oplus \langle \boldsymbol{y} \rangle_{\mathbb{F}_{q^m}}$ and let $\boldsymbol{C} \in \mathbb{F}_q^{r\times n}$ be the secret coefficient matrix associated to $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$. The MaxMinors modeling is the system defined by $\{P_J\}_{J\subset\{1..n-k-1\},\ \#J=r}$, where*

$$P_J \overset{def}{=} \left| \boldsymbol{C}(\boldsymbol{H}_{\boldsymbol{y}}^{\mathsf{T}})_{*,J} \right|.$$

By using the Cauchy-Binet formula, this system is known to be linear (over $\mathbb{F}_{q^m}$) in the maximal minors $c_T := |\mathbf{C}|_{*,T}$ of $\mathbf{C}$ for $T \subset \{1..n\}$, $\#T = r$. As these minors are over $\mathbb{F}_q$, the attack proceeds by solving a system projected over $\mathbb{F}_q$ containing $m\binom{n-k-1}{r}$ equations.

In order to solve ℓ-RD, the authors propose to fix certain variables in the MaxMinors system. A previous attempt of the same type can be found in the RQC submission [1]. To attack an ℓ-RD instance of block size $n := \sum_{i=1}^{\ell} n_i$ and dimension k with $r := \sum_{i=1}^{\ell} r_i$, the idea is to write the coefficient matrix as

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_1 & & & \\ & \mathbf{C}_2 & & \\ & & \ddots & \\ & & & \mathbf{C}_\ell \end{pmatrix} \in \mathbb{F}_q^{r\times n},\ \mathbf{C}_i \in \mathbb{F}_q^{r_i \times n_i}.$$

If we set $n_{\leq j} := \sum_{i=1}^{j} n_i$, we notice that the minor variables that are possibly non-zero are such that $T_j := (T - n_{\leq j-1}) \cap \{1..n_j\}$ is of size r_j for $j \in \{1..\ell\}$. This allows to consider $\prod_{i=1}^{\ell}\binom{n_i}{r_i}$ unknowns instead of $\binom{n}{r}$. Moreover, such minors can be seen as product of smaller ones, *i.e.*,

$$c_T = \textstyle\prod_{i=1}^{\ell} c_{i,T_i},\ c_{i,T_i} := |\mathbf{C}_i|_{*,T_i}. \tag{7}$$

The question left open in [30] is the study of linear dependencies between the MaxMinor equations by zeroing the rest of the variables.

We attempted to study such relations on the system over $\mathbb{F}_{q^m}$, mainly for blocks of the same size. In turns out that there always exist some when $\ell \geq 3$. In that respect, the situation is comparable to that of [17]. When $\ell = 2$, there is a collision between leading terms which does not occur in the random case but we observed in our tests that the equations remained linearly independent.

Message attack. We restrict ourselves to blocks of the same size, for $\ell = 2$ and $\ell = 3$. Estimate 1 is based upon the assumption that the equations remain linearly independent when $\ell = 2$. We set $N_2(n, r_1, r_2) \overset{def}{=} \binom{n-1}{r_1+r_2}$.

Estimate 1 (2 blocks) *We expect to solve a* 2-RD *instance of parameters* $(m,\ n_1 = n,\ n_2 = n,\ k = n,\ (r_1, r_2))$ *by Gaussian elimination on the MaxMinors system whenever*

$$mN_2(n, r_1, r_2) \geq \binom{n}{r_1}\binom{n}{r_2} - 1, \tag{8}$$

with cost $\mathcal{O}\left(mN_2(n, r_1, r_2)\binom{n}{r_1}^{\omega-1}\binom{n}{r_2}^{\omega-1}\right)$, $2 \leq \omega \leq 3$. *When Equation* (8) *does not hold, we estimate the cost of the hybrid approach of by*

$$\mathcal{O}\left(\min_{\substack{(a_1,a_2) \\ mN_2(n,r_1,r_2)\geq\binom{n-a_1}{r_1}\binom{n-a_2}{r_2}-1}} \left(q^{a_1r_1+a_2r_2}mN_2(n, r_1, r_2)\binom{n-a_1}{r_1}^{\omega-1}\binom{n-a_2}{r_2}^{\omega-1}\right)\right).$$

When $\ell = 3$, we replace the total number of equations $m\binom{2n-1}{r_1+r_2+r_3}$ by the following sharper bound on the number of linearly independent equations (obtained from preliminary analysis):

$$mN_3(n, r_1, r_2, r_3) \overset{def}{=} m\sum_{j=r_2-1}^{r_1+r_2}\binom{n-1}{j}\binom{n-1}{r_1+r_2+r_3-j}.$$

On our parameters, this value is still quite close to the maximum number of equations.

Estimate 2 (3 blocks) *We expect to solve a* 3-RD *instance of parameters* $(m,\ n_1 = n,\ n_2 = n,\ n_3 = n,\ k = n,\ (r_1, r_2, r_3))$ *by Gaussian elimination on the MaxMinors system whenever*

$$mN_3(n, r_1, r_2, r_3) \geq \binom{n}{r_1}\binom{n}{r_2}\binom{n}{r_3} - 1, \tag{9}$$

with cost $\mathcal{O}\left(mN_3(n, r_1, r_2, r_3)\binom{n}{r_1}^{\omega-1}\binom{n}{r_2}^{\omega-1}\binom{n}{r_3}^{\omega-1}\right)$, $2 \leq \omega \leq 3$. *When Equation* (9) *does not hold, we estimate the cost of the hybrid approach of by*

$$\mathcal{O}\left(\min_{\substack{(a_1,a_2,a_3) \\ mN_3(n,r_1,r_2,r_3)\geq\binom{n-a_1}{r_1}\binom{n-a_2}{r_2}\binom{n-a_3}{r_3}-1}} \left(q^{a_1r_1+a_2r_2+a_3r_3}mN_3(n, r_1, r_2, r_3)\binom{n-a_1}{r_1}^{\omega-1}\binom{n-a_2}{r_2}^{\omega-1}\binom{n-a_3}{r_3}^{\omega-1}\right)\right).$$

Structural attack. In this case, we have more freedom to fix coordinates to zero in the error vector. We reduce to a problem with a unique solution with probability 1 and we then proceed as before. On an instance with parameters $(m,\ n_1 = n,\ n_2 = n,\ k = n,\ (d_1, d_2))$, we can freely

- fix b_1 on the left and then the rest $b_2 = \left\lfloor \frac{n_1+n_2-k-r_1b_1}{r_2} \right\rfloor$ on the right;
- fix b_2 zeroes on the right first and then $b_1 = \left\lfloor \frac{n_1+n_2-k-r_2b_2}{r_1} \right\rfloor$ on the left.

By doing so, we expect to attack a new instance with block size $n_1 = n - b_1,\ n_2 = n - b_2$ and with dimension $n - b_1 - b_2$. The codimension remains $(2n - b_1 - b_2) - (n - b_1 - b_2) = n$.

Estimate 3 *The complexity of this attack is* $\mathcal{O}(m \times \min(A, B))$, *where*

$$A = \min_{\substack{0\leq b_1\leq\lfloor n/d_1\rfloor \\ b_2=\left\lfloor\frac{n-r_1b_1}{d_2}\right\rfloor}} \left(\min_{\substack{(a_1,a_2) \\ mN_2(n,d_1,d_2)\geq\binom{n-b_1-a_1}{d_1}\binom{n-b_2-a_2}{d_2}-1}} q^{a_1d_1+a_2d_2}N_2(n, d_1, d_2)\binom{n-b_1-a_1}{d_1}^{\omega-1}\binom{n-b_2-a_2}{d_2}^{\omega-1}\right),$$

$$B = \min_{\substack{0\leq b_2\leq\lfloor n/d_2\rfloor \\ b_1=\left\lfloor\frac{n-d_2b_2}{d_1}\right\rfloor}} \left(\min_{\substack{(a_1,a_2) \\ mN_2(n,d_1,d_2)\geq\binom{n-b_1-a_1}{d_1}\binom{n-b_2-a_2}{d_2}-1}} q^{a_1d_1+a_2d_2}N_2(n, d_1, d_2)\binom{n-b_1-a_1}{d_1}^{\omega-1}\binom{n-b_2-a_2}{d_2}^{\omega-1}\right).$$

6.2 Attack based on Support-Minors

The Support-Minors system was introduced in [15] as a new modeling for the MinRank problem but its analysis in the context of RD was inaccurate. This was corrected in [14] where they propose the SM-$\mathbb{F}_{q^m}^+$ attack. When MaxMinors projected over $\mathbb{F}_q$ cannot be solved by direct linearization, it consists in adding the following equations:

Modeling 2 (Support-Minors for RD) *Let $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k\times n}$ be a systematic generator matrix of $\mathcal{C}$ and let $\boldsymbol{C} \in \mathbb{F}_q^{r\times n}$ be the secret coefficient matrix associated to $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$. The Support-Minors modeling is the system defined by $\{Q_I\}_{I\subset\{1..n\},\ \#I=r+1}$, where*

$$Q_I \overset{def}{=} \left| \begin{pmatrix} \boldsymbol{xG}+\boldsymbol{y} \\ \boldsymbol{C} \end{pmatrix}_{*,I} \right|.$$

This is a bilinear system in $c_T \in \mathbb{F}_q$ and $x_j \in \mathbb{F}_{q^m}$ for $j \in \{1..k\}$.

On some RD instances, it can lead to better complexities than the hybrid MaxMinors attack.

However, we observe that Support-Minors is much sparser than MaxMinors. In particular, a lot more relations are to be expected when we apply it to ℓ-RD. By Laplace expansion along the first row, the c_T variables present in Q_I are included in the set $\left\{c_{I\setminus\{i\}},\ i \in I\right\}$. Now, a $c_{I\setminus\{i\}}$ that remains after specialization is necessarily as in Equation (7). In other words, this means that $(I \setminus \{i\} - n_{\leq j-1}) \cap \{1..n_j\}$ is of size r_j for all j. It imposes that $(I - n_{\leq j-1}) \cap \{1..n_j\}$ is of size r_j except for one j where it is of size $r_j + 1$. Conversely, for such an I and j_0 for which $(I - n_{\leq j_0-1}) \cap \{1..n_{j_0}\}$ is of size $r_{j_0} + 1$ and the rest of the intersections are of size r_j, the c_T present are of the form $c_{I\setminus\{i\}}$, $i \in I \cap \{n_{\leq j_0-1} + 1..n_{\leq j_0}\}$.

We have not studied the full SM-$\mathbb{F}_{q^m}^+$ modeling. For this reason and as the progress over MaxMinors in the random case was often only by a few bits, we adopt Estimate 4:

Estimate 4 *We do not take into account SM-$\mathbb{F}_{q^m}^+$ to derive our parameters.*

6.3 Algebraic attack on ℓ-RSL

In addition to the RD attacks, we should also mention the algebraic attack on RSL of [12]. Since it involves the same minor variables as in these previous methods, it can easily be adapted to ℓ-RSL. Still, the same difficulty arises when studying algebraic relations. Since our number of syndromes is quite reduced, this attack does not seem to be limiting. We plan to strengthen this intuition by adding further details.

7 Application to cryptanalysis

In this section, we apply the above attacks on the parameters given by [30] for their improvement of Lake (ROLLO-I), based on 2-LRPC codes. There are two types of attacks to consider for the security of their parameters, the structural attacks targeting weights (d_1, d_2) and the message attacks targeting weights (r_1, r_2). In our case we propose two new structural attacks to recover the secret key of the system.

A first attack (attack1) corresponds to the attack against 2-LRPC codes explained in Section 5.3. The idea of the attack is to shorten as much as possible the block corresponding to the higher d_i, then shorten on these $\frac{n}{d_i}$ positions and then truncate the block corresponding to d_i, then one gets an homogeneous error that we can attack with algebraic attacks for homogeneous errors. It is also possible to increase the number of terms shortened by guessing zero positions on the d_i part at a cost of 2^{d_i} per new zero coordinate. In practice the best results are obtained when guessing sufficiently many more zeros coordinates the part corresponding to the case where the MaxMinor attack is the most efficient, in that case we estimated the polynomial part at the cost of n^2 as it is usually the case for attacks and parameters and also we consider $w = 2.8$ the Strassen exponent.

A second attack consists in having the same Shortening and Truncating approach but rather than truncating, we just attack directly the code with algebraic attacks for blockwise errors described in [30], notice that at the difference of Attack1, it is more efficient to shorten on the smallest r_i, which permits to better decrease the dimension of the code.

For instance consider the first parameter set n=67 m=61 $d_1 = 5, d_2 = 4$, we shorten on the $d_1 = 4$ block, there are naturally $67/5 = 13$ zero positions, if we attack an error of weigth 4 for the [67,54] code obtained, the best attack is the classical AGHT attack which gives a complexity of 170 bits. Now we may also decide to guess more zero positions, for instance guessing 13 zeros positions on the $d_1 = 5$ block comes at a cost of 2^{65} and permits to attack an error of weight 4 on a [67,41] code which hast a complexity through MaxMinor approach of 2^{54} which with the linear algebra cost gives an overall complexity of 2^{131}.

For attack2, on this parameters we shorten the $d_2 = 4$ block on 67/4=16 positions and search for a (5,4) error of blocksize (51,67) for a code of dimension 51 and length 118. The compelxity described in [30] gives a security if 119 bits, in fact it is even possible to optimize by considering that there is on the average probability 1/2 to have 17 positions at zero ((67+1)/4), which a security of 115 bits, hence 116 bits with the 1/2 probability.

The following table gathers the complexities of our cryptanalyzes.

We provide the resulting security of these parameters in Figure 10.

n	m	(d_1, d_2)	(r_1, r_2)	Security	Claimed Message attack Security	Claimed Structural attack Security	Attack 1	Attack 2
67	61	(5,4)	(4,4)	128	145	160	132	**116**
79	71	(5,5)	(5,5)	192	225	255	181	**166**
89	79	(6,5)	(5,5)	256	281	266	246	**224**

Fig. 10: Security of parameters on Lake given by [30]

Our new attack is very efficient againt LAKE parameters given in [30], outperforming by 44 bits the security for structural attacks for the 128 bits NIST type parameters.

8 Parameters

We discuss here on the security and parameters of our two new schemes: ILRPC-Block-MS and RQC-Block-MS-AG. For a given security of λ bits, we choose our parameters in order to respect two constraints: have a low decoding failure rate and resist to the attacks described in previous sections. For all our protocols, both 128 and 192 bits security level are considered. Parameters proposed are compliant with NIST security levels 1 and 3 of 143 and 207 classical bit security. The complexity of the attacks have been computed with a value of $\omega = 2.8$ as the Strassen constant.

To have available both several syndromes and blockwise errors allows to achieve excellent signature sizes: the first idea allows to obtain more coordinates to guess the support error, and the second gives syndromes relying to smaller spaces, which makes decoding easier.

8.1 Parameters of ILRPC-Block-MS

The security of the scheme relies on the hardness to solve the instance of a 2-IRSL problem on a code $[2n, n]_{q^m}$ with parity check matrix: $\begin{pmatrix} \mathbf{1} & \mathbf{h} \end{pmatrix}$, where N syndromes with the same block support of size (n, n) and dimension (r_1, r_2) are given in input. However, the attacks against 2-IRSL are not the best because the number of syndromes given is too small within the parameters we propose.

The parameters have been chosen such that the Decode algorithm is able to retrieve the support with a low failure rate. The resulting parameters for 128 and 192 bits of security are presented in Figure 11. Note that the previous version proposed parameters with $(r + 1)$ for 128 bits of security, that we removed because they

Scheme	n	m	(d_1, d_2)	(r_1, r_2)	N	DFR	Security
ILRPC-Block-xMS-128 $(r+2)$	84	59	(5,5)	(4,4)	2	-128	128
ILRPC-Block-xMS-128 $(r+5)$	84	53	(5,5)	(4,4)	2	-128	128
ILRPC-Block-xMS-192 $(r+1)$	83	83	(6,5)	(5,5)	3	-194	192
ILRPC-Block-xMS-192 $(r+2)$	79	83	(6,5)	(5,5)	3	-182	192

Fig. 11: Comparaison of parameters of ILRPC schemes

had a security of 126 bits against the best attack, so do not achieve the 143 bits of security required to be compliant with the NIST security level of 128 bits.

One may use seeds to represent the random data in order to decrease the key size. Since the ideal parity check matrix is completely determined by the polynomial $\mathbf{h}$, its size is reduced to $\lfloor \frac{nm}{8} \rfloor$ bytes. The $\mathbf{c}$ is made of N polynomials of degree n whose coefficients belong to $\mathbb{F}_{q^m}$, so its size is $\lfloor \frac{nmN}{8} \rfloor$ bytes. The resulting sizes can be found in Figure 12. The parameters we obtain compare very well with previous results: 3.8 kB for 128 bits security in [30] and 2.4 kB for the multiple syndromes approach [3]. Indeed as explained in the introductory section, the blockwise approach is essentially interesting for RQC and less for LRPC, since blockwise small weight errors are more vulnerable to the Shortening and Truncating approach of Section 5, indeed the smallest the d_i the greater the zeros set for shortening. Overall the approach becomes more interesting when one considers the XMS approach (originally described in [3]) that uses an extended decoding algorithm for LRPC, decoding algorithm that we generalize in Section 3 to the case of blockwise rank errors.

Scheme	pk (kB)	ct (kB)	pk + ct (kB)
ILRPC-Block-xMS-128 $(r+1)$	0.6	1.3	1.9
ILRPC-Block-xMS-128 $(r+2)$	0.6	1.2	1.8
ILRPC-Block-xMS-128 $(r+5)$	0.6	1.1	1.7
ILRPC-Block-xMS-192 $(r+1)$	0.8	2.6	3.4
ILRPC-Block-xMS-192 $(r+2)$	0.8	2.5	3.3

Fig. 12: Comparaison of sizes of different ILRPC schemes

8.2 Parameters of RQC-Block-MS-AG scheme

The attacks 1 and 2 relies on the algebraic attack which consists on solving the 2-IRSL (on the $[2n_1, n_1]_{q^m}$ ideal code with parity check matrix $(\mathbf{1}\ \mathbf{h})$) and 3-IRSL problem (on the $[3n_1, n_1]_{q^m}$ ideal code whose $\begin{pmatrix}\mathbf{1} & \mathbf{0} & \mathbf{h} \\ \mathbf{0} & \mathbf{1} & \mathbf{s}\end{pmatrix}$ is a parity check matrix). The attack 3 is the Shortening and Truncating attack on the 2-IRSL instance. Note that there is currently no attack that takes advantage of the ideal structure of the parity check matrix, this is why these instances are considered as difficult to solve as 2-RSL and 3-RSL instances.

We explain here what constraints the parameters must respect. The decoding algorithm takes as input n_2 vectors having the same errors support, that is to say it has $n_1 n_2$ available coordinates to compute the support. We use a public Augmented Gabidulin code of length $n_1 n_2$ and dimension k, constructed from a vector $\mathbf{g}$ of size m. Let ε the number of erasure coordinates one uses to recover the support error. The values above must be chosen such that the decoding capacity of the code thus obtained: $\delta = \lfloor \frac{m-k+\varepsilon}{2} \rfloor$, must be greater than or equal to the weight of the error which is $r_{\mathbf{x}} r_1 + r_{\mathbf{y}} r_2 + r_{\mathbf{e}}$. On the other hand, the resulting decryption failure rate (see Proposition 2) must be remain low.

The resulting parameters for 128 and 192 bits of security are presented in Figure 13. For comparison, we also present the parameters of previous versions of RQC. We observe that the different developments have made it possible to consider increasingly smaller parameters, particularly due to the weight of the error in the message to decode which decreases for the same security.

Scheme	m	n_1	q	k	ε	$r_{\mathbf{x}}$	$r_{\mathbf{y}}$	r_1	r_2	$r_{\mathbf{e}}$	n_2	Security	Att. 1	Att. 2	Att. 3
RQC-Block-MS-AG-128	43	52	2	3	32	4	4	4	4	4	2	128	145	153	154
RQC-Block-MS-AG-192	67	68	2	3	45	5	5	5	5	6	2	192	228	206	231

Fig. 13: Proposed parameters for RQC-Block-MS-AG and resistance to attacks

Scheme	m	n_1	q	k	ε	$r_{\mathbf{x}}$	$r_{\mathbf{y}}$	r_1	r_2	$r_{\mathbf{e}}$	n_2	DFR	Security
RQC-Block-MS-AG-128 (this paper)	43	52	2	3	32	4	4	4	4	4	2	-145	128
RQC-Block-128 [30]	83	79	2	7	-	4	4	4	4	4	1	-	128
RQC-NH-MS-AG-128 [17]	61	50	2	3	51	7	7	7	5	12	3	-158	128
RQC-128 [1]	127	113	2	3	-	7	7	7	7	13	1	-	128
RQC-Block-MS-AG-192 (this paper)	67	68	2	3	45	5	5	5	5	6	2	-206	192
RQC-Block-192	127	113	2	3	-	5	5	5	5	5	1	-	192
RQC-NH-MS-AG-192	79	95	2	5	65	8	8	8	5	13	2	-238	192
RQC-192	151	149	2	5	-	8	8	8	8	16	1	-	192

Fig. 14: Comparaison of parameters of different RQC schemes

The sizes of the associated public key pk and ciphertext ct are expressed in kiloBytes (kB). Ours are computing according to the following formulas: $|\mathsf{pk}| = \lceil \frac{n_1 m}{8} \rceil + \frac{2\lambda}{8}$ and $|\mathsf{ct}| = \lceil \frac{2 n_1 n_2 m}{8} \rceil$. Since $\mathbf{g}$ and $\mathbf{h}$ are uniformly sampled from their respective spaces, they can be represented as seeds of size λ bits. Only the vector $\mathbf{s} \in \mathbb{F}_{q^m}^{n_1}$ must be completely expressed in the public key pk. The ciphertext ct contains two matrices lying in $\mathbb{F}_{q^m}^{n_2 \times n_1}$. The resulting size can be found in Figure 15. The decrease in size of public key and ciphertext over time is a direct consequence of the decrease in the size of the parameters.

Scheme	Security	pk (kB)	ct (kB)	pk + ct (kB)
RQC-Block-MS-AG-128 (this paper)	128	**0.3**	**1.1**	**1.4**
RQC-Block-128 [30]	128	0.8	1.7	2.5
RQC-NH-MS-AG-128 [17]	128	0.4	2.3	2.7
RQC-128 [1]	128	1.8	3.6	5.3
RQC-Block-MS-AG-192 (this paper)	192	**0.6**	**2.2**	**2.8**
RQC-Block-192	192	1.8	3.6	5.3
RQC-NH-MS-AG-192	192	0.9	3.8	4.7
RQC-192	192	2.8	5.7	8.3

Fig. 15: Comparaison of sizes of different RQC schemes

8.3 Comparison with other schemes

For comparison, we compare our sizes with those of other encryption schemes, see Figure 16. We can see that our scheme has very competitive performances for 128 bits of security,by getting slightly smaller sizes than the lattice-based scheme KYBER.

Scheme	128 bits	192 bits
RQC-Block-MS-AG (this paper)	**1.4**	**2.8**
ILRPC-Block-MS (this paper)	**1.7**	**3.3**
KYBER [11]	1.5	2.2
BIKE [6]	3.1	6.2
HQC [2]	6.7	13.5
Classic McEliece [5]	261.2	624.3

Fig. 16: Comparaison of different schemes, the sizes represent the sum of the key and the ciphertext, expressed in kB

References

1. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Maxime Bros, Alain Couvreur, Jean-Christophe Deneuville, Philippe Gaborit, Gilles Zémor, and Adrien Hauteville. Rank quasi cyclic (RQC). Second Round submission to NIST Post-Quantum Cryptography call, April 2020.
2. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Gilles Zémor, and Jurjen Bos. HQC. Round 3 Submission to the NIST Post-Quantum Cryptography Call, June 2021. https://pqc-hqc.org/.
3. Carlos Aguilar-Melchor, Nicolas Aragon, Victor Dyseryn, Philippe Gaborit, and Gilles Zémor. Lrpc codes with multiple syndromes: near ideal-size kems without ideals. In *International Conference on Post-Quantum Cryptography*, pages 45–68. Springer, 2022.
4. Carlos Aguilar-Melchor and Philippe Gaborit. Cryptographic method for communicating confidential information.
5. Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane

Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. Classic mceliece: conservative code-based cryptography. Third round submission to the NIST post-quantum cryptography call, October 2020.

6. N. Aragon, P. Barreto, S. Bettaieb, Loic Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Güneysu, C. Aguilar Melchor, R. Misoczki, E. Persichetti, N. Sendrier, J.-P. Tillich, and G. Zémor. BIKE, November 2017. NIST Round 1 submission for Post-Quantum Cryptography.
7. Nicolas Aragon, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Adrien Hauteville, Olivier Ruatta, Jean-Pierre Tillich, Gilles Zémor, Carlos Aguilar Melchor, Slim Bettaieb, Loïc Bidoux, Bardet Magali, and Ayoub Otmani. ROLLO (merger of Rank-Ouroboros, LAKE and LOCKER). Second round submission to the NIST post-quantum cryptography call, March 2019.
8. Nicolas Aragon, Victor Dyseryn, Philippe Gaborit, Pierre Loidreau, Julian Renner, and Antonia Wachter-Zeh. Lowms: a new rank metric code-based kem without ideal structure. *Cryptology ePrint Archive*, 2022.
9. Nicolas Aragon, Philippe Gaborit, Adrien Hauteville, Olivier Ruatta, and Gilles Zémor. Low rank parity check codes: New decoding algorithms and applications to cryptography. *IEEE Transactions on Information Theory*, 65(12):7697–7717, 2019.
10. Nicolas Aragon, Philippe Gaborit, Adrien Hauteville, and Jean-Pierre Tillich. A new algorithm for solving the rank syndrome decoding problem. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2421–2425. IEEE, 2018.
11. Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, , and Damien Stehlé. Crystals-kyber. Third round submission to the NIST post-quantum cryptography call, August 2021.
12. Magali Bardet and Pierre Briaud. An algebraic approach to the rank support learning problem, 2021.
13. 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*, pages 64–93. Springer, 2020.
14. 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, 2023.
15. Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier Verbel. Improvements of algebraic attacks for solving the rank decoding and minrank problems. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020*, pages 507–536. Springer, 2020.
16. Slim Bettaieb, Loïc Bidoux, Yann Connan, Philippe Gaborit, and Adrien Hauteville. The learning with rank errors problem and an application to symmetric authentication. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2629–2633. IEEE, 2018.
17. Loïc Bidoux, Pierre Briaud, Maxime Bros, and Philippe Gaborit. Rqc revisited and more cryptanalysis for rank-based cryptography. *arXiv preprint arXiv:2207.01410*, 2022.
18. Ernest Mukhamedovich Gabidulin. Theory of codes with maximum rank distance. *Problemy peredachi informatsii*, 21(1):3–16, 1985.
19. Ernst M Gabidulin, AV Paramonov, and OV Tretjakov. Ideals over a non-commutative ring and their application in cryptology. In *Advances in Cryptology—EUROCRYPT'91: Workshop on the Theory and Application of Cryptographic Techniques Brighton, UK, April 8–11, 1991 Proceedings 10*, pages 482–489. Springer, 1991.
20. Philippe Gaborit, Adrien Hauteville, Duong Hieu Phan, and Jean-Pierre Tillich. Identity-based encryption from codes with rank metric. In *Annual International Cryptology Conference*, pages 194–224. Springer, 2017.
21. Philippe Gaborit, Gaétan Murat, Olivier Ruatta, and Gilles Zémor. Low rank parity check codes and their application to cryptography. In *Proceedings of the Workshop on Coding and Cryptography WCC*, volume 2013, 2013.
22. Philippe Gaborit, Olivier Ruatta, Julien Schrek, and Gilles Zémor. New results for rank-based cryptography. In *Progress in Cryptology–AFRICACRYPT 2014: 7th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 28-30, 2014. Proceedings 7*, pages 1–12. Springer, 2014.
23. Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In *International algorithmic number theory symposium*, pages 267–288. Springer, 1998.
24. Pierre Loidreau. A new rank metric codes based encryption scheme. In *Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings 8*, pages 3–17. Springer, 2017.
25. Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo SLM Barreto. Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In *2013 IEEE international symposium on information theory*, pages 2069–2073. IEEE, 2013.
26. Roberto W Nóbrega and Bartolomeu F Uchôa-Filho. Multishot codes for network coding using rank-metric codes. In *2010 Third IEEE International Workshop on Wireless Network Coding*, pages 1–6. IEEE, 2010.
27. Alexei V Ourivski and Thomas Johansson. New technique for decoding codes in the rank metric and its cryptography applications. *Problems of Information Transmission*, 38:237–246, 2002.

28. Gaborit Philippe and Zemor Gilles. On the hardness of the decoding and the minimum distance problems for rank codes, 2014.
29. Sven Puchinger, Julian Renner, and Johan Rosenkilde. Generic decoding in the sum-rank metric. *IEEE Transactions on Information Theory*, 68(8):5075–5097, 2022.
30. Yongcheng Song, Jiang Zhang, Xinyi Huang, and Wei Wu. Blockwise rank decoding problem and lrpc codes: Cryptosystems with smaller sizes. *Cryptology ePrint Archive*, 2023.