



**Politecnico  
di Torino**

**Politecnico di Torino**

Master's Degree in Communications Engineering

A.Y. 2025/2026

Graduation Session March 2026

# **A Framework for McEliece-Like Cryptosystems Based on Gabidulin Matrix Codes**

Supervisors:

Felicitas Hörmann  
Hannes Bartz  
Roberto Garelo

Candidate:

Alida Scribano

## Abstract

Code-based cryptography is one of the main candidates for post-quantum cryptography. In fact, its security relies on the hardness of decoding a random-looking linear code, a problem believed to remain difficult even for quantum computers. In particular, rank-metric code-based cryptosystems have attracted significant interest due to their ability to achieve smaller key sizes compared to classical Hamming-metric constructions.

Gabidulin codes, which are the rank-metric analogues of Reed–Solomon codes, play a central role among rank-metric codes thanks to their optimal error-correction capability. However, their strong algebraic structure makes them vulnerable to structural attacks, such as Overbeck’s attack, which can distinguish these codes from random ones and recover the hidden structure.

To address this issue, some recent cryptographic proposals rely on Gabidulin matrix codes, obtained by expanding Gabidulin codes defined over extension fields into equivalent representations over the base field. This transformation hides the extension-field linearity while preserving efficient decoding.

The main contribution of this thesis is the development of a general framework for cryptosystems based on Gabidulin matrix codes. This framework unifies and generalizes existing constructions, including both McEliece and Niederreiter variants, by combining base-field expansion and masking through random transformations. Additionally, the most common known structural attacks against Reed–Solomon and Gabidulin codes as well as their applicability to Gabidulin matrix codes are revisited.

# Acknowledgements

*First of all, I would like to thank my supervisors, Professor Roberto Garelo, Felicitas Hörmann, and Hannes Bartz, for giving me the opportunity to undertake this thesis experience, which has been an important moment of academic and personal growth.*

*A special thanks goes to Professor Roberto Garelo, an inspiring Professor who is able to recognize and encourage the potential of his students, and who is always present in supporting and encouraging them.*

*I would also like to thank Felicitas Hörmann for the kindness, patience, and clarity with which she guided me during this work. Her precision, together with the calm and positive attitude she always conveyed, made this experience particularly enjoyable.*

*Desidero poi ringraziare la mia famiglia — mia mamma, mio papà, mio fratello e mia nonna — per il loro sostegno costante e per avermi sempre fatto sentire il loro orgoglio e la loro fiducia in me.*

*Infine, un grazie speciale alle mie amiche, che mi hanno accompagnata lungo tutto questo percorso, facendomi spesso sentire più capace di quanto pensassi e offrendomi sempre ascolto e supporto nei momenti di confronto su università, lavoro e futuro.*

# Table of Contents

<b>List of Tables</b>	III
<b>1 Introduction</b>	1
1.1 Context and Motivation . . . . .	1
1.2 Brief Overview of Post-Quantum Cryptography . . . . .	2
1.3 Overview of Code-Based Cryptography . . . . .	3
<b>2 Notation and Preliminaries</b>	7
2.1 Basic Notation . . . . .	7
2.2 Linear Codes . . . . .	7
2.3 Hamming Metric and Reed-Solomon Codes . . . . .	8
<b>3 Gabidulin Codes and Gabidulin Matrix Codes</b>	10
3.1 Rank Metric and Gabidulin Codes . . . . .	10
3.2 Expanded Codes over the Base Field . . . . .	12
3.3 Gabidulin Matrix Codes and Expanded Gabidulin Codes . . . . .	18
<b>4 Algebraic Attacks</b>	25
4.1 Overbeck’s Attack . . . . .	25
4.2 Square-Code Attack . . . . .	29
<b>5 Cryptosystems Based on Gabidulin Matrix Codes</b>	34
5.1 McEliece Approach . . . . .	36
5.2 Niederreiter Approach . . . . .	41
5.3 Special Cases . . . . .	45
<b>6 Conclusion</b>	51
<b>Bibliography</b>	53

# List of Tables

4.1	Square-code dimensions for expanded Gabidulin codes. . . . .	32
4.2	Square-code dimensions for Gabidulin codes over $\mathbb{F}_{q^m}$ . . . . .	33
5.1	Parameter sets from [6] and [7]. Gray rows correspond to [6] ( $\ell_1 = \ell_2 = 0$ ), while blue rows correspond to [7] ( $s = 0$ ). . . . .	48

# Chapter 1

## Introduction

### 1.1 Context and Motivation

Error-correcting codes were originally developed and are commonly used in telecommunications to correct transmission errors. In this setting, a notion of distance, or metric, is required to quantify how much a received word differs from the transmitted one and to determine how many errors can be corrected. In classical coding theory, the most commonly used metric is the *Hamming metric*, which measures the number of positions in which two vectors differ.

In code-based cryptography, however, error-correcting codes are exploited in a different way: errors are deliberately introduced into the message to disrupt it, and only the holder of the private key can correct these errors to recover the original message. As in the telecommunication setting, the choice of a suitable metric plays a crucial role, as it directly affects the security properties of the resulting cryptosystem. A good choice, in this context, is the *rank metric*, so we talk about *rank-metric codes*. Indeed, using rank-metric codes allows achieving a comparable level of security with, in many cases, significantly smaller public keys than codes suited for the Hamming metric. Having smaller public keys is beneficial both for storage and for transmission.

Among rank-metric codes, *Gabidulin codes* have received considerable attention. Several cryptosystems have been proposed based on Gabidulin codes [1, 2, 3, 4, 5]; however, only a few (for instance, the one proposed in [6]) appear to be secure. In this thesis, we focus on studying this system: we aim to implement it and explore *distinguishers* to test a preliminary level of security. A *distinguisher* is an algorithm that aims to detect whether a public code is, in fact, not behaving randomly but like a specific structured code.

It is important to note that successfully distinguishing the code does not constitute a complete cryptanalysis, i.e., it does not immediately allow an attacker to recover the private key. Nevertheless, distinguishing the code can be considered a first step toward a potential attack. For example, in the case of Reed–Solomon codes, the existence of an efficient square-code distinguisher directly leads to secret-key recovery.

A major security concern for Gabidulin-based cryptosystems is the *Overbeck attack*. This attack exploits the specific  $\mathbb{F}_{q^m}$ -linear structure of Gabidulin codes, allowing an attacker not only to distinguish the code from a random one but also to recover the structured generator matrix of the Gabidulin code (the secret key). The key idea behind the system studied in [6] is to hide the  $\mathbb{F}_{q^m}$ -linear structure, rendering Overbeck’s attack ineffective.

Interestingly, a very recent proposal from 2024 [7] appears to introduce a cryptosystem similar to the one presented in 2017 [6]. Another goal of this thesis is to compare the two schemes, highlighting their similarities and differences, and investigating whether a formal connection exists between them, for example, whether one system can be reduced to the other.

## 1.2 Brief Overview of Post-Quantum Cryptography

Progress in quantum computing poses a serious threat to modern public-key cryptography, as sufficiently powerful quantum computers are expected to efficiently break many widely deployed asymmetric cryptosystems.

In response to this threat, the cryptographic community has started developing cryptographic schemes designed to remain secure even in the presence of quantum computers. This research area is known as *post-quantum cryptography*.

Following this, the National Institute of Standards and Technology (NIST) initiated a standardization process in 2016 for key-encapsulation mechanisms and digital signature schemes. This process has led to the selection of several candidate schemes, which are currently being standardized for practical deployment <sup>1</sup>.

Post-quantum cryptography includes several families of cryptographic schemes, each based on different mathematical problems believed to be hard even for quantum

---

<sup>1</sup>at present, this standardization is completed for key-encapsulation mechanisms, while it is still ongoing for digital signature schemes

computers. The main approaches include lattice-based cryptography, code-based cryptography, multivariate cryptography, hash-based cryptography, and isogeny-based cryptography.

### 1.3 Overview of Code-Based Cryptography

Code-based cryptography originates from the proposal of McEliece (1978)[8], which introduced a public-key encryption scheme whose security relies on the hardness of decoding random-looking linear error-correcting codes. Since then, many cryptosystems following the same principle have been developed, which together form part of a broader research area known as *code-based cryptography*.

This research area is particularly relevant today, as decoding linear codes that appear random (i.e., for which no exploitable algebraic structure is known) remains computationally hard even in the presence of quantum computers, making code-based schemes strong candidates for *post-quantum cryptography*.

Despite their long history, McEliece-type systems still present two major challenges:

1. the **public keys are extremely large**;
2. there is **no known reduction** proving that breaking these systems is equivalent to solving a well-studied hard problem such as Syndrome Decoding on random codes.

Another important consideration is the **trade-off** between decoding efficiency and the ability to hide the structure of the code. In fact, in code-based cryptosystems, increasing the amount of algebraic structure typically leads to more efficient decoding algorithms, but at the cost of making the underlying structure harder to conceal from an adversary.

**Large public keys.** McEliece-like cryptosystems are known for having relatively large public keys. In particular, in Classic McEliece they range from hundreds of kilobytes to over a megabyte; Such large public keys pose practical challenges:

- they require significant storage,
- they must be transmitted at high communication cost.

In contrast, traditional number-theoretic systems such as RSA require public keys of only a few kilobytes.

**Lack of a reduction to a canonical hard problem.** A central goal in cryptography is to justify the security of a scheme by reducing it to a problem widely believed to be hard. For code-based cryptosystems, the natural candidate is the *Syndrome Decoding Problem* (SDP). Given a parity-check matrix  $\mathbf{H}$  and a syndrome  $\mathbf{s}$ , the problem consists in finding an error vector  $\mathbf{e}$  of bounded Hamming weight such that  $\mathbf{H}\mathbf{e}^T = \mathbf{s}$ . This problem, with random linear codes, is NP-hard in general. A random linear code is obtained by selecting the entries of a generator or parity-check matrix uniformly at random over the underlying field. McEliece-like systems, however, don't use random codes; instead, they, usually, employ highly structured codes which do not behave like random codes. As a result, no proof is known that breaking McEliece-like systems is equivalent to solving Syndrome Decoding on random codes, and the security of the scheme relies on an unproven but well-established hardness assumption.

A nice strategy to reduce public-key sizes is to employ the *rank metric*, which was first applied to cryptography by Gabidulin, Paramonov, and Tretjakov in 1991 [1]. The best known attacks against generic, rank-metric decoding (i.e., decoding a random code) are significantly more expensive than those for the Hamming metric [9], which enables the use of much smaller key sizes.

Numerous McEliece-like cryptosystems based on Gabidulin codes, which are the rank-metric analogue of Reed-Solomon codes, have therefore been proposed. The first such proposal was the GPT cryptosystem, introduced by Gabidulin, Paramonov, and Tretjakov in 1991 [1]. Later works proposed several variants and modifications of the original GPT system [10, 11].

Most of these schemes attempt to mask the structure of Gabidulin codes through various distortion techniques. Unfortunately, the *Overbeck attack* [12] broke most of them, for example [1, 11], exploiting the action of the Frobenius map to distinguish a Gabidulin-based public key from a random one.

This Frobenius-based distinguisher plays the same role in the rank metric as the *square-code* distinguisher does for Reed-Solomon codes [13, 14]. The presence of these distinguishers remains a major obstacle to the secure use of Gabidulin codes in McEliece-type constructions.

One masking technique that still remains resistant to structural attacks is the one used in the original Classic McEliece scheme. This approach uses Goppa codes and can be interpreted as taking generalized Reed-Solomon codes over an extension field  $\mathbb{F}_{2^m}$  and then considering their  $\mathbb{F}_2$ -subcodes. Passing to the  $\mathbb{F}_2$ -subcode destroys the structure over  $\mathbb{F}_{2^m}$ , thereby preventing the use of a Reed-Solomon distinguisher, except in situations where the code rate is close to 1, which is typically not the case (see also [15]).

Loidreau proposed an approach based on mixing Gabidulin codes with LRPC (Low Rank Parity Check) codes [10, 16] which seems to sufficiently hide the Gabidulin structure. The resulting public keys are not as small as in the original GPT system, which reduced key sizes by a factor of 100 compared to Classic McEliece, but they are still about ten times smaller than the Classic McEliece keys.

For completeness, we also mention that code-based cryptography includes a second major design paradigm, different from the McEliece approach discussed in this thesis. This alternative line of work originates from the Alekhnovich approach [17] and its variants, such as RQC [18] and HQC [19]. In these constructions no structural masking is used: encryption relies on a random (or random quasi-cyclic) code, while decryption employs a different code with an efficient decoding algorithm. The main advantage of this approach is that it avoids structural attacks, and is therefore conceptually stronger from a security point of view.

The main drawback, however, is that ciphertexts are typically much larger, often quadratic in the security parameter, whereas McEliece-type schemes achieve ciphertexts of linear size at the price of a larger public key. For example, for a 128-bit security level, Classic McEliece yields ciphertexts of roughly 100 bytes, whereas HQC produces ciphertexts of about 5 kilobytes. For applications in which having very small ciphertexts is critical, McEliece-style constructions may therefore be more attractive.

It is also worth mentioning the *Niederreiter variant*, which is a dual formulation of the McEliece encryption scheme. In this variant, the public key is given by a parity-check matrix instead of a generator matrix, and encryption amounts to syndrome computation. Both formulations are equivalent in terms of security.

**Thesis structure.** The remainder of this thesis is organized as follows.

In Chapter 2 we introduce the notation and review the main concepts from coding theory that are needed throughout the thesis. In particular, we recall the definition of linear codes, the Hamming metric, and Reed–Solomon codes.

Chapter 3 focuses on rank-metric codes and, in particular, on Gabidulin codes. After recalling the rank metric and the algebraic notions required to define Gabidulin codes, we study their expansion over the base field. This leads to the notions of Gabidulin matrix codes and expanded Gabidulin codes.

In Chapter 4 we review two important algebraic attacks against structured codes, namely Overbeck’s attack and the square-code attack.

Chapter 5 presents a general framework for cryptosystems based on Gabidulin matrix codes. The framework unifies and extends existing constructions, including McEliece and Niederreiter variants, and highlights the relationships among different

proposals in the literature.

Finally, Chapter 6 summarizes the main results and discusses some open problems.

# Chapter 2

## Notation and Preliminaries

### 2.1 Basic Notation

We denote by  $\mathbb{F}_q$  a finite field of size  $q$ , where  $q$  is a prime power.

The set of row vectors of length  $n$  over  $\mathbb{F}_q$  is denoted by  $\mathbb{F}_q^n$ , and the set of matrices of size  $m \times n$  over  $\mathbb{F}_q$  is denoted by  $\mathbb{F}_q^{m \times n}$ .

Matrices and vectors are denoted by bold uppercase and by lowercase letters, respectively, e.g.,  $\mathbf{A} \in \mathbb{F}_q^{m \times n}$  and  $\mathbf{a} \in \mathbb{F}_q^n$ .

The set of integers  $\{i \mid a \leq i \leq b\}$  is denoted by  $[a, b]$ .

The entries of a matrix  $\mathbf{A} \in \mathbb{F}_q^{m \times n}$  are denoted  $A_{i,j}$  for  $i \in [1, m]$ ,  $j \in [1, n]$ , and the entries of a vector  $\mathbf{a} \in \mathbb{F}_q^n$  are denoted  $a_i$  for  $i \in [1, n]$ .

The *General Linear Group* of the  $m \times m$  invertible matrices with entries in  $\mathbb{F}_q$  is denoted by  $GL(m, q)$ .

The identity matrix and the all-zero matrix of size  $n$  are denoted, respectively, by  $\mathbf{I}_n$  and  $\mathbf{0}_n$ .

### 2.2 Linear Codes

A linear code  $\mathcal{C} \subseteq \mathbb{F}_q^n$  is a (sub)vectorspace (so a linear space) of  $\mathbb{F}_q^n$ . The elements of  $\mathcal{C}$  are called *codewords*. The *dimension* of  $\mathcal{C}$  is its dimension as a vector space over  $\mathbb{F}_q$ , i.e.,  $k = \dim_q(\mathcal{C})$ .

The parameter  $n$  is called the *length* of the code and corresponds to the number of coordinates of its codewords. For a linear code of length  $n$ , and dimension  $k$  over  $\mathbb{F}_q$  we write  $[n, k]_q$ .

**Dual code.** For two vectors  $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$ , let  $\langle \mathbf{a}, \mathbf{b} \rangle \stackrel{\text{def}}{=} \sum_{i=1}^n a_i b_i$  define the Euclidean inner product and let  $\mathcal{C}$  be a linear  $[n, k]_q$  code. Then, the set of vectors

$$\mathcal{C}^\perp := \{ \mathbf{c}^\perp \in \mathbb{F}_q^n : \langle \mathbf{c}^\perp, \mathbf{c} \rangle = 0, \forall \mathbf{c} \in \mathcal{C} \}$$

is called the *dual code* of  $\mathcal{C}$ .

**Square code.** For two vectors  $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$ , let  $\mathbf{a} * \mathbf{b} := (a_1 b_1, \dots, a_n b_n)$  define the Schur product, i.e., the coordinatewise product of  $\mathbf{a}$  and  $\mathbf{b}$ .

With this definition we can also define the Schur product of two linear codes.

For two linear codes  $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}_q^n$  the Schur product of  $\mathcal{C}_1$  and  $\mathcal{C}_2$  is defined as the  $\mathbb{F}_q$ -span generated by the Schur product of all combinations of codewords, i.e.,

$$\mathcal{C}_1 * \mathcal{C}_2 := \langle \{ \mathbf{c}_1 * \mathbf{c}_2 : \mathbf{c}_1 \in \mathcal{C}_1, \mathbf{c}_2 \in \mathcal{C}_2 \} \rangle_q \subseteq \mathbb{F}_q^n.$$

For a linear code  $\mathcal{C} \subseteq \mathbb{F}_q^n$ , we define its *square code* as

$$\mathcal{C}^{*2} := \langle \{ \mathbf{c}_1 * \mathbf{c}_2 : \mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C} \} \rangle_q \subseteq \mathbb{F}_q^n.$$

## 2.3 Hamming Metric and Reed-Solomon Codes

In order to quantify the distance between two vectors (codewords, in particular), we equip the vector space  $\mathbb{F}_q^n$  with a metric; the most commonly used metric in this context is the *Hamming metric*.

For a vector  $\mathbf{a} \in \mathbb{F}_q^n$ , the *Hamming weight*  $\text{wt}_H(\mathbf{a})$  is the number of non-zero coordinates of  $\mathbf{a}$ .

For two vectors  $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$ , the *Hamming distance*  $d_H(\mathbf{a}, \mathbf{b})$  is the number of coordinates in which  $\mathbf{a}$  and  $\mathbf{b}$  differ.

The Hamming distance can be expressed in terms of the Hamming weight, since it coincides with the weight of the difference of two vectors, namely

$$d_H(\mathbf{a}, \mathbf{b}) = \text{wt}_H(\mathbf{a} - \mathbf{b}).$$

Once a metric is fixed, one can associate to a code  $\mathcal{C} \subseteq \mathbb{F}_q^n$  an important parameter, called the *minimum Hamming distance*, which measures the smallest distance between any two distinct codewords. It is defined as

$$\begin{aligned} d_H(\mathcal{C}) &:= \min\{ d_H(\mathbf{a}, \mathbf{b}) : \mathbf{a}, \mathbf{b} \in \mathcal{C}, \mathbf{a} \neq \mathbf{b} \} \\ &= \min\{ \text{wt}_H(\mathbf{a}) : \mathbf{a} \in \mathcal{C}, \mathbf{a} \neq \mathbf{0} \}. \end{aligned}$$

A code with minimum distance  $d_H(\mathcal{C})$  can correct up to

$$t = \left\lfloor \frac{d_H(\mathcal{C}) - 1}{2} \right\rfloor$$

errors. The integer  $t$  is called the *error-correction capability* of the code.

One important family of codes in the Hamming metric are the *Reed-Solomon codes*, which we now define.

A linear Reed-Solomon code over  $\mathbb{F}_q$  of length  $n$ , with  $k \leq n \leq q$ , and dimension  $k$ , denoted by  $\mathcal{RS}[\mathbf{g}; n, k]_q$ , is defined as follows

$$\mathcal{RS}[\mathbf{g}; n, k]_q := \left\{ (f(g_1), \dots, f(g_n)) = f(\mathbf{g}) : f(x) \in \mathbb{F}_q[x]_{<k} \right\} \subseteq \mathbb{F}_q^n$$

where  $\mathbf{g} = (g_1, \dots, g_n) \in \mathbb{F}_q^n$  with distinct elements and  $\mathbb{F}_q[x]_{<k}$  is the set of all polynomials over  $\mathbb{F}_q$  with degree less than  $k$ .

We have seen the Hamming weight and Hamming distance, which are suited for classical codes like Reed-Solomon codes. Similarly, one can define another metric, called the *rank metric*, which is used for *rank-metric codes* such as *Gabidulin codes*.

## Chapter 3

# Gabidulin Codes and Gabidulin Matrix Codes

### 3.1 Rank Metric and Gabidulin Codes

Let  $\mathbb{F}_{q^m}$  denote an extension field of  $\mathbb{F}_q$  of degree  $m$ .

Let  $\mathbf{a}, \mathbf{b} \in \mathbb{F}_{q^m}^n$ . The *rank weight* of  $\mathbf{a}$  is defined as the dimension of the  $\mathbb{F}_q$ -vector space generated by its entries, that is,

$$\text{wt}_R(\mathbf{a}) := \dim_q \langle a_1, \dots, a_n \rangle_q. \quad (3.1)$$

The *rank distance* between two vectors  $\mathbf{a}$  and  $\mathbf{b}$  is then defined as

$$d_R(\mathbf{a}, \mathbf{b}) := \text{wt}_R(\mathbf{a} - \mathbf{b}). \quad (3.2)$$

For a linear code  $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$ , its *minimum rank distance* is given by

$$\begin{aligned} d_R(\mathcal{C}) &:= \min\{d_R(\mathbf{a}, \mathbf{b}) : \mathbf{a}, \mathbf{b} \in \mathcal{C}, \mathbf{a} \neq \mathbf{b}\} \\ &= \min\{\text{wt}_R(\mathbf{a}) : \mathbf{a} \in \mathcal{C}, \mathbf{a} \neq \mathbf{0}\}. \end{aligned}$$

Before introducing Gabidulin codes, we recall the notions of *linearized polynomials* and  *$q$ -Vandermonde matrices*. We also introduce the *Frobenius automorphism*, which is used to define linearized polynomials.

We define an automorphism of the field  $\mathbb{F}_{q^m}$  as a bijective mapping  $\phi : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_{q^m}$ , such that

$$\phi(a) + \phi(b) = \phi(a + b) \quad \forall a, b \in \mathbb{F}_{q^m} \quad (\text{i})$$

$$\phi(a) \cdot \phi(b) = \phi(a \cdot b) \quad \forall a, b \in \mathbb{F}_{q^m} \quad (\text{ii})$$

The Galois group of the field extension is denoted by  $\text{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q)$ , and consists of all automorphisms  $\sigma$  of  $\mathbb{F}_{q^m}$  that fix the base field  $\mathbb{F}_q$ , i.e.,  $\sigma(a) = a$  for all  $a \in \mathbb{F}_q$ . For finite fields, the Galois group consists of all powers of the *Frobenius automorphism*

$$\phi_q : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_{q^m}, \quad a \mapsto a^q,$$

i.e.,

$$\text{Gal}(\mathbb{F}_{q^m}/\mathbb{F}_q) = \{\phi_q^i : 0 \leq i \leq m-1\}.$$

Indeed, every element  $a \in \mathbb{F}_{q^m}$  satisfies  $a^{q^m} = a$ , and hence  $\phi_q^m = \text{id}$ . Therefore, the Frobenius automorphism has order  $m$  and its distinct powers are exactly  $\phi_q^0, \dots, \phi_q^{m-1}$ .

A polynomial  $a(x)$  is called a *linearized polynomial* if it has the form

$$a(x) = \sum_{i=0}^{d_a} a_i x^{[i]} = \sum_{i=0}^{d_a} a_i x^{q^i}, \quad a_i \in \mathbb{F}_{q^m} \quad \forall i \in [0, d_a],$$

where  $x^{[i]} = x^{q^i}$  denotes the element obtained by applying the Frobenius automorphism  $i$  times to  $x$ .

The non-commutative univariate linearized polynomial ring with indeterminate  $x$ , consisting of all such polynomials over  $\mathbb{F}_{q^m}$ , is denoted by  $\mathbb{L}_{q^m}[x]$ .

The  $q$ -degree of  $a(x)$  is defined to be the largest  $i \in [0, d_a]$  such that  $a_i \neq 0$ .

For a vector  $\mathbf{a} = (a_1, \dots, a_n) \in \mathbb{F}_{q^m}^n$ , we obtain the  $s \times n$   *$q$ -Vandermonde matrix*  $\text{van}_{s,q}$ :

$$\begin{aligned} \text{van}_{s,q} : \quad \mathbb{F}_{q^m}^n &\longrightarrow \mathbb{F}_{q^m}^{s \times n} \\ \mathbf{a} = (a_1, \dots, a_n) &\longmapsto \text{van}_{s,q}(\mathbf{a}) := \begin{pmatrix} a_1 & \cdots & a_n \\ a_1^q & \cdots & a_n^q \\ \vdots & \ddots & \vdots \\ a_1^{q^{s-1}} & \cdots & a_n^{q^{s-1}} \end{pmatrix}. \end{aligned}$$

A linear Gabidulin code over  $\mathbb{F}_{q^m}$  of length  $n \leq m$  and dimension  $k$ , denoted by  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$ , is defined by its  $k \times n$  generator matrix

$$\mathbf{G} = \text{van}_{k,q}(\mathbf{g}),$$

where  $\mathbf{g} = (g_1, \dots, g_n) \in \mathbb{F}_{q^m}^n$  and  $\text{wt}_R(\mathbf{g}) = n$ .

It was shown by Gabidulin [20] that Gabidulin codes are MRD (Maximum Rank-Distance) codes, that is, they reach the largest possible minimum rank distance

among all linear codes with the same length and dimension.

More precisely, the Singleton bound in the rank metric states that for a linear code over  $\mathbb{F}_{q^m}$  of length  $n \leq m$  and dimension  $k$ , the minimum rank distance satisfies

$$d \leq n - k + 1.$$

Gabidulin codes meet this bound with equality, and therefore have minimum rank distance  $d = n - k + 1$ .

Achieving the Singleton bound is optimal, since a larger minimum rank distance directly implies a higher error-correction capability. In particular, an MRD code can correct up to

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor = \left\lfloor \frac{n-k}{2} \right\rfloor$$

rank errors, which is the maximum possible for the given parameters.

Equivalently, we can define Gabidulin codes by evaluating  $q$ -degree restricted linearized polynomials:

$$\mathcal{G}[\mathbf{g}; n, k]_{q^m} := \left\{ (f(g_1), \dots, f(g_n)) = f(\mathbf{g}) : f(x) \in \mathbb{L}_{q^m}[x]_{<k} \right\},$$

where the fixed elements  $g_1, \dots, g_n \in \mathbb{F}_{q^m}$  are linearly independent over  $\mathbb{F}_q$  and  $\mathbb{L}_{q^m}[x]_{<k}$  is the set of all linearized polynomials with  $q$ -degree less than  $k$ .

Let  $h_1, \dots, h_n \in \mathbb{F}_{q^m}$  be a non-zero solution of the following system of  $n - 1$  linear equations:

$$\sum_{i=1}^n g_i^{[j]} h_i = 0, \quad \forall j \in [-n + k + 1, k - 1].$$

Then, the  $(n - k) \times n$  matrix

$$\mathbf{H} := \text{van}_{n-k,q}((h_1, \dots, h_n)) = \begin{pmatrix} h_1^{[0]} & \dots & h_n^{[0]} \\ h_1^{[1]} & \dots & h_n^{[1]} \\ \vdots & \ddots & \vdots \\ h_1^{[n-k-1]} & \dots & h_n^{[n-k-1]} \end{pmatrix},$$

is a parity-check matrix of the  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$  code.

## 3.2 Expanded Codes over the Base Field

Let  $\mathcal{B} = (b_1, \dots, b_m)$  be a basis of  $\mathbb{F}_{q^m}$  over  $\mathbb{F}_q$  ( $\mathbb{F}_{q^m}/\mathbb{F}_q$ ), i.e., the elements  $b_1, \dots, b_m \in \mathbb{F}_{q^m}$  are linearly independent over  $\mathbb{F}_q$ , implying that every element

of  $\mathbb{F}_{q^m}$  can be written uniquely as a linear combination  $a_1b_1 + \cdots + a_mb_m$  with coefficients  $a_i \in \mathbb{F}_q$ .

We define a mapping from  $\mathbb{F}_{q^m}$  to  $\mathbb{F}_q^m$  by

$$\begin{aligned} \phi_{\mathcal{B}} : \mathbb{F}_{q^m} &\longrightarrow \mathbb{F}_q^m \\ x &\longmapsto \mathbf{x} = (x_1, \dots, x_m) \\ \text{s.t.} \quad x &= \sum_{i=1}^m x_i b_i \end{aligned}$$

analogously, we define the generalized mappings, from  $\mathbb{F}_{q^m}^n$  to  $\mathbb{F}_q^{m \times n}$  and, respectively, from  $\mathbb{F}_{q^m}^n$  to  $\mathbb{F}_q^{mn}$ , by

$$\begin{aligned} \text{ext}_{\mathcal{B}} : \quad \mathbb{F}_{q^m}^n &\longrightarrow \mathbb{F}_q^{m \times n} \\ \mathbf{y} = (y_1, \dots, y_n) &\longmapsto \begin{pmatrix} Y_{1,1} & \cdots & Y_{1,n} \\ \vdots & \ddots & \vdots \\ Y_{m,1} & \cdots & Y_{m,n} \end{pmatrix} \\ \\ \Phi_{\mathcal{B}} : \quad \mathbb{F}_{q^m}^n &\longrightarrow \mathbb{F}_q^{mn} \\ \mathbf{y} = (y_1, \dots, y_n) &\longmapsto (Y_{1,1}, \dots, Y_{m,1}, \dots, Y_{1,n}, \dots, Y_{m,n}) \\ \text{s.t.} \quad y_j &= \sum_{i=1}^m Y_{i,j} b_i, \quad \forall j \in [1, n] \end{aligned}$$

From (3.1) it follows,

$$\text{wt}_R(\mathbf{a}) = \text{rank} \left( \text{ext}_{\mathcal{B}}(\mathbf{a}) \right), \quad \mathbf{a} \in \mathbb{F}_{q^m}^n. \quad (3.3)$$

From (3.2) and (3.3) it follows,

$$d_R(\mathbf{a}, \mathbf{b}) = \text{rank} \left( \text{ext}_{\mathcal{B}}(\mathbf{a}) - \text{ext}_{\mathcal{B}}(\mathbf{b}) \right), \quad \mathbf{a}, \mathbf{b} \in \mathbb{F}_{q^m}^n.$$

Moreover, we define a mapping **Fold** which turns a vector

$$\mathbf{a} = (a_1, \dots, a_m, a_{m+1}, \dots, a_{2m}, \dots, a_{m(n-1)+1}, \dots, a_{mn}) \in \mathbb{F}_q^{mn}$$

into the matrix

$$\mathbf{Fold}(\mathbf{a}) := \begin{pmatrix} a_1 & a_{m+1} & \cdots & a_{m(n-1)+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_m & a_{2m} & \cdots & a_{mn} \end{pmatrix} \in \mathbb{F}_q^{m \times n}$$

obtained by grouping the coordinates of  $\mathbf{a}$  into  $n$  consecutive blocks of length  $m$  and using each block as a column of the matrix.

The inverse mapping is defined as well by **Unfold** which turns a matrix

$$\mathbf{A} = \begin{pmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,n} \end{pmatrix} \in \mathbb{F}_q^{m \times n}$$

into a vector

$$\mathbf{Unfold}(\mathbf{A}) := (A_{1,1}, \dots, A_{m,1}, \dots, A_{1,n}, \dots, A_{m,n}) \in \mathbb{F}_q^{mn}.$$

An  $\mathbb{F}_q$ -linear subspace of  $\mathbb{F}_q^{m \times n}$  is called a *matrix code*.

Let  $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$  be a  $[n, k]_{q^m}$  code and let  $\mathcal{B}$  be a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ .

Applying the map  $\text{ext}_{\mathcal{B}}$  componentwise to the codewords of  $\mathcal{C}$ , we obtain a matrix code over  $\mathbb{F}_q$ , called the *matrix expansion* of  $\mathcal{C}$  with respect to the basis  $\mathcal{B}$ , defined as

$$\text{ext}_{\mathcal{B}}(\mathcal{C}) := \{\text{ext}_{\mathcal{B}}(\mathbf{c}) \in \mathbb{F}_q^{m \times n} : \mathbf{c} \in \mathcal{C}\} \subseteq \mathbb{F}_q^{m \times n}.$$

Similarly, applying the map  $\Phi_{\mathcal{B}}$  to the codewords of  $\mathcal{C}$ , we obtain a vector code over  $\mathbb{F}_q$ , called the *vector expansion* of  $\mathcal{C}$  with respect to the basis  $\mathcal{B}$ , defined as

$$\Phi_{\mathcal{B}}(\mathcal{C}) := \{\Phi_{\mathcal{B}}(\mathbf{c}) \in \mathbb{F}_q^{mn} : \mathbf{c} \in \mathcal{C}\} \subseteq \mathbb{F}_q^{mn}.$$

### Relation between matrix and vector expansion.

The matrix expansion  $\text{ext}_{\mathcal{B}}(\mathcal{C})$  and the vector expansion  $\Phi_{\mathcal{B}}(\mathcal{C})$  are in bijective correspondence.

More precisely, they are related through the bijective maps **Fold** and **Unfold**.

In particular, we have

$$\text{ext}_{\mathcal{B}}(\mathcal{C}) = \mathbf{Fold}(\Phi_{\mathcal{B}}(\mathcal{C})) = \{\mathbf{Fold}(\Phi_{\mathcal{B}}(\mathbf{c})) \in \mathbb{F}_q^{m \times n} : \mathbf{c} \in \mathcal{C}\},$$

and

$$\Phi_{\mathcal{B}}(\mathcal{C}) = \mathbf{Unfold}(\text{ext}_{\mathcal{B}}(\mathcal{C})) = \{\mathbf{Unfold}(\text{ext}_{\mathcal{B}}(\mathbf{c})) \in \mathbb{F}_q^{mn} : \mathbf{c} \in \mathcal{C}\}.$$

Since the map  $\phi_{\mathcal{B}}$  is an isomorphism of  $\mathbb{F}_q$ -vector spaces, the associated matrix and vector expansions are invertible whenever the basis  $\mathcal{B}$  is known.

Conversely, when only the matrix expansion or the vector expansion is given, the corresponding code over  $\mathbb{F}_{q^m}$  cannot be recovered without knowledge of the basis  $\mathcal{B}$ . In particular, it is still  $\mathbb{F}_{q^m}$ -linear, but the  $\mathbb{F}_{q^m}$ -linear structure is hard to recover without knowledge of the basis  $\mathcal{B}$ .

## Generator Matrix of the Vector Expansion of an $\mathbb{F}_{q^m}$ -Linear Code

Given a generator matrix of an  $\mathbb{F}_{q^m}$ -linear code, we now show a way to compute a generator matrix of its vector expansion.

Before proceeding, let us define a mapping from  $\mathbb{F}_{q^m}$  to  $\mathbb{F}_{q^m}$  by

$$\begin{aligned} \sigma_\alpha : \mathbb{F}_{q^m} &\longrightarrow \mathbb{F}_{q^m} \\ x &\longmapsto \alpha x \end{aligned}$$

where  $\alpha \in \mathbb{F}_{q^m}$ .

The following diagram illustrates the relation between multiplication by  $\alpha$  in  $\mathbb{F}_{q^m}$  and its matrix representation over  $\mathbb{F}_q$ . In particular, the diagram is commutative, meaning that both paths from  $\mathbb{F}_{q^m}$  to  $\mathbb{F}_q^m$  yield the same result.

$$\begin{array}{ccc} \mathbb{F}_{q^m} & \xrightarrow{\sigma_\alpha} & \mathbb{F}_{q^m} \\ \downarrow \phi_{\mathcal{B}} & & \downarrow \phi_{\mathcal{B}} \\ \mathbb{F}_q^m & \xrightarrow{\mathbf{M}_\alpha} & \mathbb{F}_q^m \end{array}$$

**Figure 3.1:** Matrix representation over  $\mathbb{F}_q$  of the multiplication-by- $\alpha$  map with respect to the basis  $\mathcal{B}$

In particular, for any  $x \in \mathbb{F}_{q^m}$ ,

$$\phi_{\mathcal{B}}(x)\mathbf{M}_\alpha = \phi_{\mathcal{B}}(\alpha x).$$

Once a basis  $\mathcal{B} = (b_1, \dots, b_m)$  of  $\mathbb{F}_{q^m}/\mathbb{F}_q$  is fixed, the matrix  $\mathbf{M}_\alpha \in \mathbb{F}_q^{m \times m}$  can be written explicitly as

$$\mathbf{M}_\alpha := \left( \begin{array}{c|c|c} & & \\ \phi_{\mathcal{B}}(\sigma_\alpha(b_1))^\top & \dots & \phi_{\mathcal{B}}(\sigma_\alpha(b_m))^\top \\ & & \end{array} \right) \in \mathbb{F}_q^{m \times m} \quad (3.4)$$

**Lemma 1.** *The matrix  $\mathbf{M}_\alpha$  defined in (3.4) satisfies*

$$\phi_{\mathcal{B}}(x)\mathbf{M}_\alpha = \phi_{\mathcal{B}}(\alpha x)$$

for every  $x \in \mathbb{F}_{q^m}$ .

*Proof.* Given  $\mathcal{B} = (b_1, \dots, b_m)$  a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ , let  $x \in \mathbb{F}_{q^m}$  and  $\alpha \in \mathbb{F}_{q^m}$ . By definition of the map  $\phi_{\mathcal{B}}$ ,

$$\phi_{\mathcal{B}}(x) = (x_1, \dots, x_m), \quad x = \sum_{j=1}^m x_j b_j, \quad x_j \in \mathbb{F}_q \quad \forall j \in [1, m].$$

We compute

$$\alpha x = \alpha \sum_{j=1}^m x_j b_j = \sum_{j=1}^m x_j \alpha b_j.$$

Since  $\alpha b_j \in \mathbb{F}_{q^m}$  for every  $j \in [1, m]$ , we can express it in the basis  $\mathcal{B}$  as

$$\alpha b_j = \sum_{\ell=1}^m m_{\ell j} b_{\ell}, \quad m_{\ell j} \in \mathbb{F}_q, \quad \forall \ell, j \in [1, m].$$

Equivalently,

$$\phi_{\mathcal{B}}(\alpha b_j) = (m_{1j}, \dots, m_{mj}).$$

Substituting into the expression of  $\alpha x$ , we obtain

$$\alpha x = \sum_{j=1}^m x_j \left( \sum_{\ell=1}^m m_{\ell j} b_{\ell} \right) = \sum_{\ell=1}^m \left( \sum_{j=1}^m x_j m_{\ell j} \right) b_{\ell}.$$

By inspecting the right-hand side of the last expression and recalling the definition of the map  $\phi_{\mathcal{B}}$ , we observe that

$$\sum_{j=1}^m x_j m_{1j}, \dots, \sum_{j=1}^m x_j m_{mj}$$

are precisely the coordinates of  $\phi_{\mathcal{B}}(\alpha x)$  with respect to the basis  $\mathcal{B}$ .

$$\phi_{\mathcal{B}}(\alpha x) = \left( \sum_{j=1}^m x_j m_{1j}, \dots, \sum_{j=1}^m x_j m_{mj} \right).$$

Hence,

$$\phi_{\mathcal{B}}(\alpha x) = (x_1, \dots, x_m) \begin{pmatrix} m_{11} & \dots & m_{1m} \\ \vdots & \ddots & \vdots \\ m_{m1} & \dots & m_{mm} \end{pmatrix} = \phi_{\mathcal{B}}(x) \mathbf{M}_{\alpha}.$$

Finally,

$$\begin{aligned}
 \mathbf{M}_\alpha &= \begin{pmatrix} m_{11} & \dots & m_{1m} \\ \vdots & \ddots & \vdots \\ m_{m1} & \dots & m_{mm} \end{pmatrix} \\
 &= \begin{pmatrix} \phi_{\mathcal{B}}(\alpha b_1)^\top & \dots & \phi_{\mathcal{B}}(\alpha b_m)^\top \\ \vdots & & \vdots \\ \phi_{\mathcal{B}}(\sigma_\alpha(b_1))^\top & \dots & \phi_{\mathcal{B}}(\sigma_\alpha(b_m))^\top \end{pmatrix} \\
 &= \begin{pmatrix} \phi_{\mathcal{B}}(\sigma_\alpha(b_1))^\top & \dots & \phi_{\mathcal{B}}(\sigma_\alpha(b_m))^\top \\ \vdots & & \vdots \\ \phi_{\mathcal{B}}(\sigma_\alpha(b_1))^\top & \dots & \phi_{\mathcal{B}}(\sigma_\alpha(b_m))^\top \end{pmatrix} \in \mathbb{F}_q^{m \times m}.
 \end{aligned}$$

□

**Lemma 2.** Let  $\alpha \in \mathbb{F}_{q^m}$  be a primitive element, and let

$$p(x) = x^m + a_{m-1}x^{m-1} + \dots + a_0 \in \mathbb{F}_q[x]$$

be its minimal polynomial over  $\mathbb{F}_q$ . Consider the multiplicative basis

$$\mathcal{B} = (1, \alpha, \alpha^2, \dots, \alpha^{m-1})$$

of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ . Then the matrix representation of  $\sigma_\alpha$  with respect to  $\mathcal{B}$ , namely  $\mathbf{M}_\alpha$ , is the companion matrix of  $p(x)$ .

*Proof.* By definition of the minimal polynomial, we have

$$p(\alpha) = \alpha^m + a_{m-1}\alpha^{m-1} + \dots + a_0 = 0,$$

so that

$$\alpha^m = -a_{m-1}\alpha^{m-1} - \dots - a_0.$$

Let  $\mathbf{M}_\alpha$  be the matrix representation of  $\sigma_\alpha$  with respect to the basis  $\mathcal{B}$ . By construction, the  $j$ -th column of  $\mathbf{M}_\alpha$  is  $\phi_{\mathcal{B}}(\sigma_\alpha(\alpha^{j-1})) = \phi_{\mathcal{B}}(\alpha^j)$ .

- For  $0 \leq j \leq m-2$ ,  $\alpha^j$  is mapped to  $\alpha^{j+1}$ , which is the next basis element. This is reflected by the fact that the first  $m-1$  columns of  $\mathbf{M}_\alpha$  contain a single 1 on the subdiagonal and 0 elsewhere.
- For  $j = m-1$ ,  $\alpha^{m-1}$  is mapped to  $\alpha^m$ , which is expressed as a linear combination of the basis elements using the minimal polynomial:

$$\sigma_\alpha(\alpha^{m-1}) = \alpha^m = -a_{m-1}\alpha^{m-1} - \dots - a_0.$$

Therefore, the last column of  $\mathbf{M}_\alpha$  is  $(-a_0, -a_1, \dots, -a_{m-1})^T$ .

Combining these observations, we obtain

$$\mathbf{M}_\alpha = \begin{pmatrix} 0 & 0 & \dots & 0 & -a_0 \\ 1 & 0 & \dots & 0 & -a_1 \\ 0 & 1 & \dots & 0 & -a_2 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & 1 & -a_{m-1} \end{pmatrix},$$

which is precisely the companion matrix of  $p(x)$ .  $\square$

Now that we have what we need, it is straightforward to describe a method to compute a generator matrix of the vector expansion of an  $\mathbb{F}_{q^m}$ -linear code.

Let  $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$  be a linear code,  $\mathcal{B} = (b_1, \dots, b_m)$  a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ ,  $\mathbf{G} = (G_{i,j}) \in \mathbb{F}_{q^m}^{k \times n}$  be a generator matrix of  $\mathcal{C}$ .

Then the matrix

$$\mathbf{G}_q := (\mathbf{M}_{G_{i,j}}) \in \mathbb{F}_q^{mk \times mn},$$

obtained by replacing each entry  $G_{i,j}$  of  $\mathbf{G}$  with its associated matrix  $\mathbf{M}_{G_{i,j}}$ , is a generator matrix of  $\Phi_{\mathcal{B}}(\mathcal{C})$ , where each entry  $G_{i,j}$  plays the role of  $\alpha$  in (3.4).

### 3.3 Gabidulin Matrix Codes and Expanded Gabidulin Codes

Let  $\mathcal{G} = \mathcal{G}[g; n, k]_{q^m} \subseteq \mathbb{F}_{q^m}^n$  be a Gabidulin code and let  $\mathcal{B}$  be a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ .

As introduced in Section 3.2, starting from an  $\mathbb{F}_{q^m}$ -linear code it is possible to construct codes over  $\mathbb{F}_q$  by applying the matrix expansion  $\text{ext}_{\mathcal{B}}$  or the vector expansion  $\Phi_{\mathcal{B}}$ . When the underlying  $\mathbb{F}_{q^m}$ -linear code is a Gabidulin code, these constructions lead to the corresponding *Gabidulin matrix code* and *expanded Gabidulin code*.

Applying the matrix expansion to  $\mathcal{G}$ , we obtain the *Gabidulin matrix code*

$$\text{ext}_{\mathcal{B}}(\mathcal{G}) := \{\text{ext}_{\mathcal{B}}(\mathbf{c}) \in \mathbb{F}_q^{m \times n} : \mathbf{c} \in \mathcal{G}\} \subseteq \mathbb{F}_q^{m \times n}.$$

Similarly, applying the vector expansion to  $\mathcal{G}$ , we obtain the *expanded Gabidulin code*

$$\Phi_{\mathcal{B}}(\mathcal{G}) := \{\Phi_{\mathcal{B}}(\mathbf{c}) \in \mathbb{F}_q^{mn} : \mathbf{c} \in \mathcal{G}\} \subseteq \mathbb{F}_q^{mn}.$$

Since these constructions are particular cases of the matrix and vector expansions

defined in Section 3.2, the bijective correspondence between matrix and vector representations induced by the maps **Fold** and **Unfold** also holds for Gabidulin codes. In particular, we have

$$\text{ext}_{\mathcal{B}}(\mathcal{G}) = \mathbf{Fold}(\Phi_{\mathcal{B}}(\mathcal{G})), \quad \Phi_{\mathcal{B}}(\mathcal{G}) = \mathbf{Unfold}(\text{ext}_{\mathcal{B}}(\mathcal{G})).$$

It is worth emphasizing how Gabidulin matrix codes are defined in [6, Def. 8]. In fact, in this paper, Gabidulin matrix codes are defined in a slightly more general way with respect to the definition we adopted above: a matrix code  $\mathcal{C}_{mat} \subseteq \mathbb{F}_q^{m \times n}$  is called a Gabidulin matrix code if there exist a Gabidulin code  $\mathcal{G}[\mathbf{g}; n, k]_{q^m} \subseteq \mathbb{F}_{q^m}^n$ , a basis  $\mathcal{B}$  of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ , and matrices  $\mathbf{P} \in \text{GL}(m, q)$  and  $\mathbf{Q} \in \text{GL}(n, q)$  such that

$$\mathcal{C}_{mat} = \{\mathbf{P} \text{ext}_{\mathcal{B}}(\mathbf{c})\mathbf{Q} : \mathbf{c} \in \mathcal{G}\}.$$

The following lemmas show that matrix codes obtained in this way can be interpreted as matrix expansions of Gabidulin codes. More precisely, the left multiplication by  $\mathbf{P}$  corresponds to a change of basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ , while the right multiplication by  $\mathbf{Q}$  corresponds to a change of the support of the Gabidulin code.

Therefore, this observation allows us to reinterpret all Gabidulin matrix codes, as defined in [6], as matrix expansions of suitable Gabidulin codes with respect to suitable bases.

**Lemma 3.** *Let  $\mathcal{G} = \mathcal{G}[\mathbf{g}; n, k]_{q^m} \subseteq \mathbb{F}_{q^m}^n$  be a Gabidulin code, let  $\mathcal{B} = (b_1, \dots, b_m)$  be a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ , and let  $\mathbf{P} \in \text{GL}(m, q)$ . Define the matrix code*

$$\mathcal{C}_{mat} = \{\mathbf{M}_c : \mathbf{M}_c = \mathbf{P} \text{ext}_{\mathcal{B}}(\mathbf{c}), \mathbf{c} \in \mathcal{G}\}.$$

Then

$$\mathcal{C}_{mat} = \mathbf{Fold}(\Phi_{\mathcal{B}\mathbf{P}^{-1}}(\mathcal{G})).$$

*Proof.* Before proceeding with the proof, we present a simple identity that helps clarify the key step of the argument.

Let  $\mathcal{B} = (b_1, \dots, b_m)$  be a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ , let  $x \in \mathbb{F}_{q^m}$ .

Then

$$\phi_{\mathcal{B}}(x)^{\top} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}, \quad x = \sum_{j=1}^m x_j b_j, \quad x_j \in \mathbb{F}_q \quad \forall j \in [1, m].$$

Then we can write

$$\mathcal{B}\phi_{\mathcal{B}}(x)^{\top} = x. \tag{3.5}$$

Analogously,

$$\phi_{\mathcal{B}\mathbf{P}^{-1}}(x)^\top = \begin{pmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_m \end{pmatrix}, \quad x = \sum_{j=1}^m \tilde{x}_j (\mathcal{B}\mathbf{P}^{-1})_j, \quad \tilde{x}_j \in \mathbb{F}_q \forall j \in [1, m].$$

Thus,

$$\mathcal{B}\mathbf{P}^{-1}\phi_{\mathcal{B}\mathbf{P}^{-1}}(x)^\top = x.$$

By associativity,

$$\mathcal{B}(\mathbf{P}^{-1}\phi_{\mathcal{B}\mathbf{P}^{-1}}(x)^\top) = x. \quad (3.6)$$

Combining (3.5) and (3.6), we obtain

$$\phi_{\mathcal{B}}(x)^\top = \mathbf{P}^{-1}\phi_{\mathcal{B}\mathbf{P}^{-1}}(x)^\top.$$

Multiplying both sides by  $\mathbf{P}$ , we obtain

$$\phi_{\mathcal{B}\mathbf{P}^{-1}}(x)^\top = \mathbf{P}\phi_{\mathcal{B}}(x)^\top.$$

Transposing both sides,

$$\boxed{\phi_{\mathcal{B}\mathbf{P}^{-1}}(x) = \phi_{\mathcal{B}}(x)\mathbf{P}^\top.}$$

We now proceed with the main argument of the proof.

Let  $\mathbf{c}$  be a codeword of  $\mathcal{G}$ ,

$$\mathbf{c} = (c_1, \dots, c_n) \in \mathbb{F}_q^n, \quad \mathbf{c} \in \mathcal{G}.$$

Consider

$$\begin{aligned} \Phi_{\mathcal{B}}(\mathbf{c}) &= (C_{1,1}, \dots, C_{m,1}, \dots, C_{1,n}, \dots, C_{m,n}) \\ &= (\text{---}\phi_{\mathcal{B}}(c_1)\text{---}, \dots, \text{---}\phi_{\mathcal{B}}(c_n)\text{---}) \in \mathbb{F}_q^{mn}. \end{aligned}$$

$$\phi_{\mathcal{B}}(c_i)^\top = \begin{pmatrix} C_{1,i} \\ \vdots \\ C_{m,i} \end{pmatrix} \in \mathbb{F}_q^{m \times 1}, \quad c_i = \sum_{j=1}^m C_{j,i} b_j, \quad \forall i \in [1, n].$$

We rewrite the last equation as

$$c_i = \mathcal{B}\phi_{\mathcal{B}}(c_i)^\top, \quad \forall i \in [1, n]. \quad (3.7)$$

Analogously,

$$\begin{aligned} \Phi_{\mathcal{B}\mathbf{P}^{-1}}(\mathbf{c}) &= (\tilde{C}_{1,1}, \dots, \tilde{C}_{m,1}, \dots, \tilde{C}_{1,n}, \dots, \tilde{C}_{m,n}) \\ &= (\text{---}\phi_{\mathcal{B}\mathbf{P}^{-1}}(c_1)\text{---}, \dots, \text{---}\phi_{\mathcal{B}\mathbf{P}^{-1}}(c_n)\text{---}) \in \mathbb{F}_q^{mn}. \end{aligned} \quad (3.8)$$

$$\phi_{\mathcal{B}\mathcal{P}^{-1}}(c_i)^\top = \begin{pmatrix} \tilde{C}_{1,i} \\ \vdots \\ \tilde{C}_{m,i} \end{pmatrix} \in \mathbb{F}_q^{m \times 1}, \quad c_i = \sum_{j=1}^m \tilde{C}_{j,i} (\mathcal{B}\mathcal{P}^{-1})_j, \quad \forall i \in [1, n].$$

Hence,

$$c_i = \mathcal{B}\mathcal{P}^{-1} \phi_{\mathcal{B}\mathcal{P}^{-1}}(c_i)^\top, \quad \forall i \in [1, n].$$

By associativity,

$$c_i = \mathcal{B}(\mathcal{P}^{-1} \phi_{\mathcal{B}\mathcal{P}^{-1}}(c_i)^\top), \quad \forall i \in [1, n]. \quad (3.9)$$

Combining (3.7) and (3.9), we obtain

$$\phi_{\mathcal{B}}(c_i)^\top = \mathcal{P}^{-1} \phi_{\mathcal{B}\mathcal{P}^{-1}}(c_i)^\top.$$

Multiplying both sides by  $\mathcal{P}$ ,

$$\phi_{\mathcal{B}\mathcal{P}^{-1}}(c_i)^\top = \mathcal{P} \phi_{\mathcal{B}}(c_i)^\top.$$

Transposing both sides,

$$\boxed{\phi_{\mathcal{B}\mathcal{P}^{-1}}(c_i) = \phi_{\mathcal{B}}(c_i) \mathcal{P}^\top.}$$

Substituting this expression into (3.8), we obtain

$$\Phi_{\mathcal{B}\mathcal{P}^{-1}}(\mathbf{c}) = \left( \_ \phi_{\mathcal{B}}(c_1) \mathcal{P}^\top \_, \dots, \_ \phi_{\mathcal{B}}(c_n) \mathcal{P}^\top \_ \right)$$

Therefore,

$$\begin{aligned} \mathbf{Fold} \left( \Phi_{\mathcal{B}\mathcal{P}^{-1}}(\mathbf{c}) \right) &= \left( \begin{array}{c|c} & \left( \phi_{\mathcal{B}}(c_1) \mathcal{P}^\top \right)^\top \\ \hline & \dots \\ \hline & \left( \phi_{\mathcal{B}}(c_n) \mathcal{P}^\top \right)^\top \end{array} \right) \\ &= \left( \begin{array}{c|c} & \left( \mathcal{P} \phi_{\mathcal{B}}(c_1) \right)^\top \\ \hline & \dots \\ \hline & \left( \mathcal{P} \phi_{\mathcal{B}}(c_n) \right)^\top \end{array} \right) \in \mathbb{F}_q^{m \times n} \end{aligned}$$

Hence

$$\begin{aligned} \mathbf{Fold} \left( \Phi_{\mathcal{B}\mathcal{P}^{-1}}(\mathbf{c}) \right) &= \mathcal{P} \left( \begin{array}{c|c} & \phi_{\mathcal{B}}(c_1)^\top \\ \hline & \dots \\ \hline & \phi_{\mathcal{B}}(c_n)^\top \end{array} \right) \\ &= \mathcal{P} \mathbf{Fold} \left( \Phi_{\mathcal{B}}(\mathbf{c}) \right) \\ &= \mathcal{P} \text{ext}_{\mathcal{B}}(\mathbf{c}). \end{aligned}$$

Since this holds for every  $\mathbf{c} \in \mathcal{G}$ , we conclude that

$$\mathbf{Fold} \left( \Phi_{\mathcal{B}\mathcal{P}^{-1}}(\mathcal{G}) \right) = \mathcal{C}_{mat}$$

where

$$\mathcal{C}_{mat} = \{M_c : M_c = P \text{ext}_{\mathcal{B}}(\mathbf{c}), \mathbf{c} \in \mathcal{G}\}.$$

□

**Lemma 4.** *Let  $\mathcal{G}[\mathbf{g}; n, k]_{q^m} \subseteq \mathbb{F}_{q^m}^n$  be a Gabidulin code, let  $\mathcal{B} = (b_1, \dots, b_m)$  be a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ , and let  $\mathbf{Q} \in \text{GL}(n, q)$ .*

*Define the matrix code*

$$\mathcal{C}_{mat} = \{M_c : M_c = \text{ext}_{\mathcal{B}}(\mathbf{c})\mathbf{Q}, \mathbf{c} \in \mathcal{G}\}.$$

*Then*

$$\mathcal{C}_{mat} = \mathbf{Fold}\left(\Phi_{\mathcal{B}}(\mathcal{G}[\mathbf{g}\mathbf{Q}; n, k]_{q^m})\right).$$

*Proof.* This lemma follows from the fact that for a Gabidulin code  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$  and a matrix  $\mathbf{Q} \in \text{GL}(n, q)$ , the code

$$\{\mathbf{c}\mathbf{Q} : \mathbf{c} \in \mathcal{G}\}$$

is a Gabidulin code with support  $\mathbf{g}\mathbf{Q}$ , see [6, Section 2.4].

□

**Corollary 5.** *Let  $\mathcal{G}[\mathbf{g}; n, k]_{q^m} \subseteq \mathbb{F}_{q^m}^n$  be a Gabidulin code, let  $\mathcal{B} = (b_1, \dots, b_m)$  be a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ , and let  $\mathbf{P} \in \text{GL}(m, q)$  and  $\mathbf{Q} \in \text{GL}(n, q)$ .*

*Define the matrix code*

$$\mathcal{C}_{mat} = \{M_c : M_c = \mathbf{P} \text{ext}_{\mathcal{B}}(\mathbf{c})\mathbf{Q}, \mathbf{c} \in \mathcal{G}\}.$$

*Then*

$$\mathcal{C}_{mat} = \mathbf{Fold}\left(\Phi_{\mathcal{B}\mathbf{P}^{-1}}(\mathcal{G}[\mathbf{g}\mathbf{Q}; n, k]_{q^m})\right).$$

*Proof.* The result follows by combining Lemma 3 and Lemma 4, since the left multiplication by  $\mathbf{P}$  (corresponding to a change of basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ ) and the right multiplication by  $\mathbf{Q}$  (corresponding to a change of support of the Gabidulin code) act independently. □

## Recovery of a Generator Matrix of the Gabidulin Code from Its Expanded Gabidulin Code

In this subsection we present a method to recover a Gabidulin code, and in particular its vector of code locators, from which an expanded Gabidulin code has been constructed. More precisely, we assume that a generator matrix of the expanded Gabidulin code is available, constructed as described in Section 3.2, and that the basis used for the expansion is known.

We place ourselves in the perspective of an attacker who obtains a generator matrix

$$\mathbf{G}_q \in \mathbb{F}_q^{mk \times mn}$$

of the expanded Gabidulin code  $\Phi_{\mathcal{B}}(\mathcal{G})$ . We also assume that the basis  $\mathcal{B}$  used for the expansion is known to the attacker.

As shown in Section 3.2, this matrix is obtained by replacing each entry  $G_{i,j} \in \mathbb{F}_{q^m}$  of a generator matrix  $\mathbf{G} \in \mathbb{F}_{q^m}^{k \times n}$  of the Gabidulin code with its corresponding matrix representation  $\mathbf{M}_{G_{i,j}} \in \mathbb{F}_q^{m \times m}$ .

Hence,  $\mathbf{G}_q$  can be viewed as a block matrix of the form

$$\mathbf{G}_q = \begin{pmatrix} \mathbf{M}_{G_{1,1}} & \cdots & \mathbf{M}_{G_{1,n}} \\ \vdots & \ddots & \vdots \\ \mathbf{M}_{G_{k,1}} & \cdots & \mathbf{M}_{G_{k,n}} \end{pmatrix},$$

where each block  $\mathbf{M}_{G_{i,j}}$  has dimension  $m \times m$ .

Let us now see how the elements  $G_{i,j} \in \mathbb{F}_{q^m}$  can be recovered from the blocks  $\mathbf{M}_{G_{i,j}}$ .

Let  $\tilde{\alpha} \in \mathbb{F}_{q^m}$  be a primitive element and consider the matrix  $\mathbf{M}_{\tilde{\alpha}} \in \mathbb{F}_q^{m \times m}$  representing the multiplication-by- $\tilde{\alpha}$  map with respect to the basis  $\mathcal{B}$ , as defined in (3.4). Since the basis  $\mathcal{B}$  is assumed to be known, the attacker can explicitly construct the matrix  $\mathbf{M}_{\tilde{\alpha}}$ .

By construction of the representation matrices, we have the property

$$\mathbf{M}_{\tilde{\alpha}}^u = \mathbf{M}_{\tilde{\alpha}^u} \quad \text{for all } u \in \mathbb{Z}_{\geq 0}.$$

Indeed,  $\mathbf{M}_{\tilde{\alpha}}$  represents the linear map  $x \mapsto \tilde{\alpha}x$  over  $\mathbb{F}_q$ , and composing this map  $u$  times corresponds to the multiplication-by- $\tilde{\alpha}^u$  map.

Suppose we focus on one block of  $\mathbf{G}_q$ , for instance  $\mathbf{M}_{G_{1,1}}$ . If  $G_{1,1} = 0$ , then  $\mathbf{M}_{G_{1,1}}$  is the zero matrix and the corresponding entry is immediately determined. Hence we restrict to the case  $G_{1,1} \neq 0$ .

Since every non-zero element of  $\mathbb{F}_{q^m}$  can be written as a power of the primitive element  $\tilde{\alpha}$ , there exists an integer  $\tilde{u}$  such that

$$G_{1,1} = \tilde{\alpha}^{\tilde{u}}.$$

Consequently,

$$\mathbf{M}_{G_{1,1}} = \mathbf{M}_{\tilde{\alpha}^{\tilde{u}}} = \mathbf{M}_{\tilde{\alpha}}^{\tilde{u}}.$$

Therefore, the attacker can recover  $G_{1,1}$  by computing successive powers of  $M_{\tilde{\alpha}}$  until finding an exponent  $\tilde{u}$  such that

$$M_{\tilde{\alpha}}^{\tilde{u}} = M_{G_{1,1}}.$$

Once such an exponent is found, it follows that

$$G_{1,1} = \tilde{\alpha}^{\tilde{u}}.$$

The same procedure can be applied to every block  $M_{G_{i,j}}$  of  $\mathbf{G}_q$ . For each block, one searches for an exponent  $u$  satisfying

$$M_{\tilde{\alpha}}^u = M_{G_{i,j}},$$

which yields

$$G_{i,j} = \tilde{\alpha}^u.$$

Repeating this process for all  $i = 1, \dots, k$  and  $j = 1, \dots, n$  allows the attacker to reconstruct all entries of the matrix  $\mathbf{G}$ , thereby recovering a generator matrix of the underlying Gabidulin code.

It is important to point out that the procedure described above works only if the block structure of the matrix  $\mathbf{G}_q$  is preserved. If this block structure is scrambled, the method cannot be applied. However, this is usually not an issue in cryptographic constructions, since the public generator matrix is typically assumed to admit a systematic form; therefore, it is either already in systematic form or can be transformed into one. Passing to a systematic form enforces a consistent  $m \times m$  block structure, which allows the attacker to correctly identify the blocks.

At this point the attacker has obtained a generator matrix of the underlying Gabidulin code. However, recovering a generator matrix of the Gabidulin code does not immediately reveal the vector of code locators

$$\mathbf{g} = (g_1, \dots, g_n).$$

The recovery of the code locators can be achieved by exploiting the algebraic structure of Gabidulin codes. In particular, if the matrix  $\mathbf{G}$  is obtained from a generator matrix in  $q$ -Vandermonde form by an isometric disguising transformation, one can apply Overbeck's attack. This attack allows the recovery of the vector of code locators  $\mathbf{g} = (g_1, \dots, g_n)$ .

Overbeck's attack will be described in Chapter 4, Section 4.1, where we also explain what is meant by an isometric disguising transformation.

# Chapter 4

## Algebraic Attacks

In this chapter we review two algebraic attacks that play an important role in the study of code-based cryptosystems based on structured codes.

As discussed in the introduction, Gabidulin codes are attractive for cryptographic constructions because they allow efficient decoding in the rank metric while potentially enabling significantly smaller public keys compared to Hamming-metric codes. However, the strong algebraic structure that makes Gabidulin codes efficient also makes them vulnerable to structural attacks.

### 4.1 Overbeck's Attack

A prominent example is the *Overbeck attack* [12], which exploits the  $\mathbb{F}_{q^m}$ -linear structure of Gabidulin codes and the action of the Frobenius automorphism. This attack can distinguish a Gabidulin code from a random code and even recover a generator matrix of the underlying Gabidulin code. As a consequence, Gabidulin-based cryptosystems, such as the GPT system, were successfully broken.

This observation shows that, in Gabidulin-based cryptosystems, the  $\mathbb{F}_{q^m}$ -linear structure of the code must be hidden. One strategy is to expand Gabidulin codes over the base field  $\mathbb{F}_q$ , thereby destroying the structure exploited by Overbeck's attack. Several cryptographic constructions, including the systems studied later in this thesis, follow this approach.

Let us consider a  $q$ -*Vandermonde* generator matrix  $\mathbf{G} \in \mathbb{F}_{q^m}^{k \times n}$  of a Gabidulin code  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$

$$\begin{aligned} \mathbf{G} = \text{van}_{k,q}(\mathbf{g}) &= \begin{pmatrix} g_1 & \cdots & g_n \\ g_1^q & \cdots & g_n^q \\ \vdots & \ddots & \vdots \\ g_1^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{pmatrix} = \begin{pmatrix} g_1^{[0]} & \cdots & g_n^{[0]} \\ g_1^{[1]} & \cdots & g_n^{[1]} \\ \vdots & \ddots & \vdots \\ g_1^{[k-1]} & \cdots & g_n^{[k-1]} \end{pmatrix} \\ &= \begin{pmatrix} \phi_q^0(g_1) & \cdots & \phi_q^0(g_n) \\ \phi_q^1(g_1) & \cdots & \phi_q^1(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^{k-1}(g_1) & \cdots & \phi_q^{k-1}(g_n) \end{pmatrix} \end{aligned}$$

with an abuse of notation we write:

$$\phi_q(\mathbf{G}) = \begin{pmatrix} \phi_q(\phi_q^0(g_1)) & \cdots & \phi_q(\phi_q^0(g_n)) \\ \phi_q(\phi_q^1(g_1)) & \cdots & \phi_q(\phi_q^1(g_n)) \\ \vdots & \ddots & \vdots \\ \phi_q(\phi_q^{k-1}(g_1)) & \cdots & \phi_q(\phi_q^{k-1}(g_n)) \end{pmatrix} = \begin{pmatrix} \phi_q^1(g_1) & \cdots & \phi_q^1(g_n) \\ \phi_q^2(g_1) & \cdots & \phi_q^2(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^k(g_1) & \cdots & \phi_q^k(g_n) \end{pmatrix}$$

Note that  $\phi_q(\mathbf{G})$  generates the Gabidulin code  $\mathcal{G}(\phi_q(\mathbf{g}); n, k)$ .

The matrix  $\phi_q(\mathbf{G})$  can be obtained from  $\mathbf{G}$  by removing the first row  $\mathbf{g}$  and appending the new row  $\phi_q^k(\mathbf{g})$  at the bottom. In particular, the rows  $\phi_q(\mathbf{g}), \dots, \phi_q^{k-1}(\mathbf{g})$  already appear as rows of  $\mathbf{G}$ , while  $\phi_q^k(\mathbf{g})$  is the only new row. The latter is not only new, but also linearly independent from the others as long as  $k < n$ <sup>1</sup>.

As a consequence, the code generated by the vertically concatenated matrix

$$\begin{pmatrix} \mathbf{G} \\ \phi_q(\mathbf{G}) \end{pmatrix}$$

has dimension exactly  $k + 1$  for  $k < n$ .

If  $k = n$ , the vectors  $\mathbf{g}, \phi_q(\mathbf{g}), \dots, \phi_q^{n-1}(\mathbf{g})$  form a set of  $n$  linearly independent vectors. Since the ambient space  $\mathbb{F}_{q^m}^n$  has dimension  $n$ , it cannot contain any additional linearly independent vectors. In particular, the Frobenius power  $\phi_q^n(\mathbf{g})$  must be a linear combination of the previous ones, and therefore the dimension cannot increase and remains equal to  $n$ .

Therefore we now know that the code generated by the vertical concatenation

$$\begin{pmatrix} \mathbf{G} \\ \phi_q(\mathbf{G}) \end{pmatrix}$$

---

<sup>1</sup>this follows from having chosen the coordinates of  $\mathbf{g}$  being  $\mathbb{F}_q$ -linearly independent

has dimension exactly  $\min\{k + 1, n\}$ .

In contrast, for a random code over  $\mathbb{F}_{q^m}$ , the dimension of such a concatenation would be  $\min\{2k, n\}$  with overwhelming probability. If  $2k > n$ , again, the dimension cannot exceed that of the ambient space  $\mathbb{F}_{q^m}^n$ , therefore, in this case, the code saturates the whole space and has dimension  $n$ .

Therefore, at this point, one is already able to distinguish the Gabidulin code from a random one.

Moreover, with some further steps, one can even do better.

The idea is to apply the Frobenius automorphism multiple times:

$$\begin{aligned} \phi_q(\phi_q(\mathbf{G})) &= \phi_q^2(\mathbf{G}) = \begin{pmatrix} \phi_q(\phi_q^1(g_1)) & \cdots & \phi_q(\phi_q^1(g_n)) \\ \phi_q(\phi_q^2(g_1)) & \cdots & \phi_q(\phi_q^2(g_n)) \\ \vdots & \ddots & \vdots \\ \phi_q(\phi_q^k(g_1)) & \cdots & \phi_q(\phi_q^k(g_n)) \end{pmatrix} \\ &= \begin{pmatrix} \phi_q^2(g_1) & \cdots & \phi_q^2(g_n) \\ \phi_q^3(g_1) & \cdots & \phi_q^3(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^{k+1}(g_1) & \cdots & \phi_q^{k+1}(g_n) \end{pmatrix} \end{aligned}$$

in general:

$$\phi_q^r(\mathbf{G}) = \begin{pmatrix} \phi_q^r(g_1) & \cdots & \phi_q^r(g_n) \\ \phi_q^{r+1}(g_1) & \cdots & \phi_q^{r+1}(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^{k+r-1}(g_1) & \cdots & \phi_q^{k+r-1}(g_n) \end{pmatrix}$$

Then, we stack vertically the successive powers of the Frobenius automorphism applied to the generator matrix  $\mathbf{G}$ , forming the matrix  $\mathbf{M}$ . We consider all powers

up to  $n - k - 1$ :

$$\mathbf{M} = \begin{pmatrix} \mathbf{G} \\ \hline \phi_q(\mathbf{G}) \\ \hline \phi_q^2(\mathbf{G}) \\ \hline \vdots \\ \hline \phi_q^{n-k-1}(\mathbf{G}) \end{pmatrix} = \begin{pmatrix} \phi_q^0(g_1) & \cdots & \phi_q^0(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^{k-1}(g_1) & \cdots & \phi_q^{k-1}(g_n) \\ \hline \phi_q(g_1) & \cdots & \phi_q(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^k(g_1) & \cdots & \phi_q^k(g_n) \\ \hline \phi_q^2(g_1) & \cdots & \phi_q^2(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^{k+1}(g_1) & \cdots & \phi_q^{k+1}(g_n) \\ \hline \vdots & & \vdots \\ \hline \phi_q^{n-k-1}(g_1) & \cdots & \phi_q^{n-k-1}(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^{n-2}(g_1) & \cdots & \phi_q^{n-2}(g_n) \end{pmatrix}$$

Note that many rows of this matrix are the same.

We consider the code spanned by  $\mathbf{M}$ :

$$\langle \mathbf{M} \rangle_{q^m} = \left\langle \begin{pmatrix} \phi_q^0(g_1) & \cdots & \phi_q^0(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^{k-1}(g_1) & \cdots & \phi_q^{k-1}(g_n) \\ \hline \phi_q(g_1) & \cdots & \phi_q(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^k(g_1) & \cdots & \phi_q^k(g_n) \\ \hline \phi_q^2(g_1) & \cdots & \phi_q^2(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^{k+1}(g_1) & \cdots & \phi_q^{k+1}(g_n) \\ \hline \vdots & & \vdots \\ \hline \phi_q^{n-k-1}(g_1) & \cdots & \phi_q^{n-k-1}(g_n) \\ \vdots & \ddots & \vdots \\ \phi_q^{n-2}(g_1) & \cdots & \phi_q^{n-2}(g_n) \end{pmatrix} \right\rangle_{q^m} = \mathcal{G}(\mathbf{g}; n, n-1)$$

The dual of the Gabidulin code  $\mathcal{G}[\mathbf{g}; n, n-1]_{q^m}$ , denoted by  $\mathcal{G}[\mathbf{g}; n, n-1]_{q^m}^\perp$ , has dimension  $n - (n-1) = 1$ . Since the dual of a Gabidulin code is itself a Gabidulin code, we can view  $\mathcal{G}[\mathbf{g}; n, n-1]_{q^m}^\perp$  as a Gabidulin code of length  $n$  and dimension 1. Its generator matrix is given by the parity-check matrix  $\mathbf{H}$  of  $\mathcal{G}[\mathbf{g}; n, n-1]_{q^m}$ ,

which in this case consists of a single row  $(h_1, \dots, h_n)$  satisfying the system

$$\sum_{i=1}^n g_i^{[j]} h_i = 0, \quad j = -n + k + 1, \dots, k - 1. \quad (4.1)$$

Therefore, not only we have been able to distinguish the code just by computing the dimension of the code generated by the matrix  $\mathbf{M}$ , we are also able to recover the generator matrix of the original Gabidulin code,  $\mathbf{G}$ , since we now know  $(h_1, \dots, h_n)$  from which we can recover  $\mathbf{g}$  by solving (4.1).

We point out that, even if a generator matrix is not given in the standard  $q$ -Vandermonde form, it may arise from such a matrix by multiplication on the left by an invertible matrix over  $\mathbb{F}_{q^m}$  and on the right by an invertible matrix over  $\mathbb{F}_q$ , i.e.,

$$\mathbf{G}' = \mathbf{S}\mathbf{G}\mathbf{T}, \quad \text{with } \mathbf{S} \in \text{GL}(k, \mathbb{F}_{q^m}), \quad \mathbf{T} \in \text{GL}(n, \mathbb{F}_q).$$

These transformations, often referred to as isometric disguising transformations, preserve the Gabidulin structure. Therefore, Overbeck's attack still applies [12].

## 4.2 Square-Code Attack

In addition to the Overbeck attack, we also discuss the *square-code attack*. This attack is a classical distinguisher which is successful for Generalized Reed-Solomon codes. It exploits the dimension of the square code to detect hidden algebraic structure.

For any code  $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$  of dimension  $k$ , the dimension of its square code is upper bounded by

$$\dim(\mathcal{C}^{*2}) \leq \min \left\{ \binom{k+1}{2}, n \right\}.$$

This bound follows from the fact that  $\mathcal{C}^{*2}$  is generated by all component-wise products  $\mathbf{c}_i * \mathbf{c}_j$  with  $1 \leq i \leq j \leq k$ , where  $\mathbf{c}_1, \dots, \mathbf{c}_k \in \mathbb{F}_{q^m}^n$  denote the rows of a generator matrix of  $\mathcal{C}$ , which gives at most  $\binom{k+1}{2}$  generators.

However, for codes having strong algebraic structure, the dimension of the square code may be significantly smaller than this upper bound.

In contrast, for a random linear code  $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$  of dimension  $k$ , it holds with high probability that

$$\dim(\mathcal{C}^{*2}) = \min \left\{ \binom{k+1}{2}, n \right\}.$$

We define the *Generalized Reed-Solomon* (*GRS* codes) in order to explain how to perform the square code attack.

For column multipliers  $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n) \in \mathbb{F}_{q^m}^n$  with non-zero entries, define

$$\mathcal{GRS}(\mathbf{g}, \boldsymbol{\lambda}; n, k) := \left\{ (\lambda_1 f(g_1), \dots, \lambda_n f(g_n)) = f(\mathbf{g}) : f(x) \in \mathbb{F}_{q^m}[x]_{<k} \right\} \subseteq \mathbb{F}_{q^m}^n$$

A generator matrix of a  $\mathcal{GRS}(\mathbf{g}, \boldsymbol{\lambda}; n, k)$  is

$$\mathbf{G} = \begin{pmatrix} 1 & \cdots & 1 \\ g_1 & \cdots & g_n \\ \vdots & \ddots & \vdots \\ g_1^{k-1} & \cdots & g_n^{k-1} \end{pmatrix} \cdot \text{diag}(\lambda_1, \dots, \lambda_n).$$

**Note that** this matrix has rank  $k$ . Indeed, it is the product of a Vandermonde matrix evaluated at pairwise distinct elements  $g_1, \dots, g_n$  and a diagonal matrix. The first factor has rank  $k$  as the rank of a Vandermonde matrix is given by the minimum between the number of columns and the number of distinct evaluation points  $g_i$ , which are all distinct by hypothesis. Since  $\text{diag}(\lambda_1, \dots, \lambda_n)$  is a diagonal matrix with non-zero entries, it is invertible and hence has full rank. Right multiplication by an invertible matrix preserves the rank, and therefore the generator matrix has rank  $k$ .

Let  $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C} = \mathcal{GRS}(\mathbf{g}, \boldsymbol{\lambda}; n, k)$  such that  $\mathbf{c}_1$  corresponds to  $f_1 \in \mathbb{F}_{q^m}[x]_{<k}$  and  $\mathbf{c}_2$  corresponds to  $f_2 \in \mathbb{F}_{q^m}[x]_{<k}$ .

Then

$$\mathbf{c}_1 * \mathbf{c}_2 = (\lambda_1 f_1(g_1) \cdot \lambda_1 f_2(g_1), \dots, \lambda_n f_1(g_n) \cdot \lambda_n f_2(g_n))$$

without loss of generality, looking at the first entry

$$\lambda_1^2 f_1(g_1) f_2(g_1) = \lambda_1^2 \left( \sum_{j=0}^{k-1} f_{1,j} g_1^j \right) \left( \sum_{j=0}^{k-1} f_{2,j} g_1^j \right) = \lambda_1^2 (f_1 f_2)(g_1)$$

where by  $(f_1 f_2)(g_1)$  we mean the product of the polynomials  $f_1(x)$  and  $f_2(x)$  evaluated in  $g_1$ .

Therefore

$$\mathbf{c}_1 * \mathbf{c}_2 = (\lambda_1^2 (f_1 f_2)(g_1), \dots, \lambda_n^2 (f_1 f_2)(g_n)) \in \mathcal{GRS}(\mathbf{g}, \boldsymbol{\lambda}^2; n, 2k - 1)$$

Thereby the dimension of the square-code of a code  $\mathcal{C} = \mathcal{GRS}(\mathbf{g}, \boldsymbol{\lambda}; n, k)$  is

$$\dim(\mathcal{C}^{*2}) = 2k - 1 \quad \text{where} \quad 2k - 1 \leq n.$$

However, if  $2k - 1 > n$ , the dimension cannot exceed that of the ambient space  $\mathbb{F}_{q^m}^n$ , and thus

$$\dim(\mathcal{C}^{*2}) = n \quad \text{with} \quad 2k - 1 > n.$$

Hence, in general, the dimension of the square code of a GRS code satisfies

$$\dim(\mathcal{C}^{*2}) = \min(2k - 1, n).$$

For  $2k - 1 > n$ , the square code behaves like that of a random linear code, so the square-code distinguisher no longer applies.

Consequently, the distinguisher is effective only when  $2k - 1 \leq n$ . Therefore, as long as this condition is satisfied, to distinguish whether  $\mathbf{G} \in \mathbb{F}_{q^m}^{k \times n}$  generates a GRS code or a random code, it is sufficient to compute the dimension of its square code.

### Square-Code Distinguisher on Expanded Gabidulin Codes

In this thesis we experimentally investigated whether the square-code distinguisher can detect algebraic structure in expanded Gabidulin codes.

In our implementation, we focused on computing a generator matrix of an expanded Gabidulin code using the construction introduced in Chapter 3, Section 3.2. More precisely, starting from a Gabidulin code  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$ , we computed a generator matrix of its vector expansion  $\Phi_{\mathcal{B}}(\mathcal{G})$ . We then applied the square-code distinguisher to this code. The purpose of the experiment was to verify whether this representation still exposes any exploitable algebraic structure.

The implementation was carried out in `SageMath`. For each set of parameters  $(q, m, n, k)$ , we generated a Gabidulin code, constructed the generator matrix of its vector expansion, and computed the dimension of the corresponding square code. This experiment was repeated several times (each time starting from a different Gabidulin code) for each parameter set.

Recall that if a code over  $\mathbb{F}_q$  has dimension  $K$  and length  $N$ , then the dimension of its square code is upper bounded by

$$\min \left\{ \binom{K+1}{2}, N \right\}.$$

For a random linear code, this bound is typically met with high probability. Therefore, if the computed dimension of the square code is close to this bound, the code behaves like a random code from the point of view of the square-code distinguisher.

For the expanded Gabidulin code  $\Phi_{\mathcal{B}}(\mathcal{G})$ , the parameters of the resulting code are

$$N = mn, \quad K = mk.$$

Consequently, the expected dimension of the square code for a random-looking code is

$$\min \left\{ \binom{mk+1}{2}, mn \right\}.$$

Table 4.1 reports the results of our experiments for several parameter choices. In each case, the observed dimension of the square code coincides with the value expected for random codes.

For comparison, we also report the dimension expected for a random linear code with the same parameters, denoted by  $\dim(\mathcal{C}_{rand}^{*2})$ .

$q$	$m$	$n$	$k$	$\dim(\Phi_{\mathcal{B}}(\mathcal{G})^{*2})$	$\dim(\mathcal{C}_{rand}^{*2})$
2	40	30	6	1200	1200
2	35	30	7	1050	1050
2	30	25	6	750	750
2	25	21	5	525	525
2	20	18	5	360	360
2	16	16	5	256	256
2	15	12	4	180	180
2	29	29	7	841	841
2	22	22	6	484	484

**Table 4.1:** Square-code dimensions for expanded Gabidulin codes.

For each parameter set, the experiment was repeated 100 times and the same square-code dimension was obtained in every run. In all cases, the dimension equals the value expected for random linear codes, meaning that the square-code distinguisher does not reveal any exploitable algebraic structure.

This behaviour is consistent with the fact that the square-code attack is specifically effective against Reed–Solomon and Generalized Reed–Solomon codes, whereas Gabidulin codes are not Reed–Solomon-like. In fact, they belong to a different family of algebraic codes defined via linearized polynomials and based on the rank metric. As a consequence, their vector expansions do not exhibit the same square-code dimension behaviour.

These experimental results therefore confirm that the square-code distinguisher is not effective against expanded Gabidulin codes.

For comparison, we also applied the square-code distinguisher directly to Gabidulin codes over the extension field  $\mathbb{F}_{q^m}$ .

For each parameter set  $(q, m, n, k)$  we generated a Gabidulin code  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$  and computed the dimension of its square code. The experiment was, again, repeated 100 times for each parameter set.

In this case the code has dimension  $k$  and length  $n$  over  $\mathbb{F}_{q^m}$ , and therefore the square code is less than or equal to

$$\min\left\{\binom{k+1}{2}, n\right\}.$$

As in the previous experiment, the observed dimension always coincides with the value expected for random linear codes.

$q$	$m$	$n$	$k$	$\dim(\mathcal{G}^{*2})$	$\dim(\mathcal{C}_{rand}^{*2})$
2	40	30	6	21	21
2	35	30	7	28	28
2	30	25	6	21	21
2	25	21	5	15	15
2	20	18	5	15	15
2	16	16	5	15	15
2	15	12	4	10	10
2	29	29	7	28	28
2	22	22	6	21	21

**Table 4.2:** Square-code dimensions for Gabidulin codes over  $\mathbb{F}_{q^m}$ .

As expected, the square-code distinguisher does not reveal any algebraic structure in Gabidulin codes either. This behaviour is consistent with the explanation given above: Gabidulin codes are not Reed–Solomon-like codes, while the square-code distinguisher is specifically designed to detect algebraic structure in such families of codes.

## Chapter 5

# Cryptosystems Based on Gabidulin Matrix Codes

In this chapter, we present a framework for code-based cryptosystems built upon Gabidulin matrix codes. The proposed construction combines and extends ideas introduced in [6] and [7].

Before describing the cryptographic constructions, we introduce two transformations of matrix codes that play a central role in our framework, namely the *enhancement* and the *disguise* of a matrix code.

**Enhanced code.** Let  $\mathcal{C} \subseteq \mathbb{F}_q^{M \times N}$  be a matrix code with basis  $(\mathbf{M}_1, \dots, \mathbf{M}_K)$ . Fix two non-negative integers  $\ell_1, \ell_2$ . For each  $i = 1, \dots, K$ , consider the matrix

$$\widetilde{\mathbf{M}}_i = \begin{pmatrix} \mathbf{M}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{pmatrix},$$

where

$$\mathbf{B}_i \in \mathbb{F}_q^{M \times \ell_2}, \quad \mathbf{C}_i \in \mathbb{F}_q^{\ell_1 \times N}, \quad \mathbf{D}_i \in \mathbb{F}_q^{\ell_1 \times \ell_2}$$

are sampled uniformly at random and independently for each  $i$ .

The matrix code generated by  $(\widetilde{\mathbf{M}}_1, \dots, \widetilde{\mathbf{M}}_K)$  is called the *enhanced code* of  $\mathcal{C}$ .

**Disguised code.** Let  $\mathcal{C} \subseteq \mathbb{F}_q^{M \times N}$  be a matrix code with basis  $(\mathbf{M}_1, \dots, \mathbf{M}_K)$  and let

$$\mathbf{P} \in \text{GL}(M, q), \quad \mathbf{Q} \in \text{GL}(N, q)$$

be invertible matrices sampled uniformly at random.

For each  $i = 1, \dots, K$ , define

$$\widehat{\mathbf{M}}_i := \mathbf{P} \mathbf{M}_i \mathbf{Q}.$$

The matrix code generated by  $(\widehat{\mathbf{M}}_1, \dots, \widehat{\mathbf{M}}_K)$  is called the *disguised code* of  $\mathcal{C}$ .

Using the previous transformations, we now describe a unified framework for cryptosystems based on Gabidulin matrix codes that employ the enhancement and disguise transformations defined above.

More precisely, our framework simultaneously incorporates:

- the Enhanced Matrix Codes Transformation-based scheme introduced in [7], and
- the subcode-based scheme proposed in [6].

The resulting cryptosystem is parametrized by  $(m, n, k, q, \ell_1, \ell_2, s)$  and recovers the constructions of [6] and [7] as special cases. In particular:

- setting  $s = 0$  yields the construction of [7];
- setting  $\ell_1 = \ell_2 = 0$  yields the construction of [6].

This unified description highlights the similarities between the two approaches and provides a flexible framework to explore the impact of the parameters on security in the context of Gabidulin matrix codes.

**Remark.** Throughout this chapter we consider only non-trivial Gabidulin codes, namely codes with parameters  $k < n$ .

## 5.1 McEliece Approach

---

**Algorithm 1** Key-generation algorithm
 

---

**Input:**  $m, n, k, q, \ell_1, \ell_2, s$ .

**Output:** pk, sk.

- 1: Select a random Gabidulin code  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$ , equipped with an efficient decoding algorithm that corrects up to  $t = \lfloor (n - k)/2 \rfloor$  errors.
- 2: Sample uniformly at random a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ :  $\mathcal{B} = (b_1, \dots, b_m)$ .
- 3: Compute a basis  $(\mathbf{A}_1, \dots, \mathbf{A}_{mk})$  of the code  $\text{ext}_{\mathcal{B}}(\mathcal{G})$ .
- 4: Sample uniformly at random the following matrices:

$$\mathbf{B}_i \in \mathbb{F}_q^{m \times \ell_2}, \quad \mathbf{C}_i \in \mathbb{F}_q^{\ell_1 \times n}, \quad \mathbf{D}_i \in \mathbb{F}_q^{\ell_1 \times \ell_2}, \quad i = 1, \dots, mk,$$

and the invertible matrices

$$\mathbf{P} \in \text{GL}(m + \ell_1, q), \quad \mathbf{Q} \in \text{GL}(n + \ell_2, q).$$

- 5: Construct the matrices  $(\mathbf{M}_1, \dots, \mathbf{M}_{mk})$  as follows:

$$(\mathbf{M}_1, \dots, \mathbf{M}_{mk}) := \left( \mathbf{P} \begin{pmatrix} \mathbf{A}_1 & \mathbf{B}_1 \\ \mathbf{C}_1 & \mathbf{D}_1 \end{pmatrix} \mathbf{Q}, \dots, \mathbf{P} \begin{pmatrix} \mathbf{A}_{mk} & \mathbf{B}_{mk} \\ \mathbf{C}_{mk} & \mathbf{D}_{mk} \end{pmatrix} \mathbf{Q} \right)$$

$$\mathbf{M}_i \in \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}, \quad i = 1, \dots, mk \quad \text{by construction.}$$

- 6: Consider the matrix code  $\mathcal{C}_{mat}$  generated by the basis  $(\mathbf{M}_1, \dots, \mathbf{M}_{mk})$  and compute a parity check matrix  $\mathbf{H} \in \mathbb{F}_q^{[(m+\ell_1)(n+\ell_2)-mk] \times (m+\ell_1)(n+\ell_2)}$  of  $\text{Unfold}(\mathcal{C}_{mat})$ .
- 7: Sample uniformly at random the matrix:

$$\mathbf{L} \in \mathbb{F}_q^{s \times (mk-s)},$$

construct the matrix  $\mathbf{H}'$  as follows:

$$\mathbf{H}' = \begin{pmatrix} \mathbf{L} & \mathbf{I}_s & \mathbf{0}_{s \times (mn-mk)} \\ & \mathbf{H} & \end{pmatrix} \in \mathbb{F}_q^{[(m+\ell_1)(n+\ell_2)-mk+s] \times (m+\ell_1)(n+\ell_2)},$$

consider the vector code  $\mathcal{C}'_{vec}$  generated by the parity check matrix  $\mathbf{H}'$  and compute a basis of  $\text{Fold}(\mathcal{C}'_{vec})$ :

$$(\mathbf{M}'_1, \dots, \mathbf{M}'_{mk-s}), \quad \mathbf{M}'_i \in \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}, \quad i = 1, \dots, mk - s.$$

- 8: **return** pk =  $(t, (\mathbf{M}'_1, \dots, \mathbf{M}'_{mk-s}))$ , sk =  $(\mathcal{B}, \mathbf{P}, \mathbf{Q}, \mathbf{g})$ .
-

**Note that** when the basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$  is chosen as

$$\mathcal{B} = (b_1, \dots, b_m) = (\alpha^0, \dots, \alpha^{m-1}), \text{ where } \alpha \in \mathbb{F}_{q^m} \text{ is a primitive element,}$$

one easy way to compute a basis  $(\mathbf{A}_1, \dots, \mathbf{A}_{mk})$  of the code  $\text{ext}_{\mathcal{B}}(\mathcal{G})$  is as follows.

Start from a basis of  $\mathcal{G}$ : if a generator matrix  $\mathbf{G}$  of  $\mathcal{G}$  is given, its rows form such a basis. By multiplying each row of  $\mathbf{G}$  by  $\alpha^i$  for  $i = 0, \dots, m-1$  and then applying the expansion map  $\text{ext}_{\mathcal{B}}$ , one obtains  $mk$  matrices in  $\mathbb{F}_q^{m \times n}$ , which form a basis of  $\text{ext}_{\mathcal{B}}(\mathcal{G})$ .

---

**Algorithm 2** Encryption algorithm

---

**Input:**  $\text{pk} = (t, (\mathbf{M}'_1, \dots, \mathbf{M}'_{mk-s}))$ , message  $\boldsymbol{\mu} \in \mathbb{F}_q^{mk-s}$ .

**Output:** ciphertext  $\mathbf{Y}$ .

- 1: Sample uniformly at random a matrix  $\mathbf{E} \in \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}$  such that  $\text{rank}(\mathbf{E}) \leq t$ .
  - 2: Compute  $\mathbf{Y} = \sum_{i=1}^{mk-s} \mu_i \mathbf{M}'_i + \mathbf{E} \in \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}$ .
  - 3: **return**  $\mathbf{Y}$ .
- 

---

**Algorithm 3** Decryption algorithm

---

**Input:**  $\text{pk}=(t, (\mathbf{M}'_1, \dots, \mathbf{M}'_{mk-s}))$ ,  $\text{sk}=(\mathcal{B}, \mathbf{P}, \mathbf{Q}, \mathbf{g})$ , ciphertext  $\mathbf{Y} \in \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}$ .

**Output:** Recovered message  $\hat{\boldsymbol{\mu}}$ .

- 1: Compute  $\tilde{\mathbf{Y}} := \mathbf{P}^{-1} \mathbf{Y} \mathbf{Q}^{-1}$ ,  $\tilde{\mathbf{Y}} \in \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}$ .
- 2: Erase the last  $\ell_1$  rows of the matrix  $\tilde{\mathbf{Y}}$  to obtain  $\tilde{\mathbf{Y}} \in \mathbb{F}_q^{m \times (n+\ell_2)}$ .
- 3: Compute  $\tilde{\mathbf{y}} := \text{ext}_{\mathcal{B}}^{-1}(\tilde{\mathbf{Y}})$ ,  $\tilde{\mathbf{y}} \in \mathbb{F}_{q^m}^{n+\ell_2}$ .
- 4: Erase the last  $\ell_2$  coordinates of the vector  $\tilde{\mathbf{y}}$  to obtain  $\hat{\mathbf{y}} \in \mathbb{F}_{q^m}^n$ .
- 5: Apply the decoding algorithm of  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$  on  $\hat{\mathbf{y}}$  to obtain  $\bar{\mathbf{e}}_1 \in \mathbb{F}_{q^m}^n$ .
- 6: Compute  $\bar{\mathbf{E}}_1 := \text{ext}_{\mathcal{B}}(\bar{\mathbf{e}}_1)$ ,  $\bar{\mathbf{E}}_1 \in \mathbb{F}_q^{m \times n}$ .
- 7: Solve the linear system:

$$\mathbf{Y} - \mathbf{P} \begin{pmatrix} \bar{\mathbf{E}}_1 & \bar{\mathbf{E}}_2 \\ \bar{\mathbf{E}}_3 & \bar{\mathbf{E}}_4 \end{pmatrix} \mathbf{Q} = \sum_{i=1}^{mk-s} \hat{\mu}_i \mathbf{M}'_i$$

where the unknowns are:  $\bar{\mathbf{E}}_2 \in \mathbb{F}_q^{m \times \ell_2}$ ,  $\bar{\mathbf{E}}_3 \in \mathbb{F}_q^{\ell_1 \times n}$ ,  $\bar{\mathbf{E}}_4 \in \mathbb{F}_q^{\ell_1 \times \ell_2}$ ,  $\hat{\boldsymbol{\mu}} \in \mathbb{F}_q^{mk-s}$ .

- 8: **return**  $\hat{\boldsymbol{\mu}}$ .
- 

**Lemma 6.** Let  $\boldsymbol{\mu} \in \mathbb{F}_q^{mk-s}$  be a message vector and let the matrix

$$\mathbf{Y} = \sum_{i=1}^{mk-s} \mu_i \mathbf{M}'_i + \mathbf{E} \in \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}$$

be the corresponding ciphertext. If the error matrix  $\mathbf{E} \in \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}$  satisfies

$$\text{rank}(\mathbf{E}) \leq t = \left\lfloor \frac{n-k}{2} \right\rfloor,$$

then the vector  $\bar{\mathbf{e}}_1$  obtained during the decryption procedure lies within the decoding radius of the Gabidulin code  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$ . Consequently, the decoding step succeeds and the McEliece decryption algorithm correctly recovers the original message vector  $\boldsymbol{\mu}$ .

*Proof.* Let the ciphertext be

$$\mathbf{Y} = \sum_{i=1}^{mk-s} \mu_i \mathbf{M}'_i + \mathbf{E},$$

where  $\text{rank}(\mathbf{E}) \leq t = \lfloor (n-k)/2 \rfloor$ . By construction of the public key, for each  $i = 1, \dots, mk-s$  we have

$$\mathbf{M}'_i = \mathbf{P} \begin{pmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{pmatrix} \mathbf{Q}.$$

Therefore,

$$\begin{aligned} \bar{\mathbf{Y}} &:= \mathbf{P}^{-1} \mathbf{Y} \mathbf{Q}^{-1} \\ &= \mathbf{P}^{-1} \left( \sum_{i=1}^{mk-s} \mu_i \mathbf{M}'_i + \mathbf{E} \right) \mathbf{Q}^{-1} \\ &= \sum_{i=1}^{mk-s} \mu_i \mathbf{P}^{-1} \mathbf{M}'_i \mathbf{Q}^{-1} + \mathbf{P}^{-1} \mathbf{E} \mathbf{Q}^{-1} \\ &= \sum_{i=1}^{mk-s} \mu_i \begin{pmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{pmatrix} + \bar{\mathbf{E}}, \end{aligned}$$

where we have defined

$$\bar{\mathbf{E}} := \mathbf{P}^{-1} \mathbf{E} \mathbf{Q}^{-1}.$$

Since  $\mathbf{P}$  and  $\mathbf{Q}$  are invertible, it follows that

$$\text{rank}(\bar{\mathbf{E}}) = \text{rank}(\mathbf{E}) \leq t.$$

Writing  $\bar{\mathbf{Y}}$  and  $\bar{\mathbf{E}}$  in block form,

$$\bar{\mathbf{Y}} = \left( \begin{array}{cc} \overbrace{\bar{\mathbf{Y}}_1}^n & \overbrace{\bar{\mathbf{Y}}_2}^{\ell_2} \\ \bar{\mathbf{Y}}_3 & \bar{\mathbf{Y}}_4 \end{array} \right) \left. \begin{array}{l} \} m \\ \} \ell_1 \end{array} \right\} \quad \bar{\mathbf{E}} = \left( \begin{array}{cc} \overbrace{\bar{\mathbf{E}}_1}^n & \overbrace{\bar{\mathbf{E}}_2}^{\ell_2} \\ \bar{\mathbf{E}}_3 & \bar{\mathbf{E}}_4 \end{array} \right) \left. \begin{array}{l} \} m \\ \} \ell_1 \end{array} \right\}.$$

we obtain

$$\begin{aligned} \begin{pmatrix} \bar{\mathbf{Y}}_1 & \bar{\mathbf{Y}}_2 \\ \bar{\mathbf{Y}}_3 & \bar{\mathbf{Y}}_4 \end{pmatrix} &= \sum_{i=1}^{mk-s} \mu_i \begin{pmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{pmatrix} + \begin{pmatrix} \bar{\mathbf{E}}_1 & \bar{\mathbf{E}}_2 \\ \bar{\mathbf{E}}_3 & \bar{\mathbf{E}}_4 \end{pmatrix} \\ &= \begin{pmatrix} \sum_{i=1}^{mk-s} \mu_i \mathbf{A}_i + \bar{\mathbf{E}}_1 & \sum_{i=1}^{mk-s} \mu_i \mathbf{B}_i + \bar{\mathbf{E}}_2 \\ \sum_{i=1}^{mk-s} \mu_i \mathbf{C}_i + \bar{\mathbf{E}}_3 & \sum_{i=1}^{mk-s} \mu_i \mathbf{D}_i + \bar{\mathbf{E}}_4 \end{pmatrix}. \end{aligned}$$

Removing the last  $\ell_1$  rows of  $\bar{\mathbf{Y}}$  yields

$$\tilde{\mathbf{Y}} := \begin{pmatrix} \bar{\mathbf{Y}}_1 & \bar{\mathbf{Y}}_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{mk-s} \mu_i \mathbf{A}_i + \bar{\mathbf{E}}_1 & \sum_{i=1}^{mk-s} \mu_i \mathbf{B}_i + \bar{\mathbf{E}}_2 \end{pmatrix} \in \mathbb{F}_q^{m \times (n+\ell_2)}.$$

Applying the inverse extension map  $\text{ext}_{\mathcal{B}}^{-1}$ , and using its  $\mathbb{F}_q$ -linearity, we obtain

$$\begin{aligned} \tilde{\mathbf{y}} &:= \text{ext}_{\mathcal{B}}^{-1}(\tilde{\mathbf{Y}}) \\ &= \text{ext}_{\mathcal{B}}^{-1} \left( \sum_{i=1}^{mk-s} \mu_i \mathbf{A}_i + \bar{\mathbf{E}}_1 \quad \sum_{i=1}^{mk-s} \mu_i \mathbf{B}_i + \bar{\mathbf{E}}_2 \right) \\ &= \left( \sum_{i=1}^{mk-s} \mu_i \text{ext}_{\mathcal{B}}^{-1}(\mathbf{A}_i) + \text{ext}_{\mathcal{B}}^{-1}(\bar{\mathbf{E}}_1), \quad \sum_{i=1}^{mk-s} \mu_i \text{ext}_{\mathcal{B}}^{-1}(\mathbf{B}_i) + \text{ext}_{\mathcal{B}}^{-1}(\bar{\mathbf{E}}_2) \right). \end{aligned}$$

We define

$$\begin{aligned} \mathbf{a}_i &:= \text{ext}_{\mathcal{B}}^{-1}(\mathbf{A}_i) \in \mathbb{F}_{q^m}^n, & \mathbf{b}_i &:= \text{ext}_{\mathcal{B}}^{-1}(\mathbf{B}_i) \in \mathbb{F}_{q^m}^{\ell_2}, \\ \bar{\mathbf{e}}_1 &:= \text{ext}_{\mathcal{B}}^{-1}(\bar{\mathbf{E}}_1) \in \mathbb{F}_{q^m}^n, & \bar{\mathbf{e}}_2 &:= \text{ext}_{\mathcal{B}}^{-1}(\bar{\mathbf{E}}_2) \in \mathbb{F}_{q^m}^{\ell_2}. \end{aligned}$$

Hence,

$$\tilde{\mathbf{y}} = \left( \sum_{i=1}^{mk-s} \mu_i \mathbf{a}_i + \bar{\mathbf{e}}_1, \quad \sum_{i=1}^{mk-s} \mu_i \mathbf{b}_i + \bar{\mathbf{e}}_2 \right) \in \mathbb{F}_{q^m}^{n+\ell_2}.$$

Discarding the last  $\ell_2$  coordinates of  $\tilde{\mathbf{y}}$  yields

$$\hat{\mathbf{y}} := \sum_{i=1}^{mk-s} \mu_i \mathbf{a}_i + \bar{\mathbf{e}}_1 \in \mathbb{F}_{q^m}^n.$$

By construction,  $(\mathbf{a}_1, \dots, \mathbf{a}_{mk-s})$  forms a basis of a subcode of the Gabidulin code  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$ . Since the vectors  $\mathbf{a}_1, \dots, \mathbf{a}_{mk-s}$  belong to a basis of  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$ , any linear combination of them is a codeword of  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$ . Moreover, since  $\text{rank}(\bar{\mathbf{E}}) \leq t$ , the error vector  $\bar{\mathbf{e}}_1$  has rank weight at most  $t$ . Therefore, since the secret key contains the vector of code locators  $\mathbf{g}$ , the receiver can apply the decoding algorithm of  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$  to  $\hat{\mathbf{y}}$ , which returns the error vector  $\bar{\mathbf{e}}_1$ .

Applying the extension map, we recover the first sub-block of the matrix  $\bar{\mathbf{E}}$

$$\bar{\mathbf{E}}_1 := \text{ext}_{\mathcal{B}}(\bar{\mathbf{e}}_1) \in \mathbb{F}_q^{m \times n}.$$

From the definition of  $\bar{\mathbf{E}}$ , it follows that  $\mathbf{E} = \mathbf{P}\bar{\mathbf{E}}\mathbf{Q}$ . Therefore,

$$\mathbf{Y} - \mathbf{E} = \sum_{i=1}^{mk-s} \mu_i \mathbf{M}'_i \iff \mathbf{Y} - \mathbf{P} \begin{pmatrix} \bar{\mathbf{E}}_1 & \bar{\mathbf{E}}_2 \\ \bar{\mathbf{E}}_3 & \bar{\mathbf{E}}_4 \end{pmatrix} \mathbf{Q} = \sum_{i=1}^{mk-s} \mu_i \mathbf{M}'_i.$$

At this stage,  $\bar{\mathbf{E}}_1$  is known, while

$$\bar{\mathbf{E}}_2 \in \mathbb{F}_q^{m \times \ell_2}, \quad \bar{\mathbf{E}}_3 \in \mathbb{F}_q^{\ell_1 \times n}, \quad \bar{\mathbf{E}}_4 \in \mathbb{F}_q^{\ell_1 \times \ell_2}, \quad \hat{\boldsymbol{\mu}} \in \mathbb{F}_q^{mk-s}$$

are unknown.

The number of *equations* in this linear system is equal to the total number of entries of the matrix  $\mathbf{Y}$  (or equivalently  $\mathbf{M}'_i$ ,  $i = 1, \dots, mk - s$ ), i.e.,

$$(m + \ell_1)(n + \ell_2) = n \boxed{m + m\ell_2 + \ell_1 n + \ell_1 \ell_2} \quad \text{number of equations.}$$

The number of *unknowns* is given by the sum of entries of the unknown matrices and the message vector:

$$m\ell_2 + \ell_1 n + \ell_1 \ell_2 + mk - s = k \boxed{m + m\ell_2 + \ell_1 n + \ell_1 \ell_2} - s \quad \text{number of unknowns.}$$

Since  $k < n \leq m$  and  $s$  is a positive integer, the number of unknowns is strictly smaller than the number of equations. Moreover, the system is consistent by construction. Therefore, the linear system admits a unique solution, which allows the recovery of the message vector  $\boldsymbol{\mu}$ .  $\square$

## 5.2 Niederreiter Approach

---

### Algorithm 4 Key-generation algorithm

---

**Input:**  $m, n, k, q, \ell_1, \ell_2, s$ .

**Output:** pk, sk.

- 1: Select a random Gabidulin code  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$ , equipped with an efficient decoding algorithm that corrects up to  $t = \lfloor (n - k)/2 \rfloor$  errors.
- 2: Sample uniformly at random a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ :  $\mathcal{B} = (b_1, \dots, b_m)$ .
- 3: Compute a basis  $(\mathbf{A}_1, \dots, \mathbf{A}_{mk})$  of the code  $\text{ext}_{\mathcal{B}}(\mathcal{G})$ .
- 4: Sample uniformly at random the following matrices:

$$\mathbf{B}_i \in \mathbb{F}_q^{m \times \ell_2}, \quad \mathbf{C}_i \in \mathbb{F}_q^{\ell_1 \times n}, \quad \mathbf{D}_i \in \mathbb{F}_q^{\ell_1 \times \ell_2}, \quad i = 1, \dots, mk,$$

and the invertible matrices

$$\mathbf{P} \in \text{GL}(m + \ell_1, q), \quad \mathbf{Q} \in \text{GL}(n + \ell_2, q).$$

- 5: Construct the matrices  $(\mathbf{M}_1, \dots, \mathbf{M}_{mk})$  as follows:

$$(\mathbf{M}_1, \dots, \mathbf{M}_{mk}) := \left( \mathbf{P} \begin{pmatrix} \mathbf{A}_1 & \mathbf{B}_1 \\ \mathbf{C}_1 & \mathbf{D}_1 \end{pmatrix} \mathbf{Q}, \dots, \mathbf{P} \begin{pmatrix} \mathbf{A}_{mk} & \mathbf{B}_{mk} \\ \mathbf{C}_{mk} & \mathbf{D}_{mk} \end{pmatrix} \mathbf{Q} \right)$$

$$\mathbf{M}_i \in \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}, \quad i = 1, \dots, mk \quad \text{by construction.}$$

- 6: Consider the matrix code  $\mathcal{C}_{mat}$  generated by the basis  $(\mathbf{M}_1, \dots, \mathbf{M}_{mk})$  and compute a parity check matrix  $\mathbf{H} \in \mathbb{F}_q^{[(m+\ell_1)(n+\ell_2)-mk] \times (m+\ell_1)(n+\ell_2)}$  of  $\text{Unfold}(\mathcal{C}_{mat})$ .
- 7: Sample uniformly at random the matrix:

$$\mathbf{L} \in \mathbb{F}_q^{s \times (mk-s)},$$

construct the matrix  $\mathbf{H}'$  as follows:

$$\mathbf{H}' = \begin{pmatrix} \mathbf{L} & \mathbf{I}_s & \mathbf{0}_{s \times (mn-mk)} \\ & \mathbf{H} & \end{pmatrix} \in \mathbb{F}_q^{[(m+\ell_1)(n+\ell_2)-mk+s] \times (m+\ell_1)(n+\ell_2)}.$$

- 8: **return** pk =  $(\mathbf{H}', t)$ , sk =  $(\mathcal{B}, \mathbf{P}, \mathbf{Q}, \mathbf{g})$ .
-

---

**Algorithm 5** Encryption algorithm

---

**Input:**  $\text{pk} = (\mathbf{H}', t)$ , message  $\boldsymbol{\mu} \in \mathbb{F}_q^{(m+\ell_1)(n+\ell_2)}$  such that  $\text{rank}(\mathbf{Fold}(\boldsymbol{\mu})) \leq t$ .

**Output:** ciphertext  $\mathbf{s}$ .

- 1: Compute  $\mathbf{s} = \boldsymbol{\mu} \mathbf{H}'^\top \in \mathbb{F}_q^{(m+\ell_1)(n+\ell_2)-mk+s}$ .
  - 2: **return**  $\mathbf{s}$ .
- 

---

**Algorithm 6** Decryption algorithm

---

**Input:**  $\text{pk} = (\mathbf{H}', t)$ ,  $\text{sk} = (\mathcal{B}, \mathbf{P}, \mathbf{Q}, \mathbf{g})$ , ciphertext  $\mathbf{s} \in \mathbb{F}_q^{(m+\ell_1)(n+\ell_2)-mk+s}$ .

**Output:** Recovered message  $\hat{\boldsymbol{\mu}}$ .

- 1: Find any  $\mathbf{y} \in \mathbb{F}_q^{(m+\ell_1)(n+\ell_2)}$  such that  $\mathbf{y} \mathbf{H}'^\top = \mathbf{s}$ .
- 2: Compute  $\mathbf{Y} := \mathbf{Fold}(\mathbf{y})$ ,  $\mathbf{Y} \in \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}$ .
- 3: Compute  $\tilde{\mathbf{Y}} := \mathbf{P}^{-1} \mathbf{Y} \mathbf{Q}^{-1}$ ,  $\tilde{\mathbf{Y}} \in \mathbb{F}_q^{(m+\ell_1) \times (n+\ell_2)}$ .
- 4: Erase the last  $\ell_1$  rows of the matrix  $\tilde{\mathbf{Y}}$  to obtain  $\tilde{\mathbf{Y}} \in \mathbb{F}_q^{m \times (n+\ell_2)}$ .
- 5: Compute  $\tilde{\mathbf{y}} := \text{ext}_{\mathcal{B}}^{-1}(\tilde{\mathbf{Y}})$ ,  $\tilde{\mathbf{y}} \in \mathbb{F}_{q^m}^{n+\ell_2}$ .
- 6: Erase the last  $\ell_2$  coordinates of the vector  $\tilde{\mathbf{y}}$  to obtain  $\hat{\mathbf{y}} \in \mathbb{F}_{q^m}^n$ .
- 7: Apply the decoding algorithm of  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$  on  $\hat{\mathbf{y}}$  to obtain  $\bar{\mathbf{e}}_1 \in \mathbb{F}_{q^m}^n$ .
- 8: Compute  $\bar{\mathbf{E}}_1 := \text{ext}_{\mathcal{B}}(\bar{\mathbf{e}}_1)$ ,  $\bar{\mathbf{E}}_1 \in \mathbb{F}_q^{m \times n}$ .
- 9: Solve the linear system:

$$\mathbf{s} = \text{Unfold} \left( \mathbf{P} \begin{pmatrix} \bar{\mathbf{E}}_1 & \bar{\mathbf{E}}_2 \\ \bar{\mathbf{E}}_3 & \bar{\mathbf{E}}_4 \end{pmatrix} \mathbf{Q} \right) \mathbf{H}'^\top$$

where the unknowns are:  $\bar{\mathbf{E}}_2 \in \mathbb{F}_q^{m \times \ell_2}$ ,  $\bar{\mathbf{E}}_3 \in \mathbb{F}_q^{\ell_1 \times n}$ ,  $\bar{\mathbf{E}}_4 \in \mathbb{F}_q^{\ell_1 \times \ell_2}$ .

- 10: **return**  $\hat{\boldsymbol{\mu}} = \text{Unfold} \left( \mathbf{P} \begin{pmatrix} \bar{\mathbf{E}}_1 & \bar{\mathbf{E}}_2 \\ \bar{\mathbf{E}}_3 & \bar{\mathbf{E}}_4 \end{pmatrix} \mathbf{Q} \right)$ .
- 

**Lemma 7.** Let  $\boldsymbol{\mu} \in \mathbb{F}_q^{(m+\ell_1)(n+\ell_2)}$  be a message vector and let the vector

$$\mathbf{s} = \boldsymbol{\mu} \mathbf{H}'^\top \in \mathbb{F}_q^{(m+\ell_1)(n+\ell_2)-mk+s}$$

be the corresponding ciphertext. If the message vector  $\boldsymbol{\mu}$  satisfies

$$\text{rank}(\mathbf{Fold}(\boldsymbol{\mu})) \leq t = \left\lfloor \frac{n-k}{2} \right\rfloor,$$

then the vector  $\bar{\mathbf{e}}_1$  obtained during the decryption procedure lies within the decoding radius of the Gabidulin code  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$ . Consequently, the decoding step succeeds and the Niederreiter decryption algorithm correctly recovers the message vector  $\boldsymbol{\mu}$ .

*Proof.* Let the ciphertext be

$$\mathbf{s} = \boldsymbol{\mu} \mathbf{H}'^\top,$$

where  $\boldsymbol{\mu} \in \mathbb{F}_q^{(m+\ell_1)(n+\ell_2)}$  satisfies

$$\text{rank}(\mathbf{Fold}(\boldsymbol{\mu})) \leq t = \left\lfloor \frac{n-k}{2} \right\rfloor.$$

Let us define the vector code

$$\mathcal{C}'_{vec} := \{ \mathbf{x} \in \mathbb{F}_q^{(m+\ell_1)(n+\ell_2)} \mid \mathbf{x} \mathbf{H}'^\top = 0 \},$$

that is, the linear code over  $\mathbb{F}_q$  whose parity-check matrix is  $\mathbf{H}'$ .

Now let  $\mathbf{y} \in \mathbb{F}_q^{(m+\ell_1)(n+\ell_2)}$  be any vector such that

$$\mathbf{y} \mathbf{H}'^\top = \mathbf{s}.$$

Then we have

$$(\mathbf{y} - \boldsymbol{\mu}) \mathbf{H}'^\top = \mathbf{y} \mathbf{H}'^\top - \boldsymbol{\mu} \mathbf{H}'^\top = \mathbf{s} - \mathbf{s} = 0.$$

Hence  $\mathbf{y} - \boldsymbol{\mu} \in \mathcal{C}'_{vec}$ , and therefore there exists  $\mathbf{c} \in \mathcal{C}'_{vec}$  such that

$$\mathbf{y} = \mathbf{c} + \boldsymbol{\mu}.$$

We now define

$$\mathbf{Y} := \mathbf{Fold}(\mathbf{y}), \quad \bar{\mathbf{Y}} := \mathbf{P}^{-1} \mathbf{Y} \mathbf{Q}^{-1}.$$

Using the linearity of the folding operator, we obtain

$$\begin{aligned} \bar{\mathbf{Y}} &= \mathbf{P}^{-1} \mathbf{Fold}(\mathbf{y}) \mathbf{Q}^{-1} \\ &= \mathbf{P}^{-1} \mathbf{Fold}(\mathbf{c} + \boldsymbol{\mu}) \mathbf{Q}^{-1} \\ &= \mathbf{P}^{-1} \mathbf{Fold}(\mathbf{c}) \mathbf{Q}^{-1} + \mathbf{P}^{-1} \mathbf{Fold}(\boldsymbol{\mu}) \mathbf{Q}^{-1}. \end{aligned}$$

Let  $(\mathbf{M}'_1, \dots, \mathbf{M}'_{mk-s})$  be a basis of  $\mathbf{Fold}(\mathcal{C}'_{vec})$ , i.e., of the matrix code obtained by folding the vectors of  $\mathcal{C}'_{vec}$ .

Since  $\mathbf{c} \in \mathcal{C}'_{vec}$ , there exist coefficients  $\lambda_1, \dots, \lambda_{mk-s} \in \mathbb{F}_q$  such that

$$\mathbf{Fold}(\mathbf{c}) = \sum_{i=1}^{mk-s} \lambda_i \mathbf{M}'_i.$$

Recalling how the matrices  $\mathbf{M}'_1, \dots, \mathbf{M}'_{mk-s}$  have been constructed, we obtain

$$\begin{aligned} \mathbf{P}^{-1} \mathbf{Fold}(\mathbf{c}) \mathbf{Q}^{-1} &= \mathbf{P}^{-1} \left( \sum_{i=1}^{mk-s} \lambda_i \mathbf{P} \begin{pmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{pmatrix} \mathbf{Q} \right) \mathbf{Q}^{-1} \\ &= \sum_{i=1}^{mk-s} \lambda_i \begin{pmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{pmatrix}. \end{aligned}$$

Let us define

$$\mathbf{E} := \mathbf{Fold}(\boldsymbol{\mu}), \quad \bar{\mathbf{E}} := \mathbf{P}^{-1} \mathbf{E} \mathbf{Q}^{-1}.$$

Combining the previous expressions, we obtain

$$\bar{\mathbf{Y}} = \sum_{i=1}^{mk-s} \lambda_i \begin{pmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{D}_i \end{pmatrix} + \bar{\mathbf{E}}.$$

Writing  $\bar{\mathbf{E}}$  in block form as

$$\bar{\mathbf{E}} = \begin{pmatrix} \bar{\mathbf{E}}_1 & \bar{\mathbf{E}}_2 \\ \bar{\mathbf{E}}_3 & \bar{\mathbf{E}}_4 \end{pmatrix},$$

we can express  $\bar{\mathbf{Y}}$  as

$$\bar{\mathbf{Y}} = \begin{pmatrix} \sum_{i=1}^{mk-s} \lambda_i \mathbf{A}_i + \bar{\mathbf{E}}_1 & \sum_{i=1}^{mk-s} \lambda_i \mathbf{B}_i + \bar{\mathbf{E}}_2 \\ \sum_{i=1}^{mk-s} \lambda_i \mathbf{C}_i + \bar{\mathbf{E}}_3 & \sum_{i=1}^{mk-s} \lambda_i \mathbf{D}_i + \bar{\mathbf{E}}_4 \end{pmatrix}.$$

The following step coincides with the previous McEliece proof of correctness of the decryption algorithm.

Discarding the last  $\ell_1$  rows of  $\bar{\mathbf{Y}}$ , we obtain the matrix

$$\tilde{\mathbf{Y}} := \left( \sum_{i=1}^{mk-s} \lambda_i \mathbf{A}_i + \bar{\mathbf{E}}_1 \quad \sum_{i=1}^{mk-s} \lambda_i \mathbf{B}_i + \bar{\mathbf{E}}_2 \right) \in \mathbb{F}_q^{m \times (n+\ell_2)}.$$

Applying the inverse extension map  $\text{ext}_{\mathcal{B}}^{-1}$ , we obtain

$$\tilde{\mathbf{y}} := \text{ext}_{\mathcal{B}}^{-1}(\tilde{\mathbf{Y}}) = \left( \sum_{i=1}^{mk-s} \lambda_i \mathbf{a}_i + \bar{\mathbf{e}}_1, \sum_{i=1}^{mk-s} \lambda_i \mathbf{b}_i + \bar{\mathbf{e}}_2 \right) \in \mathbb{F}_{q^m}^{n+\ell_2},$$

where  $\mathbf{a}_i := \text{ext}_{\mathcal{B}}^{-1}(\mathbf{A}_i)$  and  $\mathbf{b}_i := \text{ext}_{\mathcal{B}}^{-1}(\mathbf{B}_i)$ .

Discarding the last  $\ell_2$  coordinates yields

$$\hat{\mathbf{y}} := \sum_{i=1}^{mk-s} \lambda_i \mathbf{a}_i + \bar{\mathbf{e}}_1 \in \mathbb{F}_{q^m}^n.$$

Hence, as in the McEliece case,  $\hat{\mathbf{y}}$  is a codeword of  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$  corrupted by an error vector  $\bar{\mathbf{e}}_1$  of rank at most  $t$ . Since the secret key contains the locator vector  $\mathbf{g}$ , the decoding algorithm of  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$  recovers  $\bar{\mathbf{e}}_1$ .

Applying the extension map, we recover

$$\bar{\mathbf{E}}_1 := \text{ext}_{\mathcal{B}}(\bar{\mathbf{e}}_1) \in \mathbb{F}_q^{m \times n}.$$

Recalling that  $\mathbf{E} = \mathbf{Fold}(\boldsymbol{\mu})$  and  $\mathbf{E} = \mathbf{P} \bar{\mathbf{E}} \mathbf{Q}$ , we can write

$$\begin{aligned} \mathbf{s} &= \boldsymbol{\mu} \mathbf{H}'^{\top} \\ &= \mathbf{Unfold}(\mathbf{E}) \mathbf{H}'^{\top} \\ &= \mathbf{Unfold}(\mathbf{P} \bar{\mathbf{E}} \mathbf{Q}) \mathbf{H}'^{\top} \\ &= \mathbf{Unfold} \left( \mathbf{P} \begin{pmatrix} \bar{\mathbf{E}}_1 & \bar{\mathbf{E}}_2 \\ \bar{\mathbf{E}}_3 & \bar{\mathbf{E}}_4 \end{pmatrix} \mathbf{Q} \right) \mathbf{H}'^{\top}. \end{aligned}$$

We thus consider the following linear system:

$$\mathbf{s} = \text{Unfold} \left( P \begin{pmatrix} \bar{\mathbf{E}}_1 & \bar{\mathbf{E}}_2 \\ \bar{\mathbf{E}}_3 & \bar{\mathbf{E}}_4 \end{pmatrix} Q \right) \mathbf{H}'^\top.$$

At this stage,  $\bar{\mathbf{E}}_1$  is known, while the matrices

$$\bar{\mathbf{E}}_2 \in \mathbb{F}_q^{m \times \ell_2}, \quad \bar{\mathbf{E}}_3 \in \mathbb{F}_q^{\ell_1 \times n}, \quad \bar{\mathbf{E}}_4 \in \mathbb{F}_q^{\ell_1 \times \ell_2}$$

are unknown.

The number of *equations* in the linear system equals the length of  $\mathbf{s}$ :

$$\begin{aligned} (m + \ell_1)(n + \ell_2) - mk + s &= mn + m\ell_2 + \ell_1 n + \ell_1 \ell_2 - mk + s \\ &= m\ell_2 + \ell_1 n + \ell_1 \ell_2 + m(n - k) + s. \end{aligned}$$

The number of *unknowns* equals the total number of entries of the unknown matrices:

$$m\ell_2 + \ell_1 n + \ell_1 \ell_2.$$

Since  $k < n \leq m$  and  $s$  is a positive integer, the number of equations exceeds the number of unknowns. Moreover, the system is consistent by construction. Therefore, the linear system admits a unique solution, which allows the recovery of the message vector  $\boldsymbol{\mu}$ .  $\square$

### 5.3 Special Cases

We now show that our construction recovers the schemes of [7] and [6] as special cases.

#### Case 1: $s = 0$

When  $s = 0$ , Step 7 of the key-generation algorithm is still formally executed; however, the matrices involved in its construction (namely  $\mathbf{L}$ , the identity matrix, and the zero matrix) have 0 rows. Consequently, the step becomes void and no proper subcode is introduced.

In this case, the public code is exactly the enhanced and, then, disguised Gabidulin matrix code constructed in Steps 1–6.

More precisely, the construction coincides with [7], which:

- randomly chooses a Gabidulin code  $\mathcal{G}[\mathbf{g}; n, k]_{q^m}$ ;
- samples uniformly at random a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$ ;
- applies the matrix extension, obtaining a matrix code over  $\mathbb{F}_q$ ;

- performs the enhancement (addition of rows and columns to the matrices forming a basis of the matrix code);
- disguises the structure via left and right multiplication by invertible matrices;
- publishes either:
  - a basis of the resulting code (McEliece approach), or
  - a parity-check matrix of the resulting code (Niederreiter approach).

Therefore, for  $s = 0$ , our framework recovers exactly the scheme of [7]. In particular, the encryption and decryption algorithms (both in the McEliece and in the Niederreiter setting) coincide with those of [7].

### Case 2: $\ell_1 = \ell_2 = 0$

We now consider our construction with  $\ell_1 = \ell_2 = 0$ , i.e., without enhancement (no added rows or columns to the matrices that form a basis of the Gabidulin matrix code computed at Step 3).

At first glance, the scheme of [6] may appear quite different from our construction. However, when  $\ell_1 = \ell_2 = 0$ , the two schemes are in fact equivalent. We now explain this correspondence.

In both constructions:

- a Gabidulin code is randomly chosen;
- a basis of  $\mathbb{F}_{q^m}/\mathbb{F}_q$  is sampled uniformly at random.

Then:

- **Our construction:** applies the matrix extension, obtaining a Gabidulin matrix code  $\text{ext}_{\mathcal{B}}(\mathcal{G})$ .
- **Their construction:** considers the  $q$ -ary image of the Gabidulin code, namely  $\Phi_{\mathcal{B}}(\mathcal{G})$ , which, as discussed in Section 3.2, is an  $\mathbb{F}_q$ -representation equivalent to the representation of the Gabidulin matrix code used in our construction, since there exists a bijection between  $\Phi_{\mathcal{B}}(\mathcal{G})$  and  $\text{ext}_{\mathcal{B}}(\mathcal{G})$ .

Next:

- **Our construction:** performs an enhancement step which is void when  $\ell_1 = \ell_2 = 0$ , so we remain at the same stage as their construction.
- **Both constructions:** disguise the structure via left and right multiplication by invertible matrices and compute the parity-check matrix of the resulting vector code which corresponds to  $\mathbf{Unfold}(\mathcal{C}_{mat})$  in our case.

Finally:

- **Both constructions:** add rows to that parity-check matrix in a specific way (via the matrix  $\mathbf{L}$ , the identity matrix, and the all-zero matrix), and publish the resulting matrix, which is a parity-check matrix representing a subcode of the code obtained in the previous step.

Hence, when  $\ell_1 = \ell_2 = 0$ , our framework reduces exactly to the subcode-based construction of [6].

An important contribution of [6] is the description of a practical and explicit procedure to pass

- from a generator matrix of an  $\mathbb{F}_{q^m}$ -linear code,
- to a generator matrix of the unfolded disguised matrix code obtained by expanding the original code over  $\mathbb{F}_q$ .

This applies in particular to our setting, since a Gabidulin code is  $\mathbb{F}_{q^m}$ -linear and this procedure corresponds precisely to an explicit implementation of Steps 1–6 of the key-generation algorithm in the non-enhanced case  $\ell_1 = \ell_2 = 0$ .

The procedure to obtain a generator matrix of the vector expansion of an  $\mathbb{F}_{q^m}$ -linear code is described in the subsection on the generator matrix of the vector expansion (see Section 3.2). Applying this procedure to a Gabidulin code yields the generator matrix  $\mathbf{G}_q$  of the expanded Gabidulin code.

The next step consists in disguising the expanded Gabidulin code. In particular, a generator matrix  $\mathbf{G}'_q$  of the disguised expanded Gabidulin code can be computed as

$$\mathbf{G}'_q = \mathbf{G}_q(\mathbf{Q} \otimes \mathbf{P}^\top) \in \mathbb{F}_q^{mk \times mn}.$$

A parity-check matrix  $\mathbf{H}'_q$  of the disguised expanded Gabidulin code can then be obtained by computing a basis of the dual code, i.e., a matrix  $\mathbf{H}'_q$  such that

$$\mathbf{G}'_q \mathbf{H}'_q{}^\top = 0.$$

## Parameter Sets

Since our construction recovers the schemes of [7] and [6] as special cases, the parameter sets proposed in these works can be interpreted within our unified framework.

More precisely,

- when  $s = 0$ , our scheme coincides with the construction of [7];
- when  $\ell_1 = \ell_2 = 0$ , it reduces to the subcode-based scheme of [6].

Table 5.1 collects the parameter sets proposed in these two works and expresses them using the notation of our construction. In particular, the table combines the parameter sets from Fig. 5 and Fig. 6 of [7] and from Table 1 and Table 2 of [6].

For each security level, the table reports the base field size  $q$ , the extension degree  $m$ , the dimension  $k$  of the underlying Gabidulin code, and the parameters  $s$ ,  $\ell_1$ , and  $\ell_2$  defining the subcode and enhancement steps in our general framework. We also include the resulting public-key size (pk) and ciphertext size (ct), as reported in the original works.

Sec.	q	m	k	s	$\ell_1$	$\ell_2$	pk	ct
128	2	32	16	32	0	0	31.8 KB	1024 B
	2	34	18	34	0	0	20.3 KB	1156 B
	2	36	16	36	0	0	15.7 KB	1296 B
	2	36	16	36	0	0	11.3 KB	1296 B
	2	40	15	40	0	0	11.7 KB	1600 B
128	2	37	17	0	3	3	76 KB	121 B
	2	37	25	0	3	3	78 KB	84 B
	2	43	35	0	2	2	98 KB	65 B
	2	53	47	0	2	2	166 KB	66 B
	2	37	17	0	4	0	70 KB	111 B
	16	23	13	0	1	1	41 KB	138 B
	16	23	7	0	0	5	33 KB	207 B
192	2	59	51	0	2	2	268 KB	89 B
	2	43	23	0	5	0	133 KB	134 B
	2	47	33	0	5	0	173 KB	111 B
	2	53	41	0	4	0	230 KB	106 B
256	2	46	24	46	0	0	136.6 KB	2116 B
	2	48	23	48	0	0	79.7 KB	2304 B
	2	48	24	48	0	0	53.1 KB	2304 B
	2	52	21	52	0	0	49.0 KB	2704 B
	2	55	21	55	0	0	46.0 KB	3025 B
256	2	47	23	0	3	3	191 KB	177 B
	2	53	37	0	3	2	274 KB	139 B
	2	79	71	0	2	2	667 KB	119 B
	16	29	9	0	2	1	87 KB	334 B
	16	29	17	0	2	1	107 KB	218 B

**Table 5.1:** Parameter sets from [6] and [7]. Gray rows correspond to [6] ( $\ell_1 = \ell_2 = 0$ ), while blue rows correspond to [7] ( $s = 0$ ).

A subsequent work, following [7], by Porwal, Wachter-Zeh, and Loidreau [21] revisits the security analysis of the parameter sets proposed in [7]. In their work, three different attacks are considered: the distinguisher originally proposed in [7], a key-recovery attack proposed by the authors, and a naive attack.

For each parameter set, the workfactor of all three attacks is estimated. Their analysis shows that several parameter sets do not allow to achieve the intended security level, as the resulting workfactor falls below it.

In Table 5.1, the parameter sets affected by this issue are highlighted in blue. These correspond to the instances for which the workfactor estimated in [21] falls below the claimed security level for at least one of the considered attacks.

## Security Assumptions

The security of the cryptosystems presented in this chapter relies on the computational hardness of certain decoding problems for linear codes.

In the classical setting of vector codes, the security of McEliece-like cryptosystems is based on the difficulty of the *decoding problem* and of its equivalent formulation, the *syndrome decoding problem*.

**Decoding problem.** Given a generator matrix  $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ , a received vector  $\mathbf{y} \in \mathbb{F}_q^n$ , and a positive integer  $t$ , the problem consists in determining whether there exist a message vector  $\mathbf{x} \in \mathbb{F}_q^k$  and an error vector  $\mathbf{e} \in \mathbb{F}_q^n$  with Hamming weight  $\text{wt}_H(\mathbf{e}) \leq t$  such that

$$\mathbf{y} = \mathbf{x}\mathbf{G} + \mathbf{e}.$$

An equivalent formulation of this problem can be expressed in terms of parity-check matrices and syndromes, leading to the following problem.

**Syndrome decoding problem.** Given a parity-check matrix  $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ , a syndrome vector  $\mathbf{s} \in \mathbb{F}_q^{n-k}$ , and a positive integer  $t$ , the problem consists in determining whether there exists a vector  $\mathbf{e} \in \mathbb{F}_q^n$  such that

$$\text{wt}_H(\mathbf{e}) \leq t \quad \text{and} \quad \mathbf{e}\mathbf{H}^T = \mathbf{s}.$$

When considering matrix codes and the rank metric, the analogous problems are the *MinRank problem* and the *MinRank-syndrome problem*. The MinRank problem is known to be NP-hard, and the MinRank-syndrome problem is equivalent to it.

**MinRank problem.** Given the matrices  $\mathbf{M}_1, \dots, \mathbf{M}_K \in \mathbb{F}_q^{m \times n}$ , a received matrix  $\mathbf{Y} \in \mathbb{F}_q^{m \times n}$ , and a positive integer  $t$ , the problem consists in finding a message vector  $\mathbf{x} \in \mathbb{F}_q^K$  such that

$$\mathbf{Y} = \sum_{i=1}^K x_i \mathbf{M}_i + \mathbf{E}, \quad \mathbf{E} \in \mathbb{F}_q^{m \times n}, \quad \text{rank}(\mathbf{E}) \leq t.$$

Similarly to the Hamming-metric setting, this problem can also be expressed in terms of syndromes.

**MinRank-syndrome problem.** Given a parity-check matrix  $\mathbf{H} \in \mathbb{F}_q^{(mn-K) \times mn}$ , a syndrome vector  $\mathbf{s} \in \mathbb{F}_q^{mn-K}$ , and a positive integer  $t$ , the problem consists in finding an error vector  $\mathbf{e} \in \mathbb{F}_q^{mn}$  such that

$$\text{rank}(\mathbf{Fold}(\mathbf{e})) \leq t \quad \text{and} \quad \mathbf{s} = \mathbf{e}\mathbf{H}^\top.$$

# Chapter 6

## Conclusion

This thesis studied cryptographic constructions based on Gabidulin codes and their expansion over the base field. The main motivation is the design of secure code-based cryptosystems in the context of post-quantum cryptography, where hiding the strong algebraic structure of Gabidulin codes remains a central challenge.

After introducing the necessary background on the rank metric and Gabidulin codes, particular attention was devoted to algebraic attacks against structured codes. In particular, Overbeck's attack and the square-code distinguisher were analyzed in detail, highlighting how the algebraic structure of Gabidulin codes can be exploited to distinguish them from random codes and even recover the secret key. This analysis emphasizes the importance of masking the underlying  $\mathbb{F}_{q^m}$ -linear structure when Gabidulin codes are used in cryptographic constructions.

The main part of the thesis focused on the study and comparison of two recent proposals of cryptosystems based on Gabidulin codes expanded over  $\mathbb{F}_q$ . A detailed analysis of their key generation, encryption, and decryption procedures showed that, although the two constructions share several ideas, they differ in important aspects of the key generation process. In particular, one proposal introduces additional rows and columns in the matrices defining the matrix code obtained from the expansion, while the other derives the public key by taking a suitable subcode of a distorted expanded code.

Based on this analysis, a unified cryptosystem was proposed that combines these two approaches. The construction introduces parameters controlling the number of additional rows and columns and the dimension reduction obtained by taking a subcode. Both cryptosystems from the literature can be recovered as special cases of this framework for specific parameter choices. The scheme was described in detail, including key generation, encryption, and decryption algorithms in both

the McEliece and Niederreiter variants, together with a proof of correctness of the decoding procedure. A unified comparison of parameter sets from the literature was also presented, highlighting parameter choices that may lead to insecure instances according to known attacks.

Several directions for future work arise naturally from this study. One important open problem is the existence of distinguishers for Gabidulin matrix codes. To the best of our knowledge, no distinguisher specifically targeting this class of codes is currently known. Investigating whether techniques similar to those used for other structured codes, such as expanded Reed–Solomon codes, can be adapted to this setting would contribute to a better understanding of the security of these constructions.

Another natural direction for future work would be the practical implementation of the cryptosystem proposed in this thesis. In particular, it would be interesting to implement the construction that combines the two approaches from the literature: starting from a Gabidulin matrix code obtained by expanding a Gabidulin code over  $\mathbb{F}_{q^m}$ , one could first modify the matrices forming a basis of the matrix code by adding suitable rows and columns, as proposed in one of the schemes, then apply the usual masking transformations using invertible matrices  $(\mathbf{P})$  and  $(\mathbf{Q})$ , and finally derive the public code by considering an appropriate subcode of the resulting distorted code. Such an implementation would make it possible to experimentally analyze the resulting public keys and test them against known distinguishers and structural attacks, in order to evaluate the practical security of the construction and estimate the computational complexity required to break the system for different parameter choices.

# Bibliography

- [1] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. «Ideals over a Non-Commutative Ring and their Application in Cryptology». In: *Advances in Cryptology — EUROCRYPT '91*. Ed. by Donald W. Davies. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 482–489 (cit. on pp. 1, 4).
- [2] Keith Gibson. «The Security of the Gabidulin Public Key Cryptosystem». In: *Advances in Cryptology — EUROCRYPT '96*. Ed. by Ueli Maurer. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 212–223 (cit. on p. 1).
- [3] E.M. Gabidulin and A.V. Ourivski. «Modified GPT PKC with Right Scrambler». In: *Electronic Notes in Discrete Mathematics* 6 (2001). WCC2001, International Workshop on Coding and Cryptography, pp. 168–177 (cit. on p. 1).
- [4] A. V. Ourivski and E. M. Gabidulin. «Column scrambler for the GPT cryptosystem». In: *Discrete Appl. Math.* 128.1 (May 2003), pp. 207–221 (cit. on p. 1).
- [5] E. M. Gabidulin, A. V. Ourivski, B. Honary, and B. Ammar. «Reducible rank codes and their applications to cryptography». In: *IEEE Trans. Inf. Theor.* 49.12 (Dec. 2003), pp. 3289–3293 (cit. on p. 1).
- [6] Thierry P. Berger, Philippe Gaborit, and Olivier Ruatta. «Gabidulin Matrix Codes and Their Application to small Ciphertext Size Cryptosystems». In: *Progress in Cryptology – INDOCRYPT 2017*. Ed. by Arpita Patra and Nigel P. Smart. Cham: Springer International Publishing, 2017, pp. 247–266 (cit. on pp. 1, 2, 19, 22, 34, 35, 45–48).
- [7] Nicolas Aragon, Alain Couvreur, Victor Dörsner, Philippe Gaborit, and Adrien Vinçotte. «MinRank Gabidulin Encryption Scheme on Matrix Codes». In: *Advances in Cryptology – ASIACRYPT 2024*. Ed. by Kai-Min Chung and Yu Sasaki. Singapore: Springer Nature Singapore, 2025, pp. 68–100 (cit. on pp. 2, 34, 35, 45–49).

- [8] Nicolas Sendrier. «McEliece Public Key Cryptosystem». In: *Encyclopedia of Cryptography and Security*. Ed. by Henk C. A. van Tilborg and Sushil Jajodia. Boston, MA: Springer US, 2011, pp. 767–768 (cit. on p. 3).
- [9] Philippe Gaborit, Olivier Ruatta, and Julien Schrek. «On the Complexity of the Rank Syndrome Decoding Problem». In: *IEEE Transactions on Information Theory* 62.2 (2016), pp. 1006–1019 (cit. on p. 4).
- [10] Pierre Loidreau. «A New Rank Metric Codes Based Encryption Scheme». In: *Post-Quantum Cryptography*. Ed. by Tanja Lange and Tsuyoshi Takagi. Cham: Springer International Publishing, 2017, pp. 3–17 (cit. on pp. 4, 5).
- [11] Ernst M. Gabidulin, Haitham Rashwan, and Bahram Honary. «On improving security of GPT cryptosystems». In: *2009 IEEE International Symposium on Information Theory*. 2009, pp. 1110–1114. DOI: 10.1109/ISIT.2009.5206029 (cit. on p. 4).
- [12] R. Overbeck. «Structural Attacks for Public Key Cryptosystems based on Gabidulin Codes». In: *Journal of Cryptology* 21 (Apr. 2008), pp. 280–301 (cit. on pp. 4, 25, 29).
- [13] Alain Couvreur, Philippe Gaborit, Valérie Gauthier-Umaña, Ayoub Otmani, and Jean-Pierre Tillich. «Distinguisher-based attacks on public-key cryptosystems using Reed–Solomon codes». In: *Designs, Codes and Cryptography* 73.2 (2014), pp. 641–666 (cit. on p. 4).
- [14] Christian Wieschebrink. «Two NP-complete problems in coding theory with an application in code based cryptography». In: *2006 IEEE International Symposium on Information Theory*. IEEE. 2006, pp. 1733–1737 (cit. on p. 4).
- [15] Jean-Charles Faugère, Valérie Gauthier, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. «A Distinguisher for High Rate McEliece Cryptosystems». In: (2010) (cit. on p. 4).
- [16] Philippe Gaborit, Gaétan Murat, Olivier Ruatta, and Gilles Zémor. «Low rank parity check codes and their application to cryptography». In: *Proceedings of the Workshop on Coding and Cryptography WCC*. Vol. 2013. 2013 (cit. on p. 5).
- [17] Michael Alekhnovich. «More on average case vs approximation complexity». In: *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings*. IEEE. 2003, pp. 298–307 (cit. on p. 5).
- [18] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, and Gilles Zémor. «Rank quasi-cyclic (RQC)». In: (2017) (cit. on p. 5).

- [19] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loic Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Gilles Zémor, and I Bourges. «Hamming quasi-cyclic (HQC)». In: *NIST PQC Round 2.4* (2018), p. 13 (cit. on p. 5).
- [20] Ernst Gabidulin. «Theory of codes with maximum rank distance (translation)». In: *Problems of Information Transmission* 21 (Jan. 1985), pp. 1–12 (cit. on p. 11).
- [21] Anmoal Porwal, Antonia Wachter-Zeh, and Pierre Loidreau. «Key Attack on the ACDGV Matrix Encryption Scheme». In: (2025) (cit. on p. 49).