An extension of Overbeck's attack with an application to cryptanalysis of Twisted Gabidulin-based schemes

Alain Couvreur[1,2] and Ilaria Zappatore[3⋆]

[1] Inria
[2] LIX, CNRS UMR 7161, École Polytechnique,
Institut Polytechnique de Paris,
1 rue Honoré d'Estienne d'Orves
91120 Palaiseau Cedex, (France)
alain.couvreur@inria.fr
[3] XLIM, CNRS UMR 7252, Université de Limoges
123, avenue Albert Thomas
87060 Limoges Cedex, (France)
ilaria.zappatore@unilim.fr

Abstract. In this article, we discuss the decoding of Gabidulin and related codes from a cryptographic point of view, and we observe that these codes can be decoded solely from the knowledge of a generator matrix. We then extend and revisit Gibson and Overbeck attacks on the generalized GPT encryption scheme (instantiated with the Gabidulin code) for different ranks of the distortion matrix. We apply our attack to the case of an instantiation with twisted Gabidulin codes.

Keywords: Code-based cryptography · rank metric codes · Gabidulin codes · Overbeck's attack · twisted Gabidulin codes

Introduction

The most promising post-quantum alternatives to RSA and elliptic curve cryptography are based on error–correction based paradigms. The metric which quantifies the amount of noise, can be either Euclidean (lattice–based cryptography), Hamming (code–based cryptography) or the rank metric. The latter has been much less investigated than the first two. However, it offers an interesting range of primitives with rather short keys [2,1,3]. In addition, the Gabidulin code family benefits from a decoder that corrects any error up to a fixed threshold. This makes it possible to design schemes with a zero failure rate, such as RQC [1]. Although no rank–based submission was selected for standardization, NIST

⋆ The authors are funded by the French Agence Nationale de la Recherche through the France 2023 ANR project ANR-22-PETQ-0008 PQ-TLS and the ANR-21-CE39-0009-BARRACUDA.

encouraged the community to continue the research efforts in the design and security of rank–metric based primitives.

Historically, the first primitive based on rank metric was proposed by Gabidulin, Paramonov and Tretjakov [22]. It was a McEliece–like scheme where the structure of a Gabidulin code is hidden. This scheme was first attacked in exponential time by Gibson [26,27]. Then, Gabidulin and Ourivski proposed an improvement of the system that was resistant to Gibson's attack [21,38]. Later, Overbeck [39,40] proposed a polynomial time attack which breaks both GPT and its improvements. Gabidulin *et. al.* then introduced several variants of the GPT based on a different column scrambler $\boldsymbol{P}$, so that some entries of $\boldsymbol{P}^{-1}$ can be in $\mathbb{F}_{q^m}$ [20,17,46]. However, in [37] the authors proved that for all of the aforementioned versions, the shape of the public key is in fact unchanged and remains subject to Overbeck–type attacks.

The natural approach to circumvent Overbeck's attack is to replace the Gabidulin codes with another family equipped with an efficient decoding algorithm. However, only a few such families exist. On the one hand, there are the LRPC codes [23] which lead to the ROLLO scheme [2]. On the other hand, one can in a way deteriorate the structure of the Gabidulin codes, at the cost of a loss of efficiency of the decoder. Loidreau [34] proposed to encrypt with a Gabidulin code perturbed by some $\mathbb{F}_{q^m}$–linear operation. This proposal was subject to polynomial–time attacks for the smallest parameters [11,25,41], while, for larger parameters, it remains secure so far. In another direction, Puchinger *et. al.* [43] proposed to replace Gabidulin codes with twisted Gabidulin codes. However their proposal was only partial, since they could not provide an efficient decoder correcting up to half the minimum distance.

Our contributions. The contribution we make in this article is threefold.

First, we discuss the decoding of Gabidulin codes and twisted Gabidulin codes. Using the result of [8], we explain how to correct errors for such codes without always being able to correct up to half of the minimum distance. From a cryptographic point of view, we highlight an important observation: if in Hamming metric, decoding Reed–Solomon codes requires the knowledge of the evaluation sequence, in the rank metric, Gabidulin codes can be decoded solely from the knowledge of a generator matrix. This observation extends to twisted Gabidulin codes as soon as the decoding radius is below a certain threshold.

Second, we revisit the Overbeck's attack and propose an extension. Specifically, from a public code $\mathscr{C}_{\text{pub}}$, the original Overbeck's attack is based on the computation of $\Lambda_i(\mathscr{C}_{\text{pub}}) = \mathscr{C}_{\text{pub}} + \mathscr{C}_{\text{pub}}^q + \cdots + \mathscr{C}_{\text{pub}}^{q^i}$. For the attack to succeed, a trade–off on the parameter i must be satisfied. On the one hand, i must be large enough to rule out the random part (called *distortion matrix*) in $\mathscr{C}_{\text{pub}}$ used to mask the hidden code. On the other hand, i must be small enough so that $\Lambda_i(\mathscr{C}_{\text{pub}})$ does not to fill in the ambient space. In the this article, we propose an extension of the Overbeck's attack that limits our goal to the smallest possible i, namely $i = 1$. This relaxation is based on calculations on a certain automorphism algebra of the code $\Lambda_1(\mathscr{C}_{\text{pub}})$ and extends the range of the attack.

Third, we investigate in depth the behavior of twisted Gabidulin codes with respect to the Λ_i operator.

The aforementioned contributions lead to an attack on a variant of GPT proposed by Puchinger, Renner and Wachter–Zeh [43]. In this variant, the authors used two techniques to resist Overbeck's attack. First, they mask the code with a *distortion matrix* of very low rank. Second, they replace Gabidulin codes with twisted Gabidulin codes. The authors chose twisted Gabidulin codes $\mathscr{C}$ so that for any positive i, the code $\Lambda_i(\mathscr{C})$ may never have co-dimension 1 (see [43, Theorem 6]). In this article, we prove that the latter property is not a strong enough security assessment for twisted Gabidulin codes and that the aforementioned contributions lead directly to an attack on the Puchinger *et. al.*'s variant of GPT.

Outline of the article. The article is organized as follows. Section 1 introduces some basic notations used in this paper, as well as Gabidulin codes, their twisted version and the GPT cryptosystem. In Section 2 we first discuss the decoding of Gabidulin codes and propose an algorithm (Algorithm 1), which does not need to know the evaluation sequence. We then explain how to decode twisted Gabidulin codes, under a certain decoding radius. In Section 3, we revisit the Overbeck's attack on the GPT scheme instantiated with Gabidulin codes and we make some remarks on the structure of the generator matrix of the code obtained by applying the q-sum operator to the public key. In Section 4, we propose an extension of the Overbeck's attack to the GPT scheme instantiated with either Gabidulin or twisted Gabidulin codes. Finally, in Section 5 we examine the behavior of the q-sum operator applied to the public key of the GPT system instantiated with twisted Gabidulin codes. We then show that we can exploit the structure of its generator matrix to break the corresponding scheme using either the Overbeck's attack, or more generally, its previously proposed extension.

1 Prerequisites

In this section we introduce the basic notions we will use throughout the paper, starting with the notations used. Then, we briefly introduce the Gabidulin codes and their twisted version, and finally the GPT cryptosystem.

1.1 Notation

Let q be a prime power, $\mathbb{F}_q$ be a finite field of order q, and $\mathbb{F}_{q^m}$ be the extension field of $\mathbb{F}_q$ of degree m. In this article, vectors are represented by lowercase bold letters: $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{x}$, and matrices by uppercase bold letters $\boldsymbol{M}, \boldsymbol{G}, \boldsymbol{H}$. We also denote the space of $m \times n$ matrices with entries in a general field $\mathbb{K}$, by $\mathcal{M}_{m,n}(\mathbb{K})$. In the square case, *i.e.* $m = n$, we simplify the notation by writing $\mathcal{M}_n(\mathbb{K})$, and we denote by $\mathbf{GL}_n(\mathbb{K})$ the group of $n \times n$ invertible matrices.

1.2 Rank metric codes

Given $\boldsymbol{x} = (x_1, \ldots, x_n)$ a vector in $\mathbb{F}_{q^m}^n$, we can define its *support* as,

$$\mathrm{Supp}(\boldsymbol{x}) \stackrel{\text{def}}{=} \mathbf{Span}_{\mathbb{F}_q}\{x_1, \ldots, x_n\}$$

and

$$\mathbf{rank}_q(\boldsymbol{x}) \stackrel{\text{def}}{=} \dim(\mathrm{Supp}(\boldsymbol{x})).$$

The *rank distance* (briefly *distance*) of two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_{q^m}^n$ is

$$\mathrm{d}(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \mathbf{rank}_q(\boldsymbol{x} - \boldsymbol{y}).$$

A *rank metric code* $\mathscr{C}$ of *length* n and *dimension* k is an $\mathbb{F}_{q^m}$-vector subspace of $\mathbb{F}_{q^m}^n$. Its *minimum distance* is defined as,

$$\mathrm{d}_{min}(\mathscr{C}) \stackrel{\text{def}}{=} \min_{\boldsymbol{x} \in \mathscr{C} \setminus \{0\}} \{\mathbf{rank}_q(\boldsymbol{x})\}.$$

By choosing an $\mathbb{F}_q$-basis $\mathcal{B}$ of $\mathbb{F}_{q^m}$, any codeword $\boldsymbol{c} \in \mathscr{C}$ can be written as a matrix $\boldsymbol{M}_{\mathcal{B}}(\boldsymbol{c}) \in \mathcal{M}_{m,n}(\mathbb{F}_q)$ by representing any element $c_i \in \mathbb{F}_{q^m}$ as a column vector whose entries are its coefficients in the basis $\mathcal{B}$. With this point of view, one can introduce a second notion of support which is less considered in the literature but will be useful in the sequel.

Definition 1. *The* row support *RowSupp*$(\boldsymbol{c})$ *of a vector* $\boldsymbol{c} \in \mathbb{F}_{q^m}^n$ *is the row span of the* $m \times n$ *matrix* $\boldsymbol{M}_{\mathcal{B}}(\boldsymbol{c})$.

Note that the row support of a vector is an intrinsic notion that does not depend on the choice of the basis $\mathcal{B}$. Moreover, as for the support, the rank of a vector equals its row support.

Remark 1. One could have defined rank metric codes as spaces of matrices endowed with the same rank metric. Such a framework is more general than ours since a matrix subspace of $\mathcal{M}_{m,n}(\mathbb{F}_q)$ is not $\mathbb{F}_{q^m}$–linear in general. But considering such rank metric codes would be useless in what follows.

Two codes $\mathscr{C}, \mathscr{D} \subseteq \mathbb{F}_{q^m}^n$ are said to be *right equivalent* if there exists $\boldsymbol{P} \in \mathbf{GL}_n(\mathbb{F}_q)$ such that for any $\boldsymbol{c} \in \mathscr{C}$, $\boldsymbol{cP} \in \mathscr{D}$. We denote this as "$\mathscr{C}\boldsymbol{P} = \mathscr{D}$". We emphasize that $\boldsymbol{P}$ should have its entries in $\mathbb{F}_q$ and **not** in $\mathbb{F}_{q^m}$. In this way, the map $\boldsymbol{x} \mapsto \boldsymbol{xP}$ is rank–preserving, *i.e.* is an isometry with respect to the rank metric.

Finally, the *dual* $\mathscr{C}^\perp$ of a code $\mathscr{C} \in \mathbb{F}_{q^m}^n$ is the *orthogonal* of $\mathscr{C}$ with respect to the canonical inner product in $\mathbb{F}_{q^m}$,

$$\begin{cases} \mathbb{F}_{q^m} \times \mathbb{F}_{q^m} \longrightarrow \mathbb{F}_{q^m} \\ (\boldsymbol{x}, \boldsymbol{y}) \longmapsto \sum_{i=1}^n x_i y_i. \end{cases}$$

We frequently apply the *component-wise Frobenius map* to vectors and codes:, given $\boldsymbol{c} = (c_1, \dots, c_n) \in \mathbb{F}_{q^m}^n$ and $0 \leqslant i \leqslant m-1$, we denote

$$\boldsymbol{c}^{[i]} \stackrel{\text{def}}{=} (c_1^{q^i}, \dots, c_n^{q^i}).$$

Given an $[n,k]$ code $\mathscr{C} \subset \mathbb{F}_{q^m}^n$, we write

$$\mathscr{C}^{[i]} \stackrel{\text{def}}{=} \{\boldsymbol{c}^{[i]} \mid \boldsymbol{c} \in \mathscr{C}\}.$$

We also define the (i-*th*) q-*sum* of $\mathscr{C}$ as,

$$\Lambda_i(\mathscr{C}) \stackrel{\text{def}}{=} \mathscr{C} + \mathscr{C}^{[1]} + \cdots + \mathscr{C}^{[i]}.$$

We notice that if $\boldsymbol{G} \in \mathcal{M}_{k,n}(\mathbb{F}_{q^m})$ is a generator matrix of $\mathscr{C}$, the matrix

$$\begin{pmatrix} \boldsymbol{G} \\ \boldsymbol{G}^{[1]} \\ \vdots \\ \boldsymbol{G}^{[i]} \end{pmatrix} \in \mathcal{M}_{(i+1)k,n}(\mathbb{F}_{q^m}) \tag{1}$$

is a generator of the q-sum of $\mathscr{C}$, *i.e.* $\Lambda_i(\mathscr{C})$. By abuse of notation we sometimes denote the matrix of (1) as $\Lambda_i(\boldsymbol{G})$.

1.3 Gabidulin codes

q-*polynomials* were first introduced in [36]. They are defined as $\mathbb{F}_{q^m}$-linear combinations of the monomials $X, X^q, X^{q^2}, \dots, X^{q^i}, \dots$ respectively denoted by $X, X^{[1]}, X^{[2]}, \dots, X^{[i]}, \dots$ Formally, a nonzero q-polynomial F is defined as,

$$F = \sum_{i=0}^{d} f_i X^{[i]}$$

assuming that $f_d \neq 0$. The integer d is called q–*degree* of F and we denote it $\deg_q f$. We equip the space of q–polynomial with a non-commutative algebra structure, where the multiplication law is the composition of polynomials. In particular, the product law is given by the following relations extended by $\mathbb{F}_{q^m}$–linearity:

$$\forall i, j \in \mathbb{N},\ \forall a \in \mathbb{F}_{q^m}, \quad X^{[i]}X^{[j]} = X^{[i+j]} \quad \text{and} \quad X^{[i]}a = a^{q^i}X^{[i]}.$$

Any q–polynomial F induces an $\mathbb{F}_q$–endomorphism $\mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ and the *rank of* F will be defined as the rank of its induced endomorphism.

Denote by $\mathcal{L}$ the ring of all q–polynomial and by $\mathcal{L}^{<e}$ the $\mathbb{F}_{q^m}$–linear space of q–polynomials of q–degree less than e, namely:

$$\mathcal{L}^{<e} \stackrel{\text{def}}{=} \{f \in \mathcal{L} \mid \deg_q f < e\}.$$

Given two positive integers k, n, with $k < n \leqslant m$ and $\boldsymbol{g} \in \mathbb{F}_{q^m}^n$ of $\mathbf{rank}_q(\boldsymbol{g}) = n$, the *Gabidulin code* of length n and dimension k is defined as

$$\mathscr{G}_k(\boldsymbol{g}) \stackrel{\text{def}}{=} \{(F(g_1), \ldots, F(g_n)) \mid F \in \mathcal{L}^{<k}\}.$$

A *generator matrix of this code* is a *Moore matrix* (see for instance [28, § 1.3]), *i.e.* a matrix of the form

$$\mathbf{M}_k(\boldsymbol{g}) \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{g} \\ \boldsymbol{g}^{[1]} \\ \vdots \\ \boldsymbol{g}^{[k-1]} \end{pmatrix} = \begin{pmatrix} g_1 & g_2 & \cdots & g_n \\ g_1^q & g_2^q & \cdots & g_n^q \\ \vdots & \vdots & \cdots & \vdots \\ g_1^{q^{k-1}} & g_2^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{pmatrix}. \tag{2}$$

Gabidulin codes are *Maximum Rank Distance* (MRD) codes, *i.e.* their minimum distance is $d_{\min}(\mathscr{G}_k(\boldsymbol{g})) = n - k + 1$ and they benefit from a decoding algorithm correcting up to half the minimum distance (see [33]).

We now recall the following classical lemmas, that will be useful in the rest of the paper.

Lemma 1. *Let $\mathscr{G}_k(\boldsymbol{g})$ be a Gabidulin code and $\boldsymbol{T} \in \mathbf{GL}_n(\mathbb{F}_q)$. Then $\mathbf{M}_k(\boldsymbol{g})\boldsymbol{T}$ is a generator matrix of the Gabidulin code $\mathscr{G}_k(\boldsymbol{gT})$.*

In short, a right–equivalent code to a Gabidulin code is a Gabidulin code with another evaluation sequence.

Lemma 2 ([18, Theorem 7]). *The dual of the Gabidulin code $\mathscr{G}_k(\boldsymbol{g})$ is the Gabidulin code $\mathscr{G}_{n-k}(\boldsymbol{y}^{[-n+k+1]})$, where $\boldsymbol{y}$ is a nonzero vector in $\mathscr{G}_{n-1}(\boldsymbol{g})^{\perp}$.*

1.4 Twisted Gabidulin codes

Twisted Gabidulin codes were first introduced in [48] and contain a broad family of MRD codes that are not equivalent to Gabidulin codes. The construction of these codes was then generalized in [42,43]. We consider the q–polynomials of the form

$$F = \sum_{i=0}^{k-1} f_i X^{[i]} + \sum_{j=1}^{\ell} \eta_j f_{h_j} X^{[k-1+t_j]}, \tag{3}$$

where the f_i'a are in $\mathbb{F}_{q^m}$, $\ell \leqslant n-k$, $\boldsymbol{h} \in \{0, \ldots, k-1\}^{\ell}$, $\boldsymbol{t} \in \{1, \ldots, n-k\}^{\ell}$ (with distinct t_i) and $\boldsymbol{\eta} \in (\mathbb{F}_{q^m}^*)^{\ell}$. We denote by $\mathcal{L}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}^{n,k}$ the space of all q–polynomials of the form (3) with parameters $\boldsymbol{h}, \boldsymbol{t}, \boldsymbol{\eta}$. Now, given a vector $\boldsymbol{g} \in \mathbb{F}_{q^m}^n$, with $\mathbf{rank}_q(\boldsymbol{g}) = n$, the $[\boldsymbol{g}, \boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta}]$-*twisted Gabidulin code* of *length* n, dimension k, ℓ twists, *hook vector* $\boldsymbol{h}$, *twist vector* $\boldsymbol{t}$ and evaluation sequence $\boldsymbol{g}$ is defined as

$$\mathscr{C}_{\boldsymbol{g},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n, k] \stackrel{\text{def}}{=} \{(F(g_1), \ldots, F(g_n)) \mid F \in \mathcal{L}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}^{n,k}\}.$$

We observe that in [48], Sheekey introduced a simplified version of these codes with just one twist, *i.e.* $n = m, \ell = 1, \boldsymbol{h} = (0), \boldsymbol{t} = (1)$.

Assumption 1. *Throughout this paper, according to [43], we consider a* $[\boldsymbol{g},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}]$*-twisted Gabidulin code with* ℓ *twists, and with the following parameters,*

- $t_i \stackrel{def}{=} (i+1)(\delta+1)$, *where* $\delta \stackrel{def}{=} \frac{n-k-\ell}{\ell+1}$,
- $0 < h_1 < h_2 < \ldots < h_\ell < k-1$ *and* $|h_i - h_{i-1}| > 1$.

for any i, $1 \leqslant i \leqslant \ell$.

This choice is particularly relevant because it allows us to quantify the dimension of the q-sum operator applied to these codes (see Proposition 2).

We now observe that in general, a generator matrix of a $\mathscr{C}_{\boldsymbol{g},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ is

$$\begin{pmatrix} \boldsymbol{g} \\ \boldsymbol{g}^{[1]} \\ \vdots \\ \boldsymbol{g}^{[h_1-1]} \\ \boldsymbol{g}^{[h_1]} + \eta_1 \boldsymbol{g}^{[k-1+t_1]} \\ \boldsymbol{g}^{[h_1+1]} \\ \vdots \\ \boldsymbol{g}^{[h_\ell-1]} \\ \boldsymbol{g}^{[h_\ell]} + \eta_\ell \boldsymbol{g}^{[k-1+t_\ell]} \\ \boldsymbol{g}^{[h_\ell+1]} \\ \vdots \\ \boldsymbol{g}^{[k-1]} \end{pmatrix}. \tag{4}$$

The decoding of twisted Gabidulin codes such as their *additive variants* has recently been studied in [45,44,31,32,29,30]. However, in [45] there were proposed some algorithms which allow to decode twisted Gabidulin codes with only one twist and $\boldsymbol{t} = (1)$, for some special choices of parameters. They manage to correct up to $\lfloor \frac{n-k-1}{2} \rfloor$ errors. But their decoding up to half of the minimum distance remains an open problem.

To the best of our knowledge, the decoding of twisted Gabidulin codes with multiple twists, or one twist with $t_1 > 1$ has not been studied in the literature. We address this point in § 2 for decoding radii that remain below half the minimum distance.

1.5 GPT system and variants

The GPT cryptosystem was introduced in 1991 by Gabidulin, Paramonov and Tretjakov [22]. This system is a *rank-metric* variant of the classical *McEliece* cryptosystem [35], in which the Goppa codes are replaced by Gabidulin codes. The first version of GPT was first broken by Gibson in [26]. Gabidulin proposed a new version in [19], which was later attacked again by Gibson in [27].

In this work we present the generalized version of GPT proposed by Gabidulin and Ourivski in [21,38].

– *Key Generation*. Let,
  - $\mathscr{G}_k(\boldsymbol{g})$ an $[n,k]$-Gabidulin code with generator matrix $\boldsymbol{G}_{\text{sec}}$ (as in (2));
  - $\boldsymbol{S}$ a random invertible matrix in $\mathcal{M}_k(\mathbb{F}_{q^m})$,
  - $\boldsymbol{X}$ a random matrix in $\mathcal{M}_{k,\lambda}(\mathbb{F}_{q^m})$ of fixed rank $1 \leqslant s \leqslant \lambda$, called *distortion matrix*,
  - $\boldsymbol{P}$ a random matrix in $\mathbf{GL}_{n+\lambda}(\mathbb{F}_q)$, called *column scrambler*.

  The *secret key* is the triple,

$$(\boldsymbol{S}, \boldsymbol{G}_{\text{sec}}, \boldsymbol{P})$$

  and the *public key* is,

$$\boldsymbol{G}_{pub} \stackrel{\text{def}}{=} \boldsymbol{S}(\boldsymbol{X} \mid \boldsymbol{G}_{\text{sec}})\boldsymbol{P}, \tag{5}$$

  where $(\boldsymbol{X} \mid \boldsymbol{G}_{\text{sec}}) \in \mathcal{M}_{k,n+\lambda}(\mathbb{F}_{q^m})$ denotes the matrix whose columns are the concatenations of those of $\boldsymbol{X}$ and of $\boldsymbol{G}_{\text{sec}}$. We denote $\mathscr{C}_{\text{pub}}$ the linear code with $\boldsymbol{G}_{\text{pub}}$ as generator matrix.
– *Encryption*. To encode a plaintext $\boldsymbol{m} \in \mathbb{F}_{q^m}^k$, choose a random vector $\boldsymbol{e} \in \mathbb{F}_{q^m}^{n+\lambda}$ of $\mathbf{rank}_q(\boldsymbol{e}) = t$, where $t = \lfloor \frac{n-k}{2} \rfloor$ and compute the ciphertext as,

$$\boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{m}\boldsymbol{G}_{\text{pub}} + \boldsymbol{e}.$$

– *Decryption*. Apply the chosen decoding algorithm for Gabidulin codes to the last n components of the vector,

$$\boldsymbol{c}\boldsymbol{P}^{-1} = \boldsymbol{m}\boldsymbol{S}[\boldsymbol{X}|\boldsymbol{G}_{\text{sec}}] + \boldsymbol{e}\boldsymbol{P}^{-1}.$$

  Since $\boldsymbol{P} \in \mathbf{GL}_{n+\lambda}(\mathbb{F}_q)$, then $\mathbf{rank}_q(\boldsymbol{e}\boldsymbol{P}^{-1}) = t$ and in particular, the rank (over $\mathbb{F}_q$) of the last n rows of this matrix is at most t. So, the decoder computes $\boldsymbol{m}\boldsymbol{S}$, and by inverting $\boldsymbol{S}$, the initial message can be finally encrypted.

The description of the secret key as the triple $(\boldsymbol{S}, \boldsymbol{G}_{\text{sec}}, \boldsymbol{P})$ is not the most relevant one when it comes to instantiating the scheme with Gabidulin or twisted Gabidulin codes. In particular, once we know the secret code $\mathscr{C}_{\text{sec}}$ of the generator matrix $\boldsymbol{G}_{\text{sec}}$ and the scrambling matrix, we are able to decode. So, the knowledge of $\boldsymbol{S}$ is not relevant. Thus, in the following, we assume that $\boldsymbol{G}_{\text{pub}}$ as

$$\boldsymbol{G}_{\text{pub}} = (\boldsymbol{X} \mid \boldsymbol{G}_{\text{sec}})\boldsymbol{P}. \tag{6}$$

Remark 2. The previous scheme is instantiated with Gabidulin codes but can actually be instantiated with any code family equipped with a decoder that corrects up to t errors.

Remark 3. The original GPT scheme [22] did not involve the distortion matrix $\boldsymbol{X}$ as it is. The seminal proposal was to use either a random generator matrix $\boldsymbol{G}$ of a Gabidulin code or a matrix $\boldsymbol{G} + \boldsymbol{X}_0$, where $\boldsymbol{X}_0$ had low rank. The latter version required to reduce the weight of the error term in the encryption process. In the following, we no longer consider this masking technique. The use of a distortion matrix with a column scrambler appeared only ten years later with the works of Ourivski and Gabidulin [21,38].

2 On the decoding of Gabidulin codes and their twists

In this section, we discuss further the decoding of Gabidulin and twisted Gabidulin codes. We show that, although decoding twisted Gabidulin codes up to half the minimum distance remains an open problem, their decoding up to a smaller radius is possible, using the same decoder as for Gabidulin codes. This approach was developed in [8] and is related to that of Gaborit, Ruatta and Schrek in [24, § V–VI].

We begin by examining the decoding of Gabidulin codes.

2.1 An important remark on the decoder of Gabidulin codes

It is well–known that the Gabidulin codes have a decoder that corrects up to half the minimum distance (see for instance [33]). This algorithm is analogous to the Welch–Berlekamp algorithm for Reed–Solomon codes. An important fact from a cryptographic point of view is that, given a Reed–Solomon code

$$\mathbf{RS}(k) \stackrel{\text{def}}{=} \{(f(x_1), \dots, f(x_n)) \mid f \in \mathbb{F}_q[X],\ \deg f < k\},$$

where $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{F}_q^n$ has distinct entries, the knowledge of the vector $\boldsymbol{x}$ is necessary to run the decoding algorithm. However, given a Gabidulin code $\mathscr{G}_k(\boldsymbol{g})$, it is possible to decode without knowing $\boldsymbol{g}$. Indeed, given as input $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ where $\boldsymbol{c} \in \mathscr{G}_k(\boldsymbol{g})$ and $\mathbf{rank}_q(\boldsymbol{e}) \leqslant t \stackrel{\text{def}}{=} \frac{n-k}{2}$, the decoding algorithm first consists in finding a q–polynomial $P(x)$ of degree at most t which vanishes at the entries of $\boldsymbol{e}$. This can be done by solving the $\mathbb{F}_{q^m}$–linear system

$$P(\boldsymbol{y}) \stackrel{\text{def}}{=} (P(y_1), \dots, P(y_n)) \in \mathscr{G}_{k+t}(\boldsymbol{g}) \tag{7}$$

whose unknowns are the coefficients of $P \in \mathcal{L}^{\leqslant t}$. Next, the code $\mathscr{G}_{k+t}(\boldsymbol{g})$ can be computed by simply knowing a generator matrix of $\mathscr{G}_k(\boldsymbol{g})$, thanks to the following well–known statement.

Proposition 1 ([40, Lem. 5.1]). *Let $\boldsymbol{g} \in \mathbb{F}_{q^m}^n$, with $\mathbf{rank}_q(\boldsymbol{g}) = n$ and $\mathscr{G}_k(\boldsymbol{g})$ an $[n, k]$ Gabidulin code. Then,*

$$\Lambda_i(\mathscr{G}_k(\boldsymbol{g})) = \mathscr{G}_{k+i}(\boldsymbol{g}).$$

In particular,

$$\dim(\Lambda_i(\mathscr{G}_k(\boldsymbol{g}))) = \min\{n, k+i\}.$$

Next, for any P satisfying (7), we have $P(\boldsymbol{y}) = P(\boldsymbol{c}) + P(\boldsymbol{e})$. By construction, $P(\boldsymbol{c}) \in \Lambda_t(\mathscr{G}_k(\boldsymbol{g})) = \mathscr{G}_{k+t}(\boldsymbol{g})$ and hence, $P(\boldsymbol{e}) \in \mathscr{G}_{k+t}(\boldsymbol{g})$. Moreover, we have $\mathbf{rank}_q(P(\boldsymbol{e})) \leqslant \mathbf{rank}_q(\boldsymbol{e}) \leqslant t$, while $\Lambda_t(\mathscr{G}_k(\boldsymbol{g})) = \mathscr{G}_{k+t}(\boldsymbol{g})$ has minimum distance $n-k-t+1$. Therefore, for $t \leqslant \frac{n-k}{2}$, which entails $t < n-k-t+1$, we should have $P(\boldsymbol{e}) = 0$ for any P satisfying (7). Thus, the kernel of P contains the support of $\boldsymbol{e}$ and the knowledge of the support of the error allows to solve the decoding problem by solving a linear system. See for instance [24, § IV.a], [4, § III.A].

Algorithm 1: Decoding algorithm of Gabidulin codes without knowing the evaluation sequence

Input: A Gabidulin code $\mathscr{C}$ represented by a generator matrix $\boldsymbol{G}$, an integer t and a vector $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$

Output: A vector $\boldsymbol{c} \in \mathscr{C}$ such that $\mathbf{rank}_q(\boldsymbol{y} - \boldsymbol{c}) \leqslant t$ if exists and '?' otherwise.

Compute $P \in \mathcal{L}^{\leqslant t} \setminus \{0\}$ such that $P(\boldsymbol{y}) \in \Lambda_t(\mathscr{C})$
Compute (if exists) $\boldsymbol{e} \in \mathbb{F}_{q^m}$ such that $\mathrm{Supp}(\boldsymbol{e}) \subseteq \mathrm{Ker}(P)$ and $\boldsymbol{y} - \boldsymbol{e} \in \mathscr{C}$
if $\boldsymbol{e}$ *exists* **then**
  Return $\boldsymbol{y} - \boldsymbol{e}$
else
  Return '?'

Algorithm 1 summarizes the previous discussion. Note that, with the knowledge of the evaluation sequence $\boldsymbol{g}$, the algorithm could be terminated by performing an Euclidean division or using the Extended Euclidean Algorithm in the non-commutative ring $\mathcal{L}$ instead of using [24, § IV.a], [4, § III.A].

The key observation here is the following : **decoding a Gabidulin code $\mathscr{G}_k(\boldsymbol{g})$ is possible without knowing the vector $\boldsymbol{g}$.**

Remark 4. In GPT original public key encryption scheme [22] the public code is a Gabidulin code with no distortion matrix. In this situation, the previous discussion shows that this proposal is immediately broken without trying to compute a description (*i.e.* an evaluation sequence) of the public code.

2.2 Decoding twisted Gabidulin codes

If some twisted Gabidulin codes are proven to be MRD without being equivalent to Gabidulin codes, the question of decoding them up to half the minimum distance remains open. For *twisted Reed–Solomon codes*, the Hamming metric analogues introduced in [7], it is shown in [6] how they can be decoded up to half the minimum distance at the cost of an exhaustive search on the terms associated with the twists. Thus, the decoding complexity of a twisted Reed–Solomon code with ℓ twists is $O(q^\ell)$ times the complexity of the decoding of a Reed–Solomon code. This can be transposed to twisted Gabidulin codes but the cost overhead is $O(q^{m\ell})$ times the cost of decoding a Gabidulin code, which is exponential in m and so in the code length n (since $n \leqslant m$).

Although one does not know how to efficiently decode twisted Gabidulin codes up to half the minimum distance, one can apply the Algorithm 1 to them. Given $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$, where $\boldsymbol{c}$ is a codeword of a twisted Gabidulin code $\mathscr{C}$ and $\mathbf{rank}_q(\boldsymbol{e}) \leqslant t$ for some t we will discuss later, compute $P \in \mathcal{L}^{\leqslant t}$ such that

$$P(\boldsymbol{y}) \stackrel{\text{def}}{=} (P(y_1), \ldots, P(y_n)) \in \Lambda_t(\mathscr{C}). \tag{8}$$

Such a solution P satisfies $P(\boldsymbol{e}) \in \Lambda_t(\mathscr{C})$. The difference with the Gabidulin case is that we do not have an *a priori* lower bound on the minimum distance of $\Lambda_t(\mathscr{C})$. However we have the following result.

Proposition 2 ([43, Theorem 4]). *Given a twisted Gabidulin code* $\mathscr{C}_{\boldsymbol{g},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ *(where parameters are chosen according to Assumption 1), then*

$$\forall i \geqslant 0, \quad \dim(\Lambda_i(\mathscr{C}_{\boldsymbol{g},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k])) = \min\{k+i+\ell(i+1), n\}.$$

Proposition 2 entails that for a twisted Gabidulin code $\mathscr{C}$ with ℓ twists, we have

$$\dim_{\mathbb{F}_{q^m}} \Lambda_t(\mathscr{C}) \leqslant k-1+(t+1)(\ell+1). \tag{9}$$

Now, let us consider the dimension of $\Lambda_t(\mathscr{E})$. Since $\Lambda_t(\mathscr{E})$ is the image of $\mathcal{L}^{\leqslant t}$ by the map $Q \mapsto Q(\boldsymbol{e})$, we have

$$\dim_{\mathbb{F}_{q^m}}(\Lambda_t(\mathscr{E})) = \dim_{\mathbb{F}_{q^m}}(\mathcal{L}^{\leqslant t}) - \dim_{\mathbb{F}_{q^m}}\{Q \in \mathcal{L}^{\leqslant t} \mid Q(\boldsymbol{e}) = 0\}.$$

First, $\dim(\mathcal{L}^{\leqslant t}) = t+1$. Second, recall that there exists a unique monic q–polynomial P of q–degree $\mathbf{rank}_q(\boldsymbol{e})$ such that $P(\boldsymbol{e}) = 0$. Therefore,

$$\{Q \in \mathcal{L}^{\leqslant t} \mid Q(\boldsymbol{e}) = 0\} = \{F \circ P \mid F \in \mathcal{L}^{\leqslant t-\mathbf{rank}_q(\boldsymbol{e})}\}$$

and the latter space has dimension $t - \mathbf{rank}_q(\boldsymbol{e}) + 1 \geqslant 1$. Putting all together, we deduce that

$$\dim_{\mathbb{F}_{q^m}}(\Lambda_t(\mathscr{E})) \leqslant t.$$

We claim that if

$$\dim_{\mathbb{F}_{q^m}} \Lambda_t(\mathscr{C}) + t \leqslant n, \tag{10}$$

the spaces $\Lambda_t(\mathscr{C})$ and $\Lambda_t(\mathscr{E})$ are very likely to have a zero intersection. The validity of this claim are given in § 2.3. This would entail that for any $P \in \mathcal{L}^{\leqslant t}$ satisfying (8), we have $P(\boldsymbol{e}) = 0$. Therefore, from (9) and (10) we can conclude that if,

$$t \leqslant \frac{n-k-\ell}{\ell+2}\cdot$$

then we can decode twisted Gabidulin codes as classical Gabidulin codes : form the kernel of P, we get the error support and finally the error itself is deduced using [24, § IV.a], [4, § III.A]. This decoding radius is rather pessimistic since the dimension of $\Lambda_t(\mathscr{C})$ may be much smaller depending on the way the twists are chosen. Therefore, the above bound is what we can expect in the worst case.

2.3 Discussion about the claim

Suppose that the error $\boldsymbol{e}$ is obtained as follows: draw a uniformly random subspace $V \subseteq \mathbb{F}_q^n$ of dimension t and then draw a uniformly random vector $\boldsymbol{e}$ among the vector with row support contained in V. One can easily prove that all the elements of $\Lambda_t(\boldsymbol{e})$ have their row support contained in V.

Therefore, the intersection $\Lambda_t(\mathscr{E}) \cap \Lambda_t(\mathscr{C})$ consists in elements of $\Lambda_t(\mathscr{C})$ whose row support is in V. So, consider the subcode $\mathrm{Sh}_V(\Lambda_t(\mathscr{C}))$ called *shortening of* $\Lambda_t(\mathscr{C})$ defined as the subcode of $\Lambda_t(\mathscr{C})$ of vectors whose row support is contained in V. This space can be obtained as follows. Consider a basis $(\boldsymbol{v}_1, \dots, \boldsymbol{v}_{n-t})$ of the dual $V^\perp \subseteq \mathbb{F}_q^n$ of V for the canonical inner product. Then, $\mathrm{Sh}_V(\Lambda_t(\mathscr{C}))$ is the kernel of the map

$$\begin{cases} \Lambda_t(\mathscr{C}) \longrightarrow & \mathbb{F}_{q^m}^{n-t} \\ \quad \boldsymbol{c} \quad \longmapsto & (\boldsymbol{c} \cdot \boldsymbol{v}_1^\top, \dots, \boldsymbol{c} \cdot \boldsymbol{v}_{n-t}^\top). \end{cases}$$

Remark 5. Note that in the above equation, $\boldsymbol{c}$ and the $\boldsymbol{v}_i$'s have different nature, $\boldsymbol{c}$ has entries in $\mathbb{F}_{q^m}$ while the $\boldsymbol{v}_i$'s have their entries in $\mathbb{F}_q$.

Finally, since V is uniformly random, and $\dim \Lambda_t(\mathscr{C}) \leqslant n-t$, it is likely that the above map is injective and hence its kernel $\mathrm{Sh}_V(\Lambda_t(\mathscr{C}))$ is likely to be zero. Since the latter kernel contains $\Lambda_t(\mathscr{E}) \cap \lambda_t(\mathscr{C})$, we conclude that this intersection is likely to be zero.

2.4 A remark on the code that is actually decoded

To conclude, let us notice an important fact for the sections to follow. The previously described decoder may decode a slightly larger code than $\mathscr{C}$ defined below.

Definition 2. *Let $\mathscr{C} \subseteq \mathbb{F}_{q^m}^n$ be a code and s be a positive integer. We denote by $\overline{\mathscr{C}}^s$ the largest code $\mathscr{C}'$ containing $\mathscr{C}$ such that $\Lambda_s(\mathscr{C}) = \Lambda_s(\mathscr{C}')$.*

It is easy to check that, the aforementioned decoder actually decodes $\overline{\mathscr{C}}^t$ and not only $\mathscr{C}$.

Remark 6. It can be proved that for a random code $\mathscr{C}$ with dimension $k < \frac{n}{s}$, then $\mathscr{C} = \overline{\mathscr{C}}^s$ with a high probability. It ca also be proved that a Gabidulin code $\mathscr{C}$ of dimension k satisfies $\overline{\mathscr{C}}^i = \mathscr{C}$ for any $i < n-k$.

Remark 7. An alternative definition of $\overline{\mathscr{C}}^s$ is given by.

$$\overline{\mathscr{C}}^s \stackrel{\text{def}}{=} \bigcap_{j=0}^{s} \left(\Lambda_s(\mathscr{C})\right)^{[-j]}$$

3 Revisiting Overbeck's attack

In this section we revisit the Overbeck's attack of GPT instantiated with Gabidulin codes to introduce the extension presented in § 4, which will allow us to break [43].

3.1 A distinguisher

The core of the Overbeck's attack consists in the application of the q-sum operator, which allows to *distinguish* Gabidulin codes from random ones. In particular, the following proposition observes the behavior of random codes *w.r.t.* the i-th q-sum operator.

Proposition 3 ([10, Prop. 1]). *If $\mathscr{C} \subset \mathbb{F}_{q^m}^n$ is a k-dimensional random code, then for any $0 < i < k$,*

$$\dim(\Lambda_i(\mathscr{C})) \leqslant \min\{n, (i+1)k\}.$$

Moreover, for any $a \geqslant 0$, we have

$$\mathbf{Prob}(\dim(\Lambda_i(\mathscr{C})) \leqslant \min\{n, (i+1)k\} - a) = O(q^{-ma}).$$

Gabidulin codes have a significantly different behavior with respect to the q–sum compared to random codes (see Proposition 1). In fact, we observe that if $i < n - k$,

$$\dim(\Lambda_i(\mathscr{G}_k(\boldsymbol{g}))) = k + i < (i+1)k = \dim(\Lambda_i(\mathscr{C})),$$

where $\mathscr{G}_k(\boldsymbol{g})$ is a n-Gabidulin code of dimension k, and $\mathscr{C}$ is a random code, and we know from the previous proposition that the last equality is true with high probability.

In the Overbeck's attack, the operator $\Lambda_i(\cdot)$ is used for two related reasons.

1. It provides a distinguisher on the public key based on the peculiar behavior of Gabidulin codes with respect to $\Lambda_i(\cdot)$. This permits to rule out the distortion matrix [40] and to recover a decomposition of the form (6), in order to decrypt any ciphertext computed with this public key.
2. Once we have discarded the distortion matrix, we have access to the secret Gabidulin code and we can recover its hidden structure, *i.e.* an evaluation sequence.

We observe that the second step is not necessary since, using Algorithm 1, one can directly decode any message, without knowing the evaluation sequence. Thus, in the sequel, we focus on the first step.

3.2 The structure of $\boldsymbol{\Lambda_i(G_{\mathrm{pub}})}$

Let i be a positive integer and $\boldsymbol{G}_{\mathrm{pub}} = (\boldsymbol{X} \mid \boldsymbol{G}_{\mathrm{sec}})\boldsymbol{P}$ a public key as in (6). Recall that, in the present section, we suppose that $\boldsymbol{G}_{\mathrm{sec}}$ is a generator matrix of a Gabidulin code. Observe that, since $\boldsymbol{P} \in \mathbf{GL}_{n+\lambda}(\mathbb{F}_q)$, we have $\boldsymbol{P}^{[i]} = \boldsymbol{P}$ and hence,

$$\Lambda_i(\boldsymbol{G}_{\mathrm{pub}}) = (\Lambda_i(\boldsymbol{X}) \mid \Lambda_i(\boldsymbol{G}_{\mathrm{sec}}))\boldsymbol{P}. \tag{11}$$

We now assume that $i < n-k$ and we focus on the matrix $(\Lambda_i(\boldsymbol{X}) \mid \Lambda_i(\boldsymbol{G}_{\mathrm{sec}}))$. If we denote the distortion matrix $\boldsymbol{X}$ according to its rows, *i.e.*

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_{k-1} \end{pmatrix},$$

where $\boldsymbol{x}_j \in \mathbb{F}_{q^m}^{\lambda}$ for any $0 \leqslant j \leqslant k-1$, then

$$(\Lambda_i(\boldsymbol{X}) \mid \Lambda_i(\boldsymbol{G}_{\mathrm{sec}})) = \left(\begin{array}{c|c} \boldsymbol{x}_0 & \boldsymbol{g} \\ \boldsymbol{x}_1 & \boldsymbol{g}^{[1]} \\ \vdots & \vdots \\ \boldsymbol{x}_{k-1} & \boldsymbol{g}^{[k-1]} \\ \hline \vdots & \vdots \\ \hline \boldsymbol{x}_0^{[i]} & \boldsymbol{g}^{[i]} \\ \boldsymbol{x}_1^{[i]} & \boldsymbol{g}^{[i+1]} \\ \vdots & \vdots \\ \boldsymbol{x}_{k-1}^{[i]} & \boldsymbol{g}^{[k-1+i]} \end{array}\right).$$

Now, after performing some row elimination, we finally get

$$\left(\begin{array}{c|c} \boldsymbol{x}_0 & \boldsymbol{g} \\ \boldsymbol{x}_1 & \boldsymbol{g}^{[1]} \\ \vdots & \vdots \\ \boldsymbol{x}_{k-1} & \boldsymbol{g}^{[k-1]} \\ \vdots & \vdots \\ \boldsymbol{x}_{k-1}^{[i]} & \boldsymbol{g}^{[k-1+i]} \\ \hline \vdots & \vdots \\ \hline \boldsymbol{x}_0^{[i]} - \boldsymbol{x}_1^{[i-1]} & \boldsymbol{0} \\ \boldsymbol{x}_1^{[i]} - \boldsymbol{x}_2^{[i-1]} & \boldsymbol{0} \\ \vdots & \vdots \\ \boldsymbol{x}_{k-2}^{[i]} - \boldsymbol{x}_{k-1}^{[i-1]} & \boldsymbol{0} \end{array}\right).$$

Thus, we have the following.

Lemma 3. *Let $i < n-k$. Then, up to row elimination,*

$$(\Lambda_i(\boldsymbol{X}) \mid \Lambda_i(\boldsymbol{G}_{sec})) = \begin{pmatrix} \boldsymbol{X}' & \mathbf{M}_{k+i}(\boldsymbol{g}) \\ \Lambda_{i-1}(\boldsymbol{X}'') & \boldsymbol{0} \end{pmatrix}, \tag{12}$$

where,

$$\boldsymbol{X}' = \begin{pmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_{k-1} \\ \boldsymbol{x}_{k-1}^{[1]} \\ \vdots \\ \boldsymbol{x}_{k-1}^{[i]} \end{pmatrix} \quad \textit{and} \quad \boldsymbol{X}'' = \boldsymbol{X}^{[1]}_{\{0,\ldots,k-2\}} - \boldsymbol{X}_{\{1,\ldots,k-1\}}.$$

In detail, $\boldsymbol{X}^{[1]}_{\{0,\ldots,k-2\}}$ is the submatrix of $\boldsymbol{X}^{[1]}$ composed by its first $k-1$ rows and $\boldsymbol{X}_{\{1,\ldots,k-1\}}$ is the submatrix of $\boldsymbol{X}$ composed by its rows starting from the second one.

We now observe that the row space of $\boldsymbol{X}''$, denoted $\mathbf{RowSp}_{\mathbb{F}_{q^m}}(\boldsymbol{X}'')$, is contained in the sum of the row spaces of $\boldsymbol{X}$ and $\boldsymbol{X}^{[1]}$, which is $\mathbf{RowSp}_{\mathbb{F}_{q^m}}(\Lambda_1(\boldsymbol{X}))$ and so $\mathbf{rank}(\boldsymbol{X}'') \leqslant \min\{2s, \lambda\}$, where we recall that $s = \mathbf{rank}(\boldsymbol{X})$.

More generally, $\mathbf{RowSp}_{\mathbb{F}_{q^m}}(\Lambda_{i-1}(\boldsymbol{X}'')) \subseteq \mathbf{RowSp}_{\mathbb{F}_{q^m}}(\Lambda_i(\boldsymbol{X}))$ for any $i \geqslant 1$. And $\mathbf{rank}(\Lambda_{i-1}(\boldsymbol{X}'')) \leqslant \min\{(i+1)s, \lambda\}$.

3.3 Overbeck's attack

The attack consists in finding an $i < n-k$, for which $\mathbf{rank}(\Lambda_{i-1}(\boldsymbol{X}'')) = \lambda$. In this case,

$$\dim(\Lambda_i(\mathscr{G}_{pub})) = k + i + \lambda$$

and the dimension of the dual is

$$\dim(\Lambda_i(\mathscr{G}_{pub})^{\perp}) = n - k - i.$$

So, the code $\Lambda_i(\mathscr{G}_{pub})$ admits a parity check of this form

$$(\mathbf{0} \mid \boldsymbol{H}_i)(\boldsymbol{P}^{-1})^{\top}, \tag{13}$$

where $\boldsymbol{H}_i$ is a parity check matrix of $\Lambda_i(\mathscr{G}_k(\boldsymbol{g})) = \mathscr{G}_{k+i}(\boldsymbol{g})$.

After finding such an i, we can easily find a *valid* column scrambler $\boldsymbol{T} \in \mathbf{GL}_{n+\lambda}(\mathbb{F}_q)$,which will allow us to attack the system (see Theorem 2 ([40, Thm 5.3])).

Therefore, **the crucial part of the Overbeck's attack consists in finding (if there exists) a positive integer i, for which $\dim(\Lambda_{i-1}(\boldsymbol{X}'')) = \lambda$ and $\Lambda_i(\mathscr{C}_{\text{sec}}) \neq \mathbb{F}_{q^m}^n$ or equivalently $\dim(\Lambda_i(\mathscr{C}_{\text{pub}})) = \dim(\Lambda_i(\mathscr{C}_{\text{sec}})) + \lambda$.**

Remark 8. If for $i = n-k-1$, we have $\dim(\Lambda_{n-k-1}(\mathscr{C}_{pub}))^{\perp} = 1$, then we can perform the attack quite straightforwardly. Indeed, in this case there exists $\boldsymbol{v} \in \mathbb{F}_{q^m}^n$ which spans the entire dual. Many papers in the literature describe the attack just for this choice i, claiming that we can perform it only if $\dim(\Lambda_{n-k-1}(\mathscr{C}_{pub}))^{\perp} = 1$. We stress out that **this is not the only possible choice for i:** one only needs an $i < n-k$ for which $\Lambda_i(\mathscr{C}_{pub})^{\perp}$ has the structure (13).

Description of the attack. We now briefly detail the procedure of the attack (partially presented in the proof of [40, Thm. 5.3]).

We know that $\Lambda_i(\mathscr{C}_{pub})$ admits a parity check matrix $\boldsymbol{H}_{\mathrm{pub}}$ (for simplicity, we omit the dependency on i) of the form (13). Thus, we look for some $\boldsymbol{T} \in \mathbf{GL}_{n+\lambda}(\mathbb{F}_q)$ for which

$$\boldsymbol{H}_{\mathrm{pub}}\boldsymbol{T}^\top = (\mathbf{0} \mid \boldsymbol{H}') \tag{14}$$

The matrix $\boldsymbol{T}$ is not unique. Furthermore, the following statement taken from [40, Thm 5.3] asserts that every invertible $\boldsymbol{T}$ satisfying (14) is suitable to complete the attack. For the sake of completeness, we give the proof of this result.

Theorem 2 ([40, Thm 5.3]). *If there exists a positive integer $i < n-k$ for which the dimension of $\Lambda_i(\mathscr{G}_{pub})^\perp$ is $n-k-i$ and if we denote by $\boldsymbol{H}_{pub}$ a generator matrix of this dual, then any $\boldsymbol{T} \in \mathbf{GL}_{n+\lambda}(\mathbb{F}_q)$ such that*

$$\boldsymbol{H}_{pub}\boldsymbol{T}^\top = (\mathbf{0} \mid \boldsymbol{H}')$$

for some $\boldsymbol{H}' \in \mathcal{M}_{n-k-i,n}(\mathbb{F}_{q^m})$ is a valid column scrambler, i.e. *there exists $\boldsymbol{Z} \in \mathcal{M}_{k,\lambda}(\mathbb{F}_{q^m})$ and $\boldsymbol{g}^\star \in \mathbb{F}_{q^m}^n$ of rank n, such that*

$$\boldsymbol{G}_{pub} = \boldsymbol{S}(\boldsymbol{Z} \mid \mathbf{M}_k(\boldsymbol{g}^\star))\boldsymbol{T},$$

where $\mathbf{M}_k(\boldsymbol{g}^\star)$ denotes the Moore matrix with generator vector $\boldsymbol{g}^\star$ (see (2)*).*

Proof. Since $\dim(\Lambda_i(\mathscr{G}_{pub})^\perp) = n-k-i$, then this dual admits a generator matrix of the form (13). Now, consider $\boldsymbol{T} \in \mathbf{GL}_{n+\lambda}(\mathbb{F}_q)$ such that

$$(\mathbf{0} \mid \boldsymbol{H}_i)(\boldsymbol{P}^{-1})^\top\boldsymbol{T}^\top = (\mathbf{0} \mid \boldsymbol{H}') \tag{15}$$

for some $\boldsymbol{H}' \in \mathcal{M}_{n-k-i,n}(\mathbb{F}_{q^m})$. Denote,

$$\boldsymbol{T}\boldsymbol{P}^{-1} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}$$

where $\boldsymbol{A} \in \mathcal{M}_\lambda(\mathbb{F}_q)$, $\boldsymbol{B} \in \mathcal{M}_{\lambda,n}(\mathbb{F}_q)$, $\boldsymbol{C} \in \mathcal{M}_{n,\lambda}(\mathbb{F}_q)$ and $\boldsymbol{D} \in \mathcal{M}_n(\mathbb{F}_q)$. From (15),we have that

$$\boldsymbol{H}_i\boldsymbol{B}^\top = 0 \Longrightarrow \boldsymbol{B} = 0.$$

Since $\boldsymbol{P}\boldsymbol{T}^{-1}$ is invertible, this entails in particular that $\boldsymbol{A} \in \mathbf{GL}_\lambda(\mathbb{F}_q)$ and $\boldsymbol{D} \in \mathbf{GL}_n(\mathbb{F}_q)$. Then, we have that

$$(\boldsymbol{T}\boldsymbol{P}^{-1})^{-1} = \boldsymbol{P}\boldsymbol{T}^{-1} = \begin{pmatrix} \boldsymbol{A}^{-1} & \mathbf{0} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{D}^{-1} \end{pmatrix}$$

and so we get,

$$\boldsymbol{G}_{\mathrm{pub}}\boldsymbol{T}^{-1} = \boldsymbol{S}(\boldsymbol{X} \mid \mathbf{M}_k(\boldsymbol{g}))\boldsymbol{P}\boldsymbol{T}^{-1} = \boldsymbol{S}(\boldsymbol{Z} \mid \boldsymbol{G}')$$

for some matrix $\boldsymbol{Z}$, where $\boldsymbol{G}'$ is a generator matrix of $\mathscr{G}_k(\boldsymbol{g})\boldsymbol{D}^{-1}$, which also equals $\mathscr{G}_k(\boldsymbol{g}\boldsymbol{D}^{-1})$ since $\boldsymbol{D}$ is nonsingular with entries in $\mathbb{F}_q$ (see Lemma 1). $\square$

In order find such a $\boldsymbol{T}$, we compute the space of the matrices $\boldsymbol{T} \in \mathcal{M}_{n+\lambda}(\mathbb{F}_q)$ such that the λ leftmost columns of $\boldsymbol{H}_{\text{pub}}\boldsymbol{T}^\top$ are zero. Then , we need to extract a nonsingular matrix from this solution space. This last step can be done by picking random elements in this space until we find a nonsingular matrix.

Once such a column scrambler $\boldsymbol{T}$ is computed, we can compute $\boldsymbol{c}\boldsymbol{T}^{-1}$ and remove the leftmost λ entries. By Theorem 2, each of these $\boldsymbol{T}$'s is a valid column scrambler and it suffices to apply the Gabidulin codes decoder to the former vector to recover the plaintext. Recall that, from § 2.1, the decoder works independently on the knowledge of $\boldsymbol{g}$.

3.4 Analyzing the dimension of $\Lambda_i(\mathscr{C}_{\text{pub}})$ for small i's.

In this section we study what happens if we apply the q-sum operator to the public key for small i's, namely $i = 1$. In particular, we will see that in this case we can always attack the system by applying either strategies described in § 4 or the classical Overbeck attack.

First, we recall that, by Lemma 3, the matrix $(\Lambda_1(\boldsymbol{X}) \mid \Lambda_1(\boldsymbol{G}))$ (see (12)) can be transformed into a matrix

$$\begin{pmatrix} \boldsymbol{X}' & \mathbf{M}_{k+1}(\boldsymbol{g}) \\ \boldsymbol{X}'' & \mathbf{0} \end{pmatrix} \tag{16}$$

In this case, $\mathbf{rank}(\boldsymbol{X}'') \leqslant \min\{2s, \lambda\}$, where $s = \mathbf{rank}(\boldsymbol{X})$. We now introduce the following useful lemma.

Lemma 4. *If $k \geqslant 4s + 1$, then, up to row multiplications,*

$$[\Lambda_1(\boldsymbol{X}) \mid \Lambda_1(\boldsymbol{G})] = \begin{pmatrix} \mathbf{0} & \mathbf{M}_{k+1}(\boldsymbol{g}) \\ \boldsymbol{X}'' & \mathbf{0} \end{pmatrix} \tag{17}$$

with a high probability.

Proof. We need to prove that $\mathbf{RowSp}_{\mathbb{F}_{q^m}}(\boldsymbol{X}') \subseteq \mathbf{RowSp}_{\mathbb{F}_{q^m}}(\boldsymbol{X}'')$. We first claim that $\mathbf{RowSp}_{\mathbb{F}_{q^m}}(\boldsymbol{X}'') = \mathbf{RowSp}_{\mathbb{F}_{q^m}}(\Lambda_1(\boldsymbol{X}))$ with a high probability. We consider the submatrix of $\boldsymbol{X}''$ in $\mathcal{M}_{\lfloor \frac{k-1}{2} \rfloor,\lambda}(\mathbb{F}_{q^m})$ obtained by selecting alternate rows of $\boldsymbol{X}''$. This is a uniformly random matrix in $\mathcal{M}_{\lfloor \frac{k-1}{2} \rfloor,\lambda}(\mathbb{F}_{q^m})$. By the assumption $\frac{k-1}{2} \geqslant 2s$, it has rank equal to $\min\{2s, \lambda\} = \mathbf{rank}(\Lambda_1(\boldsymbol{X}))$ with a high probability (by Proposition 3). Thus, $\mathbf{rank}(\boldsymbol{X}'') \geqslant \mathbf{rank}(\Lambda_1(\boldsymbol{X}))$ with a high probability and so the claim follows. The result derives from remarking that $\mathbf{RowSp}_{\mathbb{F}_{q^m}}(\boldsymbol{X}') \subseteq \mathbf{RowSp}_{\mathbb{F}_{q^m}}(\Lambda_1(\boldsymbol{X}))$. □

We remark that if $\mathbf{rank}(\boldsymbol{X}) = s \geqslant \lambda/2$, then $\mathbf{rank}(\boldsymbol{X}'') = \lambda$ with high probability and so we can apply straightforwardly the Overbeck's attack (§ 3). One could then think that it suffices to take a sufficiently small s in order to repair the system. In the following section we show that thanks to the structure of the matrix (17), we can construct an attack, which is an extension of the Overbeck's one, which allows us to break the system independently from the rank of the distortion matrix, even for the twisted Gabidulin GPT scheme.

Remark 9. The condition $k \geqslant 4s+1$ required in Lemma 4, yields a range of parameters for which we can assert the validity of the result. Nevertheless, it is probably highly conservative and one could expect result to hold for smaller k or equivalently larger s.

3.5 Puchinger, Renner and Wachter–Zeh variant of GPT

In [43], the authors use simultaneously two distinct techniques in order to resist to Overbeck's attack:

1. they impose the distortion matrix to have a low rank (*e.g.* $s = 1$ or 2),
2. they replace Gabidulin codes by twisted ones (with parameters specified in Assumption 1).

The rationale behind the use of twisted Gabidulin codes is that, one step of Overbeck's attack consists in obtaining $\Lambda_{n-k-1}(\mathscr{C}_{\mathrm{sec}})$ where $\mathscr{C}_{\mathrm{sec}}$ is the hidden Gabidulin code. Then the dual $\Lambda_{n-k-1}(\mathscr{C})$ has dimension 1 and immediately provides the evaluation sequence. Based on this observation, the authors select parameters for twisted Gabidulin codes such that none of the $\Lambda_i(\mathscr{C}_{\mathrm{sec}})$'s for $i > 0$ may have codimension 1 (see [43, Thm. 6]).

Table 1. Parameters from [43]

q	k	n	m	ℓ	λ	s
2	18	26	104	2	6	1
2	21	33	132	2	8	1
2	32	48	192	2	12	2

As mentioned in Remark 8, the choice of computing $\Lambda_{n-k-1}(\mathscr{C}_{\mathrm{sec}})$ is only technical and can be circumvented in many different ways. In fact, once the distortion matrix $\boldsymbol{X}$ is discarded, we can access to $\mathscr{C}_{\mathrm{sec}}$ and, using the discussion in § 2.2, just knowing this code is generally enough to decode. However, their approach presents another difficulty for the attacker if one wants to apply Overbeck's attack. Indeed, the proposed parameters consider a distortion matrix of low rank, *e.g.* $s = 1$ or 2 (see Table 1). Then, to get for $\Lambda_i(\mathscr{C}_{\mathrm{pub}})$ a generator matrix of the form (17) with $\Lambda_{i-1}(\boldsymbol{X}'')$ of full rank, one needs i to be large, while the dimensions of the $\Lambda_i(\mathscr{C}_{\mathrm{sec}})$ increase faster than for a Gabidulin code. Thus, for some parameters it is possible that the computation of the successive $\Lambda_i(\mathscr{C}_{\mathrm{pub}})$ provide successive codes with generator matrices of the form (17), so that $\Lambda_i(\mathscr{C}_{\mathrm{sec}})$ becomes the full code $\mathbb{F}_{q^m}^n$ before $\Lambda_{i-1}(\boldsymbol{X}'')$ reaches the full rank λ. The core of our extension in § 4 is the observation that there is no need for $\boldsymbol{X}''$ to have full rank to break the scheme.

Example 1. According to the Table 1, suppose that $n = 26$, $k = 18$, $\lambda = 6$ and $s = 1$. Then, for $\boldsymbol{X}''$ to have full rank $\lambda = 6$, while $\boldsymbol{X}$ has rank 1, we need to compute $\Lambda_6(\mathscr{C}_{\text{pub}})$. But since the secret code has dimension 18 and it is a twisted Gabidulin code, we deduce that $\dim \Lambda_6(\mathscr{C}_{\text{sec}}) \geqslant 26$ and, since $n = 26$, this code is nothing else than $\mathbb{F}_{q^m}^{26}$. Thus, for such parameters, we cannot apply the Overbeck's attack. In fact, even if instantiated with a Gabidulin code, the Overbeck's attack would fail for such parameters.

4 An extension of Overbeck's attack

As explained earlier, Overbeck's technique consists in applying the q–sum operator Λ_i to the public code, for an i such that the public code has a generator matrix of the form

$$\begin{pmatrix} \boldsymbol{I}_\lambda & \boldsymbol{0} \\ \boldsymbol{0} & \Lambda_i(\boldsymbol{G}_{\text{sec}}) \end{pmatrix} \boldsymbol{P}, \tag{18}$$

where $\Lambda_i(\mathscr{C}_{\text{sec}}) \neq \mathbb{F}_{q^m}^n$. This entails that the dual code has a generator matrix of the form

$$(\boldsymbol{0} \mid \boldsymbol{H})\left(\boldsymbol{P}^{-1}\right)^\top, \tag{19}$$

where $\boldsymbol{H}^\top$ is a parity–check matrix of $\Lambda_i(\mathscr{C}_{\text{sec}})$. Then, a valid column scrambler can be computed by solving a linear system. The point of this section is to prove that one can relax the constraint on i and only expect $\Lambda_i(\boldsymbol{G}_{\text{pub}})$ to have a generator matrix "splitting in two blocks", *i.e.*

$$\begin{pmatrix} \boldsymbol{Y} & \boldsymbol{0} \\ \boldsymbol{0} & \Lambda_i(\boldsymbol{G}_{\text{sec}}) \end{pmatrix} \boldsymbol{P}, \tag{20}$$

without requiring $\boldsymbol{Y}$ to have full rank λ.

Note that the above-described setting is precisely what happens to $\Lambda_1(\boldsymbol{G}_{\text{pub}})$ when $s = \mathbf{rank}(\boldsymbol{X}) < \lambda/2$, see § 3.4, Example 1 or § 5.3.

Example 2. Back to Example 1, for such parameters, even instantiated with a Gabidulin code, the Overbeck's attack fails because there is not any $i > 0$ which gives a matrix of the shape (18). However, under some assumptions on the parameters of the code, it is likely that $\Lambda_1(\boldsymbol{G}_{\text{pub}})$ has a generator matrix of the shape (20).See for instance Lemmas 4 and 6.

4.1 Sketch of the attack

Now, let us explain how to find the hidden splitting structure (20) without any knowledge of the scrambling matrix $\boldsymbol{P}$. Assume that $\Lambda_i(\mathscr{C}_{\text{pub}})$ has a generator matrix of the form

$$\begin{pmatrix} \boldsymbol{Y} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_i \end{pmatrix} \boldsymbol{P}, \tag{21}$$

where $\boldsymbol{Y}$ is a matrix with λ columns, $\boldsymbol{G}_i$ is a generator matrix of $\Lambda_i(\mathscr{C}_{\text{sec}})$ and $\mathscr{C}_{\text{sec}}$ is the hidden code of dimension k. The code $\mathscr{C}_{\text{sec}}$ could be either a Gabidulin code in the case of classical GPT or a twisted Gabidulin code (see respectively § 1.5 and the beginning of § 5).

The idea consists in computing the *right stabilizer algebra* of $\Lambda_i(\mathscr{C}_{\text{pub}})$:

$$\text{Stab}_{\text{right}}(\Lambda_i(\mathscr{C}_{\text{pub}})) \stackrel{\text{def}}{=} \{\boldsymbol{M} \in \mathcal{M}_{n+\lambda}(\mathbb{F}_q) \mid \Lambda_i(\mathscr{C}_{\text{pub}})\boldsymbol{M} \subseteq \Lambda_i(\mathscr{C}_{\text{pub}})\}.$$

This algebra can be computed by solving a linear system (see § 4.2). It turns out that it contains two peculiar matrices, namely:

$$\boldsymbol{E}_1 = \boldsymbol{P}^{-1}\begin{pmatrix} \boldsymbol{I}_\lambda & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}\boldsymbol{P} \quad \text{and} \quad \boldsymbol{E}_2 = \boldsymbol{P}^{-1}\begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_n \end{pmatrix}\boldsymbol{P}. \tag{22}$$

The core of the attack consists in computing these two matrices, or more precisely conjugates of these matrices, and then consider the code $\mathscr{C}_{\text{pub}}\boldsymbol{E}_2$ which is somehow right equivalent to $\mathscr{C}_{\text{sec}}$. In particular, the right multiplication by $\boldsymbol{E}_2$ will annihilate the distortion matrix $\boldsymbol{X}$. Let us now present the approach in more detail.

4.2 Some algebraic preliminaries

Split and indecomposable codes. The first crucial notion is that of *split* or *decomposable* codes.

Definition 3. *A code $\mathscr{C} \subseteq \mathbb{F}_{q^m}^n$ of dimension k is said to* split *if it has a generator matrix of the form*

$$\begin{pmatrix} \boldsymbol{G}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_2 \end{pmatrix}\boldsymbol{Q},$$

for some matrices $\boldsymbol{G}_1 \in \mathcal{M}_{a,b}(\mathbb{F}_{q^m}), \boldsymbol{G}_2 \in \mathcal{M}_{k-a,n-b}(\mathbb{F}_{q^m})$ and $\boldsymbol{Q} \in \mathbf{GL}_n(\mathbb{F}_q)$. If no such block–wise decomposition exists, then the code is said to be indecomposable*.*

Remark 10. Considering the code as a space of matrices, being split means that the code is the direct sum of two subcodes whose row supports (*i.e.* the sum of the row spaces of their elements) are in direct sum. This is the rank metric counterpart of Hamming codes which are the direct sum of two subcodes with disjoint Hamming supports. Note that this property is very rare and corresponds to somehow very *degenerated* codes.

Stabilizer algebras and conductors. We now define the notions that we will use throughout this section. Stabilizers are useful invariants of codes, also called *idealizers* in the literature. Conductors, are used for instance in [12] and have often been used in cryptanalysis of schemes based on algebraic Hamming metric codes, for instance [14,13,5].

Definition 4. *Let $\mathscr{C} \subseteq \mathbb{F}_{q^m}^{n_1}$ and $\mathscr{D} \subseteq \mathbb{F}_{q^m}^{n_2}$ be two $\mathbb{F}_{q^m}$–linear codes of respective length n_1, n_2. The* conductor of $\mathscr{C}$ into $\mathscr{D}$ *is defined as:*

$$Cond(\mathscr{C}, \mathscr{D}) \stackrel{def}{=} \{\boldsymbol{A} \in \mathcal{M}_{n_1,n_2}(\mathbb{F}_q) \mid \forall \boldsymbol{c} \in \mathscr{C},\ \boldsymbol{c}\boldsymbol{A} \in \mathscr{D}.\}$$

It is an $\mathbb{F}_q$–vector subspace of $\mathcal{M}_{n_1,n_2}(\mathbb{F}_q)$. Moreover, when $\mathscr{C} = \mathscr{D}$, then the conductor is an algebra which is usually called right stabilizer *or* right idealizer *of $\mathscr{C}$ and denoted*

$$\mathit{Stab}_{\mathit{right}}(\mathscr{C}) \stackrel{\mathit{def}}{=} \mathit{Cond}(\mathscr{C}, \mathscr{C}) = \{\boldsymbol{A} \in \mathcal{M}_{n_1}(\mathbb{F}_q) \mid \forall \boldsymbol{c} \in \mathscr{C},\ \boldsymbol{c}\boldsymbol{A} \in \mathscr{C}\}.$$

Relation to our problem. The first important point is that almost any code of length $n+\lambda$ has a *trivial right stabilizer*, *i.e.* a stabilizer of the form $\{\alpha \boldsymbol{I}_{n+\lambda} \mid \alpha \in \mathbb{F}_q\}$. However, the stabilizer of $\Lambda_i(\mathscr{C}_{\text{pub}})$ is non trivial, since it contains the matrices (22).

The second point is that $\text{Stab}_{\text{right}}(\Lambda_i(\mathscr{C}_{\text{pub}}))$ can be computed by solving a linear system. In general, given a parity–check matrix $\boldsymbol{H}$ for $\mathscr{C}$, the elements of $\text{Stab}_{\text{right}}(\mathscr{C})$ are nothing but the solutions $\boldsymbol{M} \in \mathcal{M}_{n+\lambda}(\mathbb{F}_q)$ of the system

$$\boldsymbol{G}\boldsymbol{M}\boldsymbol{H}^\top = \boldsymbol{0}. \tag{23}$$

Idempotents and decomposition of the identity. The matrices $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ of (22) are *idempotents* of the right stabilizer algebra of $\Lambda_i(\mathscr{C}_{\text{pub}})$, *i.e.* elements satisfying $\boldsymbol{E}_1^2 = \boldsymbol{E}_1$ and $\boldsymbol{E}_2^2 = \boldsymbol{E}_2$. In addition, they provide what is usually called *a decomposition of the identity with orthogonal idempotents*. The general definition is given below.

Definition 5. *In a matrix algebra $\mathcal{A} \subseteq \mathcal{M}_n(\mathbb{F}_q)$, a tuple $\boldsymbol{E}_1, \dots, \boldsymbol{E}_r$ of nonzero idempotents are said to be a* decomposition of the identity into orthogonal idempotents *if they satisfy,*

$$\forall 1 \leqslant i, j \leqslant r,\ \boldsymbol{E}_i\boldsymbol{E}_j = \boldsymbol{0} \quad \textit{and} \quad \boldsymbol{E}_1 + \cdots + \boldsymbol{E}_r = \boldsymbol{I}.$$

Such a decomposition is said to be minimal *if none of the $\boldsymbol{E}_i$'s can be written as a sum of two nonzero orthogonal idempotents.*

Proposition 4. *A code $\mathscr{C} \subseteq \mathbb{F}_{q^m}^n$ is split if and only if $\mathit{Stab}_{\mathit{right}}(\mathscr{C})$ has a non-trivial decomposition of the identity into orthogonal idempotents.*

Proof. Suppose that $\text{Stab}_{\text{right}}(\mathscr{C})$ contains such a decomposition of the identity into orthogonal idempotents $\boldsymbol{I} = \boldsymbol{E}_1 + \cdots + \boldsymbol{E}_r$. Since the $\boldsymbol{E}_i$'s commute pairwise and are diagonalizable (indeed, being idempotent, they all cancelled by the split polynomial $X^2 - X$), they are simultaneously diagonalizable. Thus, there exists $\boldsymbol{Q} \in \mathbf{GL}_n(\mathbb{F}_q)$ such that

$$\boldsymbol{E}_1 = \boldsymbol{Q}^{-1} \begin{pmatrix} \boldsymbol{I}_{n_1} & & (0) \\ & \ddots & \\ (0) & & (0) \end{pmatrix} \boldsymbol{Q}, \dots,\ \boldsymbol{E}_r = \boldsymbol{Q}^{-1} \begin{pmatrix} (0) & & (0) \\ & \ddots & \\ (0) & & \boldsymbol{I}_{n_r} \end{pmatrix} \boldsymbol{Q},$$

for some positive integers $n_1, \dots, n_r$ such that $n_1 + \cdots + n_r = n$.

The code $\mathscr{C}' = \mathscr{C}\boldsymbol{Q}$ has the matrices

$$\boldsymbol{E}'_1 = \begin{pmatrix} \boldsymbol{I}_{n_1} & & (0) \\ & \ddots & \\ (0) & & (0) \end{pmatrix}, \dots, \ \boldsymbol{E}'_r = \begin{pmatrix} (0) & & (0) \\ & \ddots & \\ (0) & & \boldsymbol{I}_{n_r} \end{pmatrix} \tag{24}$$

in its right stabilizer algebra, and one can easily check that $\mathscr{C}' = \mathscr{C}'\boldsymbol{E}'_1 \oplus \cdots \oplus \mathscr{C}'\boldsymbol{E}'_r$, leading to a block–wise generator matrix of $\mathscr{C}'$. Thus, $\mathscr{C}$ has a generator matrix of the form

$$\begin{pmatrix} \boldsymbol{G}_1 & & (0) \\ & \ddots & \\ (0) & & \boldsymbol{G}_r \end{pmatrix} \boldsymbol{Q}^{-1}. \tag{25}$$

Conversely, if $\mathscr{C}$ has a generator matrix as in (25), one can easily deduce a decomposition of the identity in $\mathrm{Stab}_{\mathrm{right}}(\mathscr{C})$ into the idempotents (24). □

In particular, a code is indecomposable if and only if its right stabilizer algebra has no nontrivial idempotent. Such an algebra is said to be *local*.

A crucial aspect of minimal decompositions of the identity is the following, sometimes referred to as the Krull–Schmidt Theorem.

Theorem 3 ([15, Thm. 3.4.1]). *Let $\mathcal{A} \subseteq \mathcal{M}_n(\mathbb{F}_q)$ be a matrix algebra and $\boldsymbol{E}_1, \dots, \boldsymbol{E}_r$ and $\boldsymbol{F}_1, \dots, \boldsymbol{F}_s$ be two minimal decompositions of the identity into orthogonal idempotents. Then, $r = s$ and there exists $\boldsymbol{A} \in \mathcal{A}^\times$ such that, after possibly re-indexing the $\boldsymbol{F}_i$'s, we have $\boldsymbol{F}_i = \boldsymbol{A}\boldsymbol{E}_i\boldsymbol{A}^{-1}$, for any $i \in \{1, \dots, s\}$.*

In short: a minimal decomposition of the identity into idempotents is unique up to conjugation.

Algorithmic aspects. Given a matrix algebra, a decomposition of the identity into minimal idempotents can be efficiently computed using Friedl and Ronyái's algorithms [16,47]. Such a calculation is presented in the case of stabilizer algebras of codes in [12]. Further, in § 4.5, we present the calculation in a simple case which turns out to be the generic situation for our cryptanalysis.

4.3 Description of our extension of Overbeck's attack

The attack summarizes as follows. Recall that the public code $\mathscr{C}_{\mathrm{pub}}$ has a generator matrix

$$\boldsymbol{G}_{\mathrm{pub}} = (\boldsymbol{X} \mid \boldsymbol{G}_{\mathrm{sec}})\boldsymbol{P}.$$

Step 1. Compute i so that the code $\Lambda_i(\mathscr{C}_{\mathrm{pub}})$ splits as in (21), *i.e.* has the shape

$$\begin{pmatrix} \boldsymbol{Y} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_i \end{pmatrix} \boldsymbol{P} \tag{26}$$

where $\boldsymbol{G}_i$ is a generator matrix of $\Lambda_i(\mathscr{C}_{\mathrm{sec}})$ and $\boldsymbol{P}$ the column scrambler. In the sequel, **we suppose that** $\Lambda_i(\mathscr{C}_{\mathrm{sec}})$ **is indecomposable**. This assumption is discussed further in § 4.5.

Step 2. Compute $\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))$. We know that this algebra contains the matrices

$$\boldsymbol{E}_1 = \boldsymbol{P}^{-1}\begin{pmatrix} \boldsymbol{I}_\lambda & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}\boldsymbol{P} \quad \text{and} \quad \boldsymbol{E}_2 = \boldsymbol{P}^{-1}\begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_n \end{pmatrix}\boldsymbol{P}. \tag{27}$$

Next, using the algorithms described in [16,47,12], compute a minimal decomposition of the identity of $\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))$ into orthogonal idempotents. The following statement relates any such minimal decomposition to the matrices $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ in (27).

Lemma 5. *Assume that $\lambda < n$. Under the assumption that $\Lambda_i(\mathscr{C}_{\mathrm{sec}})$ is an indecomposable code, any minimal decomposition of the identity into orthogonal idempotents in $\Lambda_i(\mathscr{C}_{\mathrm{pub}})$ contains a unique element $\boldsymbol{F}$ of rank n. Moreover, there exists $\boldsymbol{A} \in Stab_{right}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))^\times$ such that $\boldsymbol{F} = \boldsymbol{A}^{-1}\boldsymbol{E}_2\boldsymbol{A}$ where $\boldsymbol{E}_2$ is the matrix introduced in (27).*

Proof. Consider the pair $\boldsymbol{E}_1, \boldsymbol{E}_2$ introduced in (27). The matrix $\boldsymbol{E}_2$ has rank n and projects the code $\Lambda_i(\mathscr{C}_{\mathrm{pub}})$ onto the code with generator matrix $(\boldsymbol{0}\ \boldsymbol{G}_i)\boldsymbol{P}$, where $\boldsymbol{G}_i$ is a generator matrix of $\Lambda_i(\mathscr{C}_{\mathrm{sec}})$. Since $\Lambda_i(\mathscr{C}_{\mathrm{pub}})$ is supposed to be indecomposable, $\boldsymbol{E}_2$ cannot split into $\boldsymbol{E}_2 = \boldsymbol{E}_{21} + \boldsymbol{E}_{22}$ such that $\boldsymbol{E}_{21}\boldsymbol{E}_{22} = \boldsymbol{E}_{22}\boldsymbol{E}_{21} = \boldsymbol{0}$, since this would contradict the indecomposability of $\Lambda_i(\mathscr{C}_{\mathrm{sec}})$. Next, either $\boldsymbol{E}_1, \boldsymbol{E}_2$ is a minimal decomposition or, $\boldsymbol{E}_1$ splits into a sum of orthogonal idempotents (if the code with generator matrix $\boldsymbol{Y}$ splits). In the latter situation, one deduces a minimal decomposition of the identity of the form $\boldsymbol{E}_{11}, \dots, \boldsymbol{E}_{1r}, \boldsymbol{E}_2$. Now, Theorem 3, permits to conclude that any other minimal decomposition is conjugate to the previous one and hence contains a unique element of rank n which is conjugate with $\boldsymbol{E}_2$. □

Step 3. Once we have computed a minimal decomposition of the identity into minimal idempotents, according to Lemma 5 and Theorem 3, we have computed $\boldsymbol{F} \in \mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))$ of rank n satisfying $\boldsymbol{F} = \boldsymbol{A}^{-1}\boldsymbol{E}_2\boldsymbol{A}$ for some unknown matrix $\boldsymbol{A} \in \mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))^\times$.

Proposition 5. *The code, $\mathscr{C}_{\mathrm{pub}}\boldsymbol{F}$ is contained in the code with generator matrix*

$$\left(\boldsymbol{0} \mid \overline{\boldsymbol{G}}^{i}_{sec}\right)\boldsymbol{P}\boldsymbol{A},$$

where $\overline{\boldsymbol{G}}^{i}_{sec}$ is a generator matrix of the code $\overline{\mathscr{C}}^{i}_{sec}$ introduced in Definition 2.

Before proving the previous statement, let us discuss it quickly. The result may seem disappointing since, even if we discarded the distortion matrix, we do not recover exactly the secret code. However,

1. the approach is relevant for small i's, and if $i \leqslant t$, where t is the rank of the error term in the encryption process, then, the algorithm described in § 2.2 decodes $\overline{\mathscr{C}}^t_{\mathrm{sec}}$ (and hence $\overline{\mathscr{C}}^i_{\mathrm{sec}}$ since it is contained in $\overline{\mathscr{C}}^t_{\mathrm{sec}}$) as efficiently as $\mathscr{C}_{\mathrm{sec}}$ itself.
2. In § 4.5, we provide some heuristic claiming that, most of the time, $\mathscr{C}_{\mathrm{pub}}\boldsymbol{F}$ is nothing but the code with generator matrix

$$(\mathbf{0} \mid \boldsymbol{G}_{\mathrm{sec}})\, \boldsymbol{P}\boldsymbol{A}.$$

Proof (of Proposition 5). Recall that $\boldsymbol{F} = \boldsymbol{A}^{-1}\boldsymbol{E}_2\boldsymbol{A}$ for some matrix $\boldsymbol{A} \in \mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))$. Then, since $\boldsymbol{A}$ is invertible, we deduce that $\Lambda_i(\mathscr{C}_{\mathrm{pub}})\boldsymbol{A}^{-1} = \Lambda_i(\mathscr{C}_{\mathrm{pub}})$. Therefore,

$$\Lambda_i(\mathscr{C}_{\mathrm{pub}})\boldsymbol{F} = \Lambda_i(\mathscr{C}_{\mathrm{pub}})\boldsymbol{E}_2\boldsymbol{A}.$$

From (26) and (27), the code $\Lambda_i(\mathscr{C}_{\mathrm{pub}})\boldsymbol{E}_2$ has a generator matrix of the form $(\mathbf{0} \mid \boldsymbol{G}_i)\boldsymbol{P}$ and hence the code $\Lambda_i(\mathscr{C}_{\mathrm{pub}})\boldsymbol{F}$ has a generator matrix

$$(\mathbf{0} \mid \boldsymbol{G}_i)\boldsymbol{P}\boldsymbol{A}. \tag{28}$$

Next, the code $\mathscr{C}_{\mathrm{pub}}$ is contained in $\Lambda_i(\mathscr{C}_{\mathrm{pub}})$ but also in $\overline{\Lambda_i(\mathscr{C}_{\mathrm{pub}})}^i$. Moreover, according to Remark 7, we have

$$\mathscr{C}_{\mathrm{pub}} \subseteq \overline{\mathscr{C}}^i_{\mathrm{pub}} = \bigcap_{j=0}^{i} \left(\Lambda_i(\mathscr{C}_{\mathrm{pub}})\right)^{[-j]}.$$

Since both $\boldsymbol{P}$ and $\boldsymbol{A}$ have their entries in $\mathbb{F}_q$, they commute with the operations of raising to any q–th power and we deduce that

$$\mathscr{C}_{\mathrm{pub}}\boldsymbol{F} \subseteq \overline{\Lambda_i(\mathscr{C}_{\mathrm{pub}})}^i \boldsymbol{F}.$$

Then, from (28), we deduce that $\mathscr{C}_{\mathrm{pub}}\boldsymbol{F}$ is contained in the code with generator matrix

$$\left(\mathbf{0} \mid \overline{\boldsymbol{G}}^i_{\mathrm{sec}}\right) \boldsymbol{P}\boldsymbol{A}.$$

$\square$

Step 5. With the previous results at hand, given a ciphertext $\boldsymbol{y} = \boldsymbol{m}\boldsymbol{G}_{\mathrm{pub}} + \boldsymbol{e}$ with $\mathbf{rank}(\boldsymbol{e}) \leqslant t$, we can compute

$$\boldsymbol{y}\boldsymbol{F} = \boldsymbol{m}\boldsymbol{G}_{\mathrm{pub}}\boldsymbol{F} + \boldsymbol{e}\boldsymbol{F}.$$

Then, we remove its λ leftmost entries. Since $\boldsymbol{F}$ has its entries in $\mathbb{F}_q$, $\mathbf{rank}(\boldsymbol{e}\boldsymbol{F}) \leqslant \mathbf{rank}(\boldsymbol{e})$. Next, $\boldsymbol{m}\boldsymbol{G}_{\mathrm{pub}}\boldsymbol{F}$ with the λ leftmost entries removed is a codeword in $\overline{\Lambda_i(\mathscr{C}_{\mathrm{sec}})}^i$ which can be decoded using the algorithm introduced in 2.2. This yields the plaintext $\boldsymbol{m}$.

Algorithm 2: Summary of the attack

Input: $\boldsymbol{G}_{\text{pub}}$, a ciphertext $\boldsymbol{y}$ and the rank of the error term t
Output: A pair $(\boldsymbol{me}) \in \mathbb{F}_{q^m}^k \times \mathbb{F}_{q^m}^n$ such that $\mathbf{rank}(\boldsymbol{e}) = t$ and $\boldsymbol{y} = \boldsymbol{m}\boldsymbol{G}_{\text{pub}} + \boldsymbol{e}$ or '?' if fails

```
Compute a generator matrix of Λ_i(𝒞_pub) for the least i for which the code
  splits.
if no such i exists then
  └ Return '?'
Compute a minimal decomposition of the identity of Stab_right(Λ_i(𝒞)) and
  extract its unique term F of rank n.
if no such F exists then
  └ Return '?'
Compute yF and apply to it the decoder described in 2.2.
return the output m of the decoder (possibly '?' if the decoder fails).
```

4.4 Summary of the attack

According to the description in § 4.3, the attack is now summarized in Algorithm 2 below.

4.5 Discussions and simplifications

For the attack presented in Algorithm 2 to work, several assumptions are made. Here we discuss these assumptions and their rationale. We also point out that in our specific case, the algebra $\text{Stab}_{\text{right}}(\Lambda_i(\mathscr{C}_{\text{pub}}))$ will be very specific. This may permit to avoid to consider the difficult cases of Friedl Ronyái's algorithms.

Indecomposability of $\boldsymbol{\Lambda_i}(\mathscr{C}_{\mathbf{sec}})$. An important assumption for the attack to succeed is that $\Lambda_i(\mathscr{C}_{\text{sec}})$ does not split. Note first that in the classical GPT case, $\mathscr{C}_{\text{sec}}$ is a Gabidulin code. And so, this always holds as soon as $i < n - k$.

This is a consequence of the following statement and the fact that if $\mathscr{C}_{\text{sec}}$ is a Gabidulin code, and so for any $i > 0$, also $\Lambda_i(\mathscr{C}_{\text{sec}})$ is a Gabidulin code. Thus, according to the following statement it is indecomposable.

Proposition 6. *An MRD code $\mathscr{C} \subsetneq \mathbb{F}_{q^m}$ never splits.*

Proof. Let $\mathscr{C} \subseteq \mathbb{F}_{q^m}^n$ be an MRD code of dimension k. Suppose it splits into a direct sum of two codes $\mathscr{C}_1, \mathscr{C}_2$ of respective lengths n_1, n_2 and dimensions k_1, k_2. Then, $\mathscr{C}_1$ has codewords of rank weight $n_1 - k_1 + 1$ and $\mathscr{C}_2$ has words of weight $n_2 - k_2 + 1$. Such words are also words of $\mathscr{C}$ and, since $\mathscr{C}$ is MRD, we have

$$
\begin{aligned}
n_1 - k_1 + 1 &\geqslant n - k + 1 \\
n_2 - k_2 + 1 &\geqslant n - k + 1
\end{aligned}
$$

Summing up these two inequalities and using the fact that $n_1 + n_2 = n$ and $k_1 + k_2 = k$, we get a contradiction. □

In the general case of twisted Gabidulin codes the situation is more complicated. However, twisted Gabidulin codes are contained in Gabidulin codes of larger dimensions, hence so are their images by the Λ_i operator. It seems very unlikely that a Gabidulin code could contain large subcodes that split.

On the structure of $\mathbf{Stab_{right}}(\boldsymbol{\Lambda_i}(\mathscr{C}_{\mathbf{pub}}))$. A crucial step of the attack is the computation of a decomposition of the identity of $\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))$ into a sum of orthogonal idempotents. For this, we referred to Friedl Rónyai [16,47]. Actually, our setting is rather specific and the structure of this stabilizer algebra is pretty well understood. Let us start with a proposition.

Proposition 7. *Let $\mathscr{C}$ be an $\mathbb{F}_{q^m}$–linear code of length $n+\lambda$ and dimension K with a generator matrix of the shape (21),* i.e.

$$\begin{pmatrix} \boldsymbol{G}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_2 \end{pmatrix},$$

with $\boldsymbol{G}_1 \in \mathcal{M}_{k_1,\lambda}(\mathbb{F}_{q^m})$ for some integer k_1 and $\boldsymbol{G}_2 \in \mathcal{M}_{k_2,n}(\mathbb{F}_{q^m})$ for some integer k_2 so that $k_1+k_2=K$. Denote by $\mathscr{C}_1$ and $\mathscr{C}_2$ the codes with respective generator matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$. Then any $\boldsymbol{M} \in \mathrm{Stab}_{right}(\mathscr{C})$ has the shape

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix},$$

where $\boldsymbol{A} \in \mathrm{Stab}_{right}(\mathscr{C}_1)$, $\boldsymbol{B} \in \mathrm{Cond}(\mathscr{C}_2,\mathscr{C}_1)$, $\boldsymbol{C} \in \mathrm{Cond}(\mathscr{C}_1,\mathscr{C}_2)$ and $\boldsymbol{D} \in \mathrm{Stab}_{right}(\mathscr{C}_2)$. □

Proof. Let $\boldsymbol{c}_1 \in \mathscr{C}_1$, then $(\boldsymbol{c}_1\ \boldsymbol{0}) \in \mathscr{C}$ and by definition of $\boldsymbol{M}$, $(\boldsymbol{c}_1\ \boldsymbol{0})\boldsymbol{M} = (\boldsymbol{c_1 A}\ \boldsymbol{c_1 B}) \in \mathscr{C}$. By definition of $\mathscr{C}$, we have $\boldsymbol{c}_1\boldsymbol{A} \in \mathscr{C}_1$ and $\boldsymbol{c}_1\boldsymbol{B} \in \mathscr{C}_2$. Since the previous assertions hold for any $\boldsymbol{c}_1 \in \mathscr{C}_1$, then we deduce that $\boldsymbol{A} \in \mathrm{Stab}_{\mathrm{right}}(\mathscr{C}_1)$ and $\boldsymbol{B} \in \mathrm{Cond}(\mathscr{C}_1,\mathscr{C}_2)$.

The result for $\boldsymbol{C},\boldsymbol{D}$ is obtained in the same way by considering $(\boldsymbol{0}\ \boldsymbol{c_2})\boldsymbol{M}$ for $\boldsymbol{c}_2 \in \mathscr{C}_2$. □

Consequently considering the generator matrix (26) of $\Lambda_i(\mathscr{C}_{\mathrm{pub}})$, elements of $\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))$ have the shape

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}, \tag{29}$$

where $\boldsymbol{A} \in \mathrm{Stab}_{\mathrm{right}}(\mathscr{C}_{\boldsymbol{Y}})$ ($\mathscr{C}_{\boldsymbol{Y}}$ being the code with generator matrix $\boldsymbol{Y}$), $\boldsymbol{B} \in \mathrm{Cond}(\Lambda_i(\mathscr{C}_{\mathrm{sec}}),\mathscr{C}_{\boldsymbol{Y}})$, $\boldsymbol{C} \in \mathrm{Cond}(\mathscr{C}_{\boldsymbol{Y}},\Lambda_i(\mathscr{C}_{\mathrm{sec}}))$ and $\boldsymbol{D} \in \mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{sec}}))$.

Here again, we claim that is very likely that the stabilizer algebras of $\mathscr{C}_{\boldsymbol{Y}}$ and $\Lambda_i(\mathscr{C}_{\mathrm{pub}})$ are trivial, *i.e.* contain only scalar multiples of the identity matrix and that the conductors $\mathrm{Cond}(\mathscr{C}_{\boldsymbol{Y}},\Lambda_i(\mathscr{C}_{\mathrm{sec}}))$ and $\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{sec}}))$ are zero. This claim is discussed further in § 4.7.

In such a situation, we have:

$$\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}})) = \left\{ \boldsymbol{P}^{-1} \begin{pmatrix} a\boldsymbol{I}_\lambda & \boldsymbol{0} \\ \boldsymbol{0} & b\boldsymbol{I}_n \end{pmatrix} \boldsymbol{P} \;\middle|\; a, b \in \mathbb{F}_q \right\}. \tag{30}$$

Hence this algebra has dimension 2 and the calculation of the matrix

$$\boldsymbol{P}^{-1} \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_n \end{pmatrix} \boldsymbol{P} \tag{31}$$

can be performed as follows.

1. First extract a singular matrix of $\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))$. For that, take $\boldsymbol{U}, \boldsymbol{V}$ a basis of $\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))$. If $\boldsymbol{V}$ is singular we are done. Otherwise, compute a root of the univariate polynomial $\det(\boldsymbol{U} + X\boldsymbol{V})$. This yields a singular element $\boldsymbol{R}$ of $\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))$ corresponding either to $a = 0$ or $b = 0$ in the description (30).
2. Next, rescale $\boldsymbol{R}$ as $\nu\boldsymbol{R}$ in order to get an idempotent element. If the obtained idempotent has rank n set $\boldsymbol{F} = \nu\boldsymbol{R}$, otherwise (it will have rank λ), set $\boldsymbol{F} = \boldsymbol{I}_{n+\lambda} - \nu\boldsymbol{R}$.

The obtained matrix $\boldsymbol{F}$ is nothing but the target matrix in (31). Therefore, one can even skip the proof of Proposition 5 and observe that the code $\mathscr{C}_{\mathrm{pub}}\boldsymbol{F}$ will be **exactly** the code with generator matrix

$$(\boldsymbol{0} \mid \boldsymbol{G}_{\mathrm{sec}})\, \boldsymbol{P}.$$

4.6 Complexity

Considering the previous simple case which remains very likely, we analyze the cost of the various computation steps.

- The computation of $\Lambda_i(\mathscr{C}_{\mathrm{pub}})$ can be done by iterating i successive Gaussian eliminations (we assume that raising an element of $\mathbb{F}_{q^m}$ to the q–th power can be done for free, for instance by representing $\mathbb{F}_{q^m}$ with a normal basis). Thus, a cost $O(in^\omega)$ operations in $\mathbb{F}_{q^m}$ and hence $O(im^2n^\omega)$ operations in $\mathbb{F}_q$. Here, ω denotes the usual exponent for the cost of the product of two $n \times n$ matrices.
- The computation of $\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))$ is done by solving the linear system (23). The system has n^2 unknowns in $\mathbb{F}_q$ and $k_i(n - k_i) = O(n^2)$ equations in $\mathbb{F}_{q^m}$ and hence $O(mn^2)$ equations in $\mathbb{F}_q$. This yields a cost of $O(mn^{2\omega})$ operations in $\mathbb{F}_q$ (see [9, Thm. 8.6] for the complexity of the resolution of a non square linear system).

In the aforementioned simple case, the remaining operations are negligible compared to the calculation of the stabilizer algebra, which turns out to be the bottleneck of the calculation. This overall cost is hence in

$$O(mn^{2\omega}) \;\text{ operations in } \mathbb{F}_q.$$

4.7 Discussion about the claims on conductors and stabilizers

Back to the description (29) of the elements of $\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{pub}}))$. Let us discuss the validity of the claim.

Conductors are likely to be zero. Let $\boldsymbol{C} \in \mathrm{Cond}(\mathscr{C}_{\boldsymbol{Y}}, \Lambda_i(\mathscr{C}_{\mathrm{sec}}))$, then the code $\mathscr{C}_{\boldsymbol{Y}}\boldsymbol{C}$ is a subcode of $\Lambda_i(\mathscr{C}_{\mathrm{sec}})$ and one proves easily that any element of $\mathscr{C}_{\boldsymbol{Y}}\boldsymbol{C}$ has a row support contained in the row space of $\boldsymbol{C}$. Since $\boldsymbol{C} \in \mathcal{M}_{\lambda,n}(\mathbb{F}_q)$, its rank is at most equal to λ and hence the code $\mathscr{C}_{\boldsymbol{Y}}\boldsymbol{C}$ has a row space contained in a space of dimension $\leqslant \lambda$. It seems unlikely that the code $\Lambda_t(\mathscr{C}_{\mathrm{sec}})$ contains such a space. In particular, this cannot happen if the minimum distance of $\Lambda_i(\mathscr{C}_{\mathrm{sec}})$ exceeds λ.

Now, consider $\boldsymbol{B} \in \mathrm{Cond}(\Lambda_i(\mathscr{C}_{\mathrm{sec}}), \mathscr{C}_{\boldsymbol{Y}})$. Suppose first that $\boldsymbol{B}$ has full rank. Since $\dim(\Lambda_i(\mathscr{C}_{\mathrm{sec}})) \gg \lambda$, the code $\Lambda_i(\mathscr{C}_{\mathrm{sec}})\boldsymbol{B}$ is likely to be equal to $\mathbb{F}_{q^m}^{\lambda}$ and hence cannot be contained in $\mathscr{C}_{\boldsymbol{Y}}$, a contradiction. If $\boldsymbol{B}$ has not full rank, then, the code $\Lambda_i(\mathscr{C}_{\mathrm{sec}})\boldsymbol{B}$ is likely to be equal to the subspace of $\mathbb{F}_{q^m}^{\lambda}$ of all the vectors whose row support is in the row space of $\boldsymbol{B}$ and we can assume that $\mathscr{C}_{\boldsymbol{Y}}$ has no such subspace. Indeed, if it did, it would entail that $\mathscr{C}_{\boldsymbol{Y}}^{\perp}$ (and hence $\Lambda_i(\mathscr{C}_{\mathrm{pub}})^{\perp}$ too) would have a parity-check matrix of the form $(\boldsymbol{0} \mid \boldsymbol{H}')(\boldsymbol{P^{-1}})^{\top}$ as in (19). Details are left to the reader.

Stabilizers are likely restrict to scalar matrices. For $\mathscr{C}_{\boldsymbol{Y}}$, this code is close to be random and random codes have trivial stabilizer algebras with a high probability.

For $\Lambda_i(\mathscr{C}_{\mathrm{sec}})$ the right stabilizer algebra might be a larger one. Indeed, regarding the proof of Proposition 2 (see [43, Thm. 4]) we can see that $\Lambda_t(\mathscr{C})$ is a code generated by the evaluations of q-monomials and such a code, when $n = m$ has a right stabilizer algebra equal to a matrix representation of $\mathbb{F}_{q^m}$. This is a consequence of the fact that an $\mathbb{F}_{q^m}$–space spanned by q–monomials is $\mathbb{F}_{q^m}$–linear on the left but also on the right. Thus, $\mathrm{Stab}_{\mathrm{right}}(\Lambda_i(\mathscr{C}_{\mathrm{sec}}))$ might be such a larger algebra. In this situation, the calculation of a decomposition of the identity into orthogonal idempotents is slightly more complicated but remains definitely possible in polynomial time using Friedl Rónyái algorithms.

5 Don't twist again

In this section we first show that, even for twisted Gabidulin codes, the application of the q-sum operator allows to *distinguish* them from random codes. It is therefore possible to apply the attack described in § 4 to the GPT cryptosystem instantiated with these codes. In the first part of this section we discuss the behaviour of raw twisted Gabidulin codes with respect to the operator Λ_i or equivalently, how the use of Λ_i allows to distinguish them from random codes. In the second part, we focus on q-operator applied to the corresponding public key and we will prove that even in this case, we have a generator matrix with a structure similar to (16) and that the corresponding codes split. This allows us to apply the results of § 4.

5.1 A distinguisher

First, recall Propositions 2 and 3 about the dimension of the q-sum operator applied respectively to twisted Gabidulin codes and to random codes. In particular, recall that if $\mathscr{C}$ is a random code, $\dim(\Lambda_i(\mathscr{C})) = (i+1)k$ with high probability. Then, we remark that, if $i < \frac{n-k-\ell}{\ell+1}$

$$\begin{aligned} \dim(\Lambda_i(\mathscr{C}_{\boldsymbol{g},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k])) &= k + i + \ell(i+1) < (i+1)k = \dim(\Lambda_i(\mathscr{C})) \quad (32)\\ &\iff i > \ell/(k-\ell-1), \quad (33) \end{aligned}$$

where $\mathscr{C}_{\boldsymbol{g},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ is a twisted Gabidulin code (see § 1.4).

Thus, the inequality $i > \ell/(k-\ell-1)$ is satisfied by any positive i, if $k > 2\ell+1$. We notice that this is often the case if we consider a small number of twists as in Table 1. This means that, even if the dimension of the q-sum applied to these codes is greater than that of the q-sum of a Gabidulin code, we can however still distinguish them for random codes.

Thus, this distinguisher can be exploited to construct an attack against the GPT cryptosystem instantiated with twisted Gabidulin codes, instead of classical ones.

5.2 The structure of $\boldsymbol{\Lambda_i(\mathscr{G}_{Tpub})}$

From now on, we consider the GPT cryptosystem instantiated with a twisted Gabidulin code $\mathscr{C}_{\boldsymbol{g},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ with the parameters defined in Assumption 1. We denote by $\boldsymbol{G}_{\text{Tpub}}$ the corresponding public key, obtained as (5) by just replacing $\boldsymbol{G}_{\text{sec}}$ with a generator matrix $\boldsymbol{G}_{\mathbf{T}}$ (of the form (4)) of the code $\mathscr{C}_{\boldsymbol{g},\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}[n,k]$ and by $\mathscr{G}_{\text{Tpub}}$ the linear code which has $\boldsymbol{G}_{\text{Tpub}}$ as generator matrix. Again, as for the Gabidulin codes scheme, we can discard the matrix $\boldsymbol{S}$.

We now apply the q-sum operator to $\boldsymbol{G}_{\text{Tpub}}$, and as (11), we get

$$\Lambda_i(\boldsymbol{G}_{\text{Tpub}}) = [\Lambda_i(\boldsymbol{X})|\Lambda_i(\boldsymbol{G}_{\mathbf{T}})]\boldsymbol{P},$$

where $\boldsymbol{P} \in \mathbf{GL}_{n+\lambda}(\mathbb{F}_q)$ is the column scrambler.

Let $i < \frac{n-\ell-k}{l+1}$ and write $\boldsymbol{X}$ (as in § 3.2) according to its rows.

Now, for simplicity we consider that $\ell = 1$, $\eta_1 = 1$ and $i = 1$. Recall that the structure of $\boldsymbol{G}_{\mathbf{T}}$ is given in (4). Then, we have

$$(\Lambda_1(\boldsymbol{X}) \mid \Lambda_1(\boldsymbol{G}_{\mathbf{T}})) = \left(\begin{array}{c|c} \boldsymbol{x}_0 & \boldsymbol{g} \\ \boldsymbol{x}_1 & \boldsymbol{g}^{[1]} \\ \vdots & \vdots \\ \boldsymbol{x}_{h_1} & \boldsymbol{g}^{[h_1]} + \boldsymbol{g}^{[k-1+t_1]} \\ \vdots & \vdots \\ \boldsymbol{x}_{k-1} & \boldsymbol{g}^{[k-1]} \\ \hline \boldsymbol{x}_0^{[1]} & \boldsymbol{g}^{[1]} \\ \boldsymbol{x}_1^{[1]} & \boldsymbol{g}^{[2]} \\ \vdots & \vdots \\ \boldsymbol{x}_{h_1-1}^{[1]} & \boldsymbol{g}^{[h_1]} \\ \boldsymbol{x}_{h_1}^{[1]} & \boldsymbol{g}^{[h_1+1]} + \boldsymbol{g}^{[k+t_1]} \\ \vdots & \vdots \\ \boldsymbol{x}_{k-2}^{[1]} & \boldsymbol{g}^{[k-1]} \\ \boldsymbol{x}_{k-1}^{[1]} & \boldsymbol{g}^{[k]} \end{array}\right) \longrightarrow \left(\begin{array}{c|c} \boldsymbol{x}_0 & \boldsymbol{g} \\ \boldsymbol{x}_1 & \boldsymbol{g}^{[1]} \\ \vdots & \vdots \\ \boldsymbol{x}_{h_1-1} & \boldsymbol{g}^{[h_1-1]} \\ \boldsymbol{x}_{h_1-1}^{[1]} & \boldsymbol{g}^{[h_1]} \\ \boldsymbol{x}_{h_1+1} & \boldsymbol{g}^{[h_1+1]} \\ \vdots & \vdots \\ \boldsymbol{x}_{k-1} & \boldsymbol{g}^{[k-1]} \\ \boldsymbol{x}_{k-1}^{[1]} & \boldsymbol{g}^{[k]} \\ \hline \boldsymbol{x}_0^{[1]} & \boldsymbol{g}^{[1]} \\ \boldsymbol{x}_1^{[1]} & \boldsymbol{g}^{[2]} \\ \vdots & \vdots \\ \boldsymbol{x}_{h_1-2}^{[1]} & \boldsymbol{g}^{[h_1-1]} \\ \boldsymbol{x}_{h_1} & \boldsymbol{g}^{[h_1]} + \boldsymbol{g}^{[k-1+t_1]} \\ \boldsymbol{x}_{h_1}^{[1]} & \boldsymbol{g}^{[h_1+1]} + \boldsymbol{g}^{[k+t_1]} \\ \boldsymbol{x}_{h_1+1}^{[1]} & \boldsymbol{g}^{[h_1+2]} \\ \vdots & \vdots \\ \boldsymbol{x}_{k-2}^{[1]} & \boldsymbol{g}^{[k-1]} \end{array}\right)$$

where the second matrix is obtained by permuting the rows of the first one. We now observe that the first block of the second matrix can be rewritten as $[\tilde{\boldsymbol{X}}' | \mathbf{M}_{k+1}(\boldsymbol{g})]$ and so, after performing row elimination, we get

$$\left(\begin{array}{c|c} \tilde{\boldsymbol{X}}' & \mathbf{M}_{k+1}(\boldsymbol{g}) \\ \hline \boldsymbol{x}_{h_1} - \boldsymbol{x}_{h_1-1}^{[1]} & \boldsymbol{g}^{[k-1+t_1]} \\ \boldsymbol{x}_{h_1}^{[1]} - \boldsymbol{x}_{h_1+1} & \boldsymbol{g}^{[k+t_1]} \\ \hline \boldsymbol{x}_0^{[1]} - \boldsymbol{x}_1 & \mathbf{0} \\ \boldsymbol{x}_1^{[1]} - \boldsymbol{x}_2 & \mathbf{0} \\ \vdots & \vdots \\ \boldsymbol{x}_{h_1-2}^{[1]} - \boldsymbol{x}_{h_1-1} & \mathbf{0} \\ \boldsymbol{x}_{h_1+1}^{[1]} - \boldsymbol{x}_{h_1+2} & \mathbf{0} \\ \vdots & \vdots \\ \boldsymbol{x}_{k-2}^{[1]} - \boldsymbol{x}_{k-1} & \mathbf{0} \end{array}\right)$$

Therefore, we have the following result.

Lemma 6. *Let $i < \frac{n-\ell-k}{\ell+1}$. Then, up to row elimination*

$$(\Lambda_i(\boldsymbol{X}) \mid \Lambda_i(\boldsymbol{G}_T)) = \begin{pmatrix} \boldsymbol{Y} & \Lambda_i(\boldsymbol{G}_{\mathbf{T}}) \\ \tilde{\boldsymbol{X}} & \mathbf{0} \end{pmatrix} \tag{34}$$

where,

$$\tilde{\boldsymbol{X}} = \begin{cases} \begin{pmatrix} \boldsymbol{X}_T'' \end{pmatrix} & \in \quad \mathcal{M}_{k-1-2\ell}(\mathbb{F}_{q^m}) \quad \text{if } i = 1 \\ \begin{pmatrix} \Lambda_{i-1}(\boldsymbol{X}_T'') \\ \boldsymbol{X}''' \end{pmatrix} & \in \mathcal{M}_{i(k-1-2\ell)+(i-1)\ell}(\mathbb{F}_{q^m}) \text{ if } i > 1 \end{cases}$$

$\boldsymbol{Y} \in \mathcal{M}_{k+i+\ell(i+1),\lambda}(\mathbb{F}_{q^m})$ and the matrix $\boldsymbol{X}_T''$ is defined as,

$$\boldsymbol{X}_T'' = \boldsymbol{X}^{[1]}_{\{0,\dots,k-2\}\setminus\{h_i-1,h_i|1\leqslant i\leqslant\ell\}} - \boldsymbol{X}_{\{1,\dots,k-1\}\setminus\{h_i,h_i+1|1\leqslant i\leqslant\ell\}}, \tag{35}$$

where $\boldsymbol{X}^{[1]}_{\{0,\dots,k-2\}\setminus\{h_i-1,h_i|1\leqslant i\leqslant\ell\}}$ is a submatrix of $\boldsymbol{X}^{[1]}$ composed by the first $k-1$ rows except all the (h_i-1)-th, h_i-th rows and $\boldsymbol{X}_{\{1,\dots,k-1\}\setminus\{h_i,h_i+1|1\leqslant i\leqslant\ell\}}$ is a submatrix of $\boldsymbol{X}$ determined by all the rows, starting from the second one, except the h_i-th, h_i+1-th ones. Finally, $\boldsymbol{X}''' \in \mathcal{M}_{i-1,\lambda}(\mathbb{F}_{q^m})$.

Proof. Using the same elimination techniques as before, we can extend the proof to the case $\ell > 1$, $\boldsymbol{\eta} \in (\mathbb{F}_{q^m} \setminus \{0\})^\ell$ and $i > 1$. □

Even in this case, we show that it suffices to consider $i = 1$ to attack the corresponding GPT scheme.

5.3 Attacking the system for small i's.

We now consider $i = 1$. Then by Lemma 6, $(\Lambda_i(\boldsymbol{X}) \mid \Lambda_i(\boldsymbol{G}_T))$ can be transformed into

$$\begin{pmatrix} \boldsymbol{Y} & \Lambda_1(\boldsymbol{G}_{\mathbf{T}}) \\ \boldsymbol{X}_T'' & \mathbf{0} \end{pmatrix}$$

As in § 3.4 (see Lemma 4), under some assumptions on the parameters, we can split the previous matrix into two blocks.

Lemma 7. *If $k \geqslant 4s + 2\ell + 1$, then, with a high probability,*

$$\begin{pmatrix} \mathbf{0} & \Lambda_1(\boldsymbol{G}_{\mathbf{T}}) \\ \boldsymbol{X}_T'' & \mathbf{0} \end{pmatrix} \tag{36}$$

up to row eliminations.

Proof. The proof is analogous to the proof of Lemma 4. First we prove that $\mathbf{RowSp}_{\mathbb{F}_{q^m}}(\boldsymbol{X}_T'') = \mathbf{RowSp}_{\mathbb{F}_{q^m}}(\Lambda_1(\boldsymbol{X}))$ with a high probability. Again, we consider the submatrix of $\boldsymbol{X}_T''$ in $\mathcal{M}_{\lfloor\frac{k-1-2\ell}{2}\rfloor}(\mathbb{F}_{q^m})$ obtained by alternatively selecting rows of $\boldsymbol{X}_T''$. This matrix is uniformly random and by Proposition 3, if $\frac{k-1-2\ell}{2} \geqslant 2s$ (which is true by assumption), it has rank equal to the rank of $\Lambda_1(\boldsymbol{X})$ with a high probability. Thus the equality $\mathbf{RowSp}_{\mathbb{F}_{q^m}}(\boldsymbol{X}_T'') = \mathbf{RowSp}_{\mathbb{F}_{q^m}}(\Lambda_1(\boldsymbol{X}))$ holds.

The result follows by noting that $\mathbf{RowSp}_{\mathbb{F}_{q^m}}(\boldsymbol{Y}) \subseteq \mathbf{RowSp}_{\mathbb{F}_{q^m}}(\Lambda_1(\boldsymbol{X}))$. □

Therefore we can apply the attack of § 4 in order to break the corresponding GPT cryptosystem.

Remark 11. Notice that, if $\mathbf{rank}(\boldsymbol{X}) = s \geqslant \lambda/2$, then $\mathbf{rank}(\boldsymbol{X}_T) = \lambda$ with high probability and we can apply the Overbeck's attack to this scheme. In fact, in this case (as in § 3.3), $\dim(\Lambda_1(\mathscr{G}_{Tpub})^\perp) = n - k - 1 - 2\ell$, and so the code $\Lambda_1(\mathscr{G}_{Tpub})$ admits a parity check matrix whose first λ columns are $\mathbf{0}$. We can then compute a valid column scrambler and attack the system.

More generally, we can apply this attack to any $i < \frac{n-\ell-k}{\ell+1}$ for which

$$\mathbf{rank}(\tilde{\boldsymbol{X}}) = \lambda,$$

where $\tilde{\boldsymbol{X}}$ is defined in Lemma 6.

Conclusion

In this paper, we present new observations on the decoding of Gabidulin codes. These allow us to introduce a decoder for twisted Gabidulin codes up to a certain threshold, which may be less than half of the minimum distance.

We then propose an extension of the Overbeck's attack on GPT-like systems instantiated on Gabidulin or related codes such as twisted Gabidulin codes. This attack is efficient as soon as the secret code $\Lambda_i(\mathscr{C}_{\text{sec}})$ has a small dimension compared to the dimension of $\Lambda_i(\mathscr{C})$, where $\mathscr{C}$ is a random code. One of the interesting things about our approach is that it succeeds even when the distortion matrix has a low rank, which might cause the Overbeck's attack fails. Our attack extension allows to break the proposal of [43].

References

1. Aguilar Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Bros, M., Couvreur, A., Deneuville, J.C., Gaborit, P., Zémor, G., Hauteville, A.: Rank quasi cyclic (RQC). Second Round submission to NIST Post-Quantum Cryptography call (Apr 2020), `https://pqc-rqc.org`
2. Aragon, N., Blazy, O., Deneuville, J.C., Gaborit, P., Hauteville, A., Ruatta, O., Tillich, J.P., Zémor, G., Aguilar Melchor, C., Bettaieb, S., Bidoux, L., Bardet, M., Otmani, A.: ROLLO (merger of Rank-Ouroboros, LAKE and LOCKER). Second round submission to the NIST post-quantum cryptography call (Mar 2019), `https://pqc-rollo.org`
3. Aragon, N., Blazy, O., Gaborit, P., Hauteville, A., Zémor, G.: Durandal: a rank metric based signature scheme. In: Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III. LNCS, vol. 11478, pp. 728–758. Springer (2019). `https://doi.org/10.1007/978-3-030-17659-4_25`, `https://doi.org/10.1007/978-3-030-17659-4_25`
4. Aragon, N., Gaborit, P., Hauteville, A., Tillich, J.P.: A new algorithm for solving the rank syndrome decoding problem. In: 2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018. pp. 2421–2425. IEEE (2018). `https://doi.org/10.1109/ISIT.2018.8437464`

5. Barelli, E., Couvreur, A.: An efficient structural attack on NIST submission DAGS. In: Peyrin, T., Galbraith, S. (eds.) Advances in Cryptology - ASIACRYPT'18. LNCS, vol. 11272, pp. 93–118. Springer (Dec 2018)
6. Beelen, P., Bossert, M., Puchinger, S., Rosenkilde, J.: Structural properties of twisted Reed–Solomon codes with applications to cryptography. In: 2018 IEEE International Symposium on Information Theory (ISIT). pp. 946–950 (2018). https://doi.org/10.1109/ISIT.2018.8437923
7. Beelen, P., Puchinger, S., Rosenkilde né Nielsen, J.: Twisted Reed–Solomon codes. In: 2017 IEEE International Symposium on Information Theory (ISIT). pp. 336–340 (2017). https://doi.org/10.1109/ISIT.2017.8006545
8. Bombar, M., Couvreur, A.: Decoding supercodes of Gabidulin codes and applications to cryptanalysis. In: Cheon, J.H., Tillich, J.P. (eds.) Post-Quantum Cryptography. pp. 3–22. Springer International Publishing, Cham (2021)
9. Bostan, A., Chyzak, F., Giusti, M., Lebreton, R., Lecerf, G., Salvy, B., Schost, E.: Algorithmes Efficaces en Calcul Formel. Frédéric Chyzak (auto-édit.), Palaiseau (Sep 2017), https://hal.archives-ouvertes.fr/AECF/
10. Coggia, D., Couvreur, A.: On the security of a Loidreau's rank metric code based encryption scheme. In: WCC 2019 - Workshop on Coding Theory and Cryptography. Saint Jacut de la mer, France (2019)
11. Coggia, D., Couvreur, A.: On the security of a Loidreau's rank metric code based encryption scheme. Des. Codes Cryptogr. **88**, 1941–1957 (2020)
12. Couvreur, A., Debris-Alazard, T., Gaborit, P.: On the hardness of code equivalence problems in rank metric (Nov 2020), https://hal.archives-ouvertes.fr/hal-02997801, working paper or preprint
13. Couvreur, A., Márquez-Corbella, I., Pellikaan, R.: Cryptanalysis of McEliece cryptosystem based on algebraic geometry codes and their subcodes. IEEE Trans. Inform. Theory **63**(8), 5404–5418 (8 2017)
14. Couvreur, A., Otmani, A., Tillich, J.P.: Polynomial time attack on wild McEliece over quadratic extensions. IEEE Trans. Inform. Theory **63**(1), 404–427 (1 2017)
15. Drodz, Y.A., Kirichenko, V.V.: Finite dimensional algebras. Springer–Verlag Berlin Heidelberg (1994), original Russian edition published by: Publisher of Kiev State University, Kiev 1980, Translated by V. Dlab
16. Friedl, K., Rónyai, L.: Polynomial time solutions of some problems of computational algebra. In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing. pp. 153–162. STOC '85, Association for Computing Machinery, New York, NY, USA (1985). https://doi.org/10.1145/22145.22162, https://doi.org/10.1145/22145.22162
17. Gabidulin, E., Rashwan, H., Honary, B.: On improving security of GPT cryptosystems. In: Proc. IEEE Int. Symposium Inf. Theory - ISIT. pp. 1110–1114. IEEE (2009)
18. Gabidulin, E.M.: Theory of codes with maximum rank distance. Problemy Peredachi Informatsii **21**(1), 3–16 (1985)
19. Gabidulin, E.M.: Public-key cryptosystems based on linear codes over large alphabets : efficiency and weakness. In: Farrell, P.G. (ed.) 4^{th} IMA conference on cryptography and coding, the Institute of Mathematics and its Applications. pp. 17–31 (1993)
20. Gabidulin, E.M.: Attacks and counter-attacks on the GPT public key cryptosystem. Des. Codes Cryptogr. **48**(2), 171–177 (2008)
21. Gabidulin, E.M., Ourivski, A.V.: Modified GPT PKC with right scrambler. Electron. Notes Discrete Math. **6**, 168–177 (2001). https://doi.org/10.1016/S1571-0653(04)00168-4, http://dx.doi.org/10.1016/S1571-0653(04)00168-4

22. Gabidulin, E.M., Paramonov, A.V., Tretjakov, O.V.: Ideals over a non-commutative ring and their applications to cryptography. In: Advances in Cryptology - EUROCRYPT'91. pp. 482–489. No. 547 in LNCS, Brighton (Apr 1991)
23. Gaborit, P., Murat, G., Ruatta, O., Zémor, G.: Low rank parity check codes and their application to cryptography. In: Proceedings of the Workshop on Coding and Cryptography WCC'2013. Bergen, Norway (2013), `www.selmer.uib.no/WCC2013/pdfs/Gaborit.pdf`
24. Gaborit, P., Ruatta, O., Schrek, J.: On the complexity of the rank syndrome decoding problem. IEEE Trans. Inform. Theory **62**(2), 1006–1019 (2016)
25. Ghatak, A.: Extending Coggia-Couvreur attack on Loidreau's rank-metric cryptosystem. Des. Codes Cryptogr. **90**, 215–238 (2022)
26. Gibson, K.: Severely denting the Gabidulin version of the McEliece public key cryptosystem. Des. Codes Cryptogr. **6**(1), 37–45 (1995)
27. Gibson, K.: The security of the Gabidulin public key cryptosystem. In: Maurer, U. (ed.) Advances in Cryptology - EUROCRYPT '96. LNCS, vol. 1070, pp. 212–223. Springer (1996)
28. Goss, D.: Basic Structures of Function Field arithmetic, Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)], vol. 35. Springer-Verlag, Berlin (1996)
29. Kadir, W.K., Li, C.: On decoding additive generalized twisted Gabidulin codes. Cryptography Commun. **12**(5), 987–1009 (2020). `https://doi.org/10.1007/s12095-020-00449-9`
30. Kadir, W.K., Li, C., Zullo, F.: On interpolation-based decoding of a class of maximum rank distance codes (2021), `https://arxiv.org/abs/2105.03115`
31. Li, C.: Interpolation-based decoding of nonlinear maximum rank distance codes. In: 2019 IEEE International Symposium on Information Theory (ISIT). pp. 2054–2058 (2019). `https://doi.org/10.1109/ISIT.2019.8849472`
32. Li, C., Kadir, W.K.: On decoding additive generalized twisted gabidulin codes. In: Proceedings of the International Workshop on Coding and Cryptography WCC 2019 (2019)
33. Loidreau, P.: A Welch–Berlekamp like algorithm for decoding Gabidulin codes. In: Ytrehus, Ø. (ed.) Coding and Cryptography. pp. 36–45. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
34. Loidreau, P.: Designing a rank metric based McEliece cryptosystem. In: Sendrier, N. (ed.) Post-Quantum Cryptography 2010. LNCS, vol. 6061, pp. 142–152. Springer (2010)
35. McEliece, R.J.: A Public-Key System Based on Algebraic Coding Theory, pp. 114–116. Jet Propulsion Lab (1978), dSN Progress Report 44
36. Ore, Ø.: On a special class of polynomials. Trans. Amer. Math. Soc. **35**(3), 559–584 (1933)
37. Otmani, A., Talé-Kalachi, H., Ndjeya, S.: Improved cryptanalysis of rank metric schemes based on Gabidulin codes. CoRR **abs/1602.08549** (2016), `http://arxiv.org/abs/1602.08549`
38. Ourivski, A.V., Gabidulin, E.M.: Column scrambler for the GPT cryptosystem. Discrete applied Math. **128**(1), 207–221 (2003), international Workshop on Coding and Cryptography (WCC2001).
39. Overbeck, R.: A new structural attack for GPT and variants. In: Mycrypt. LNCS, vol. 3715, pp. 50–63 (2005)
40. Overbeck, R.: Structural attacks for public key cryptosystems based on Gabidulin codes. J. Cryptology **21**(2), 280–301 (2008)

41. Pham, B., Loidreau, P.: An analysis of Coggia-Couvreur attack on Loidreau's rank-metric public-key encryption scheme in the general case. In: WCC 2022 : Twelth International Workshop on Coding and Cryptography (2022), https://www.wcc2022.uni-rostock.de/storages/uni-rostock/Tagungen/WCC2022/Papers/WCC_2022_paper_38.pdf
42. Puchinger, S., Rosenkilde né Nielsen, J., Sheekey, J.: Further generalisations of twisted Gabidulin codes. In: WCC 2017 - Workshop on Coding Theory and Cryptography (2017), https://arxiv.org/abs/1703.08093
43. Puchinger, S., Renner, J., Wachter-Zeh, A.: Twisted Gabidulin codes in the GPT cryptosystem (2018), http://arxiv.org/abs/1806.10055, arXiv:1806.10055
44. Randrianarisoa, T.: A decoding algorithm for rank metric codes. preprint (2017), https://arxiv.org/abs/1712.07060
45. Randrianarisoa, T., Rosenthal, J.: A decoding algorithm for twisted Gabidulin codes. In: 2017 IEEE International Symposium on Information Theory. p. 2771–2774 (2017). https://doi.org/10.1109/ISIT.2017.8007034
46. Rashwan, H., Gabidulin, E., Honary, B.: Security of the GPT cryptosystem and its applications to cryptography. Security and Communication Networks **4**(8), 937–946 (2011)
47. Rónyai, L.: Computing the structure of finite algebras. J. Symbolic Comput. **9**(3), 355–373 (1990)
48. Sheekey, J.: A new family of linear maximum rank distance codes. Adv. Math. Commun. **10**(3), 475–488 (2016)