Rank Metric Codes, Codes Using Linear Complexity and Applications to Public Key Cryptosystems

---

Dissertation

zur

Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)

vorgelegt der

Mathematisch-naturwissenschaftlichen Fakultät

der

Universität Zürich

von

Tovohery Hajatiana Randrianarisoa

aus

Madagaskar

Promotionskommission

Prof. Dr. Joachim Rosenthal (Vorsitz)
Prof. Dr. Valentin Féray
Prof. Dr. Andrew Kresch

Zürich, 2018

Abstract

Nowadays, the most widely used cryptosystems are based on either the difficulty of factoring integers or the difficulty of computing discrete logarithms in some group. These cryptosystems are threatened by the quantum computers. Among the candidate cryptosystems for the post-quantum era are the code based cryptosystems. These cryptosystems were originally designed with the use of linear Hamming metric codes. Further proposals were presented where the codes are replaced with rank metric codes.

A good class of codes suited for cryptography are the maximum rank distance (MRD) codes. A particular example are the Gabidulin codes. But cryptosystems using them were shown to be vulnerable against some attack. In the first part of this thesis, we show that when we work in large field, most linear rank metric codes are maximum rank distance codes. Furthermore, only a few of them are Gabidulin codes. We confirm this result by a construction using derivation on polynomial rings. Another class, discovered by Sheekey, are the twisted Gabidulin codes.

The second part of this thesis is devoted to the decoding algorithms for the class of twisted Gabidulin codes. One of the algorithm is completely new, even when we apply it to the class of Gabidulin codes. Namely, with some interpolation of linearized polynomials we can modify the problem of decoding in rank metric to the problem of finding the linear feedback-shift register with least order which generates some sequence. The idea here is that the rank of a vector is equivalent to some linear complexity. This gives us the idea of working further with linear feedback shift register.

In the last part of the thesis, we focus on this notion of linear complexity. However, we switch back to the setting with Hamming metric. In fact, the linear complexity of some sequence with fixed period corresponds to the Hamming weight of some vector. We then generalize this result by considering the linear complexity of some sequence where the period is arbitrary. We construct a new metric by using the linear complexity of these arbitrary sequences. We develop a coding theory with this linear complexity metric and we show some results like a construction of optimal codes. We will give a formula, with proof, of the number of sequences with given linear complexity. Having this new theory, finally, we consider the codes based on the linear complexity and we design a new cryptosystem using these codes. As a basis for the security of the cryptosystem, we will show that decoding random linear codes with respect to the linear complexity metric is a difficult problem.

Zusammenfassung

Die kryptografischen Systeme, die heutzutage am häufigsten verwendet werden, basieren entweder auf der (mathematischen) Schwierigkeit des Faktorisierens ganzer Zahlen oder der Berechnung des diskreten Logarithmus in einer Gruppenstruktur. In der Erwartung der Quantumcomputern sind diese kryptografischen Systeme jedoch bedroht. Mögliche Kandidaten für sogenannte post-quantum Kryptosysteme, sind unter anderem Systeme basierend auf der Theorie von Codes. Ursprünglich verwenden diese Systeme lineare Codes der Hamming Metrik, indessen gibt es auch Vorschläge für Systeme, welche eine andere Metrik, nämlich die Rang Metrik in Betracht ziehen.

Dabei haben sich die Codes mit maximaler Rang Distanz (MRD) als eine gute Wahl für kryptografische Zwecke erwiesen. Ein wichtiges Beispiel solcher Codes sind die Gabidulin Codes. Bedauerlicherweise erwiesen sich diese Systeme als anfällig gegen Angriffe. In dem ersten Teil dieser Arbeit zeigen wir, dass in grossen endlichen Körper die Eigenschaft maximalen Rang Distanz zu besitzen, generisch ist. Des Weiteren, sind nur wenige dieser MRD Codes Gabidulin Codes. Dieses Resultat unterlegen wir durch eine Konstruktion von solchen Codes mit Hilfe von Ableitungen in Polynomringen. Der sogenannte gedrehte ("twisted") Gabidulin Code ist ein weiterer MRD Code und wurde erst kürzlich von Sheekey entdeckt.

Im zweiten Teil dieser Arbeit konzentrieren wir uns auf Dekodierungs Algorithmen für gedrehte Gabidulin Codes. Einer dieser Algorithmen verwendet eine vollständig neue Idee, auch wenn wir ihn auf die Klasse der Gabidulin Codes anwenden. Genauer gesagt, können wir mit der Interpolation von linearisierten Polynomen das Dekodierungs Problem im Rang Metrik Fall übersetzen in das Finden eines linearen Feedback-Shift Register mit dem kleinsten Grad, der eine gegebene Folge generiert. Dies basiert auf der Idee, dass der Rang eines Vektors einer linearen Komplexität entspricht. Daher betrachten wir auch im Weiteren lineare Feedback-Shift Register.

Im letzten Teil dieser Arbeit fokussieren wir uns auf den Begriff der linearen Komplexität. Hierfür betrachten wir jedoch wieder die Hamming Metrik. Tatsächlich stimmt die lineare Komplexität von einigen Folgen mit gegebener Periode überein mit dem Hamming Gewicht eines Vektors. Wir verallgemeinern dieses Resultat durch in Betrachtziehung der linearen Komplexität einer Folge mit beliebiger Periode. In diesem Sinne konstruieren wir eine neue Metrik mittels der linearen Komplexität der beliebigen Folgen. Wir etablieren eine vollständige Theorie um die Codes dieser neuen Metrik, unter anderem konstruieren wir optimale Codes der linearen Komplexitäts Metrik. Wir bestimmen (und beweisen) die Anzahl der Folgen mit einer gegebenen linearen Komplexität. Mit dieser neuen Theorie können wir letztlich ein neues kryptografisches System vorschla-

gen, welches lineare Komplexitäts Codes verwendet. Bezüglich der Sicherheit dieses kryptografischen Systems zeigen wir, dass das Dekodieren eines zufälligen linearen Komplexitäts Codes ein schwieriges Problem ist.

Acknowledgments

First of all, I would like to give my gratitude to Prof. Joachim Rosenthal for giving me the opportunity to do a PhD under his supervision. I thank him for all his patience, support and especially all his help guiding me through my work.

I thank the reviewers for their invaluable comments to improve this thesis.

I also would like to acknowledge Prof. Rosenthal, Anna-Lena, Alessandro for their work with me in the co-authored papers.

I am also grateful to Alessandro, Karan, Kyle, Reto, Anna-Lena, Violetta, Gianira for all the discussion while working on mathematical research and for any help in writing this thesis.

Without the support from the Zurich Graduate School of Mathematics and the Swiss National Science Foundation, this thesis would not have started nor ended. I am very grateful for that.

Finally, my gratitude goes to my Parents, family and all my friends whether they are in Switzerland or in Madagascar.

Tovo

Contents

Chapter 1

Introduction

Nowadays, data transmission is a very important aspect of our everyday life. Two important aspects are the concepts of security and reliability. Security can be addressed by cryptography while if we want reliability, we can improve it by using coding theory. The area of cryptography and coding theory are somewhat contradictory. In cryptography, we want to hide transmission from everyone except the intended receiver by using a cryptosystem, while with coding theory, we want everyone to be able to receive the message by using codes. However, they share a common aspect: they are all about the transmission of information. Linear codes can be used to adress them. First of all, coding theory has applications in data storage, data transmission and network coding. Furthermore a lot of cryptosystems were provided where the underlying security is based on the difficulty of some problems in coding theory. The advantage of using these code based cryptosystems is that they are quantum computer resistant i.e. there is not yet any efficient attack against cryptosystems based on linear codes. In opposite, most of the cryptosytems used nowadays are based on the difficulty of factoring integers (RSA for example) and the difficulty of solving the discrete logarithm problem in some group (Elliptic curves for example). These cryptosystems are not secure since there are already some algorithms to break them in case a powerful quantum computer would become available [BBD08]. And thus finding alternative cryptosystems and codes are important.

This chapter is a brief introduction about the tools needed in this thesis. First we will give brief results from finite field theory and then we look at the concept of linear codes. And finally, we will show how to get cryptosystems using linear codes.

1.1 Finite field and vector spaces

Most constructions of linear codes are done over a finite field. In this section, we recall some results which we will need later in this thesis.

Definition 1. Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ be a finite field extension of degree m. The trace of $\alpha \in \mathbb{F}_{q^m}$ over $\mathbb{F}_q$ is given by

$$\mathbf{Tr}(\alpha) = \sum_{i=0}^{m-1} \alpha^{q^i}.$$

The norm of $\alpha \in \mathbb{F}_{q^m}$ over $\mathbb{F}_q$ is defined by

$$\mathbf{N}(\alpha) = \prod_{i=0}^{m-1} \alpha^{q^i}.$$

The trace map satisfies the following properties.

Lemma 1.

(i) $\mathbf{Tr}$ *is a surjective* $\mathbb{F}_q$*-linear map* $\mathbb{F}_{q^m} \to \mathbb{F}_q$.

(ii) All $\mathbb{F}_q$*-linear maps* $\mathbb{F}_{q^m} \to \mathbb{F}_q$ *are of the form* $x \mapsto \mathbf{Tr}(\alpha x)$ *for some* $\alpha \in \mathbb{F}_{q^m}$.

Suppose that s, $0 < s < m$ is an integer satisfying $\gcd(m, s) = 1$. We set the map ϕ_s as

$$\begin{aligned} \phi_s : \mathbb{F}_{q^m} &\to \mathbb{F}_{q^m} \\ \alpha &\mapsto \alpha^{q^s} - \alpha. \end{aligned}$$

This map is connected to the trace map in the following way.

Lemma 2.

(i) If s *is an integer such that* $\gcd(m, s) = 1$*, then* $\phi_s(\alpha) = 0$ *if and only if* $\alpha \in \mathbb{F}_q$.

(ii) $\operatorname{Ker} \mathbf{Tr} = \operatorname{Im} \phi_s$ *for* $\gcd(s, m) = 1$ *and it has* q^{m-1} *elements.*

Lemma 1 and Lemma 2 are standard results in finite field theory. For a proof, one can have a look at [LN96], Chapter 2, Section 3.

Now suppose that $\mathbf{GL}_n(q)$ is the set of all invertible square matrices in $\mathbb{F}_q^{n\times n}$. By simple counting, the number of k dimensional subspaces of $\mathbb{F}_q^n$ is given by,

$$\binom{n}{k}_q := \prod_{i=0}^{k-1} \frac{q^n - q^i}{q^k - q^i} = \frac{\prod_{i=0}^{k-1} (q^n - q^i)}{\sharp \mathbf{GL}_k(q)}.$$

And in the same way, one can show the following Lemma. A full proof can be found in [NHTRR18].

Lemma 3. *Let k, n be two positive integers with $k \leq \frac{n}{2}$ and let $\mathbf{U}$ be a k-dimensional vector subspace of $\mathbb{F}_q^n$. Then for arbitrary $r = 0, \cdots, k$, the number of k-dimensional subspaces intersecting with $\mathbf{U}$ in a $(k-r)$-dimensional subspace is*

$$\begin{pmatrix} k \\ k-r \end{pmatrix}_q \begin{pmatrix} n-k \\ r \end{pmatrix}_q q^{r^2}.$$

1.2 Hamming metric codes

Let q be a power of a prime. In this section, $\mathbb{F}_q$ denotes a finite field with q elements.

Definition 2. A code $\mathcal{C}$ of length n over $\mathbb{F}_q$ is a subset of $\mathbb{F}_q^n$. $\mathcal{C}$ is called a linear code if $\mathcal{C}$ is a subspace of $\mathbb{F}_q^n$ over $\mathbb{F}_q$ and if the dimension of $\mathcal{C}$ is k we describe the parameters of $\mathcal{C}$ as $[n, k]$.

Messages are sent as vectors $\mathbf{x} \in \mathcal{C}$ over the network and errors may happen. The errors are elements of $\mathbb{F}_q^n$ and they are quantified by the *weight*. Let $\mathbf{0}$ denote the vector whose elements are all zero.

Definition 3. Let $\mathbf{x} \in \mathbb{F}_q^n$. The weight of $\mathbf{x} = (x_1, \cdots, x_n)$ is defined by

$$w(\mathbf{x}) = \sharp\{i : \ x_i \neq 0, 1 \leq i \leq n\}.$$

The weight satisfies the triangular inequality and this allows us to define a distance of $\mathbb{F}_q^n$.

Definition 4. Let $\mathbf{x}$ and $\mathbf{y}$ be two elements of $\mathbb{F}_q^n$. Then the *Hamming distance* between $\mathbf{x}$ and $\mathbf{y}$ is given by

$$\mathbf{d}_H(\mathbf{x}, \mathbf{y}) = w(\mathbf{x} - \mathbf{y}).$$

Another important parameters for codes is the notion of *minimum distance*.

Definition 5. Let $\mathcal{C}$ be a code of length n over $\mathbb{F}_q$. Then the minimum distance d of $\mathcal{C}$ is

$$d = \min_{\mathbf{x} \neq \mathbf{y} \in C} \mathbf{d}_H(\mathbf{x}, \mathbf{y}).$$

If $\mathcal{C}$ is linear then this is simply given by $d = \min_{\mathbf{x} \in \mathcal{C} \setminus \{\mathbf{0}\}} w(\mathbf{x})$. And if k is the dimension of $\mathcal{C}$ as a linear code, we describe the code as $[n, k, d]$.

Theorem 1 (Singleton bound, [MS78])**.** *Let $\mathcal{C}$ be an $[n, k, d]$-linear code over a finite field $\mathbb{F}_q$ of size q. Then*

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

If $d = n - k + 1$, then $\mathcal{C}$ is called a maximum distance separable (MDS) code. MDS codes exist when the field is large enough as we see in the following example.

Example 1 (Reed-Solomon codes). Let $\mathbb{F}_q$ be a finite field of size q where its elements are denoted by $a_1, \cdots, a_q$. Let $n \leq q$. We define the following evaluation map

$$\begin{aligned} ev : \mathbb{F}_q[x] &\to \mathbb{F}_q^n \\ f(x) &\mapsto (f(a_1), \cdots, f(a_n)). \end{aligned}$$

If we denote the set of polynomials of degree at most $k-1$ over $\mathbb{F}_q$ by $\mathbb{F}_q[x]_{<k}$, then the image $\mathcal{C} = ev\left(\mathbb{F}_q[x]_{<k}\right)$ of $\mathbb{F}_q[x]_{<k}$ by the evaluation map above is an MDS code and it is called a Reed-Solomon code [MS78]. To see this, we use the fact that a polynomial of degree $k-1$ has $k-1$ roots at most. Therefore, $n-k+1$ elements of $(f(a_1), \cdots, f(a_n))$ are non-zero.

The nice thing about Reed-Solomon codes is that they have efficient decoding algorithms [MS78]. If we equip $\mathbb{F}_q^n$ with the bilinear form defined by the usual dot product, then we can define the orthogonal complement of $\mathcal{C}$ in $\mathbb{F}_q^n$. This vector space is called the dual code of $\mathcal{C}$ and we denote it by $\mathcal{C}^\perp$. The generator matrix of $\mathcal{C}^\perp$ is called the *parity check matrix* of $\mathcal{C}$. Parity check matrices are usually denoted by the letter $\mathbf{H}$ and they are of size $(n-k) \times n$.

1.3 Rank metric codes

This class of codes again consists of some vector space over finite field. But the notion "rank" in the name was introduce by Gabidulin when he used a metric different from the Hamming metric. With rank metric codes, we have to work with larger fields. Let q be a power of a prime and let m be a positive integer. Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ be the finite field extension of degree m. For rank metric codes we will need a new type of metric. Let n be a positive integer and let $\mathbb{F}_q^{m\times n}$ denote the set of $(m \times n)$-matrices over $\mathbb{F}_q$.

We know that $\mathbb{F}_{q^m}$ is a vector space over the field $\mathbb{F}_q$. So, if we fix a basis $(\alpha_1, \cdots, \alpha_m)$ of $\mathbb{F}_{q^m}/\mathbb{F}_q$, then we can define an isomorphism

$$\begin{aligned} V : \mathbb{F}_{q^m} &\to \mathbb{F}_q^m \\ (x_1\alpha_1 + \cdots + x_m\alpha_m) &\to (x_1, \cdots, x_m)^T \end{aligned}$$

This in turn defines an isomorphism M, with

$$\begin{aligned} M : \mathbb{F}_{q^m}^n &\to \mathbb{F}_q^{m\times n} \\ (a_1, \cdots, a_n) &\to (V(a_1), \cdots, V(a_n)). \end{aligned}$$

Definition 6. Let $\mathbf{x}$ and $\mathbf{y}$ be two elements of $\mathbb{F}_{q^m}^n$. We define the rank distance between $\mathbf{x}$ and $\mathbf{y}$ as

$$\mathbf{d}_R(\mathbf{x}, \mathbf{y}) = \mathbf{rank}(M(\mathbf{x}) - M(\mathbf{y})).$$

The rank weight of $\mathbf{x}$ is defined by $\mathbf{rank}\,\mathbf{x} = \mathbf{rank}\,M(\mathbf{x})$.

Having this new metric we can now define a rank metric code.

Definition 7. Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ be a finite field extension of degree m and let n be an integer. A rank metric code $\mathcal{C}$ of length n is a subset of $\mathbb{F}_{q^m}^n$ together with the metric defined in Definition 6. $\mathcal{C}$ is called a linear rank metric code if it is linear over $\mathbb{F}_{q^m}$. In this case, if k is its dimension as a vector space, then we write $[m \times n, k]$ to describe the code.

And we also have the notion of minimum distance.

Definition 8. Let $\mathcal{C}$ be rank metric code of length n over the extension $\mathbb{F}_{q^m}/\mathbb{F}_q$. Then the minimum distance d of $\mathcal{C}$ is

$$d = \min_{\mathbf{x} \neq \mathbf{y} \in C} \mathbf{d}_R(\mathbf{x}, \mathbf{y}).$$

If $\mathcal{C}$ is linear then this is simply given by $d = \min_{\mathbf{x} \in \mathcal{C} \setminus \{\mathbf{0}\}} \mathbf{rank}(M(\mathbf{x}))$. And if k is the dimension of $\mathcal{C}$ as a linear code, we describe the code as $[m \times n, k, d]$.

The version of the Singleton bound is given by the following theorem.

Theorem 2 (Singleton bound, [Gab85])**.** *Let $\mathcal{C}$ be an $[m \times n, k, d]$ rank metric code of length n over the extension $\mathbb{F}_{q^m}/\mathbb{F}_q$. Suppose that $n \leq m$. Then $d \leq n - k + 1$. In case $d = n - k + 1$, $\mathcal{C}$ is called a maximum rank distance MRD code.*

Remark 1. In Definitions 6, 7, 8 and Theorem 2, we used the extension $\mathbb{F}_{q^m}/\mathbb{F}_q$. It is possible to generalize all definitions and results stated for any finite field extension $\mathbf{L}/\mathbf{K}$, even when the fields are infinite. The proof of Theorem 2 is essentially the same as with the classical extension $\mathbb{F}_{q^m}/\mathbb{F}_q$. Notice also that we consider only linear codes.

Remark 2. In Theorem 2, we have used the condition $n \leq m$. This is the most interesting case for us in this thesis but a slightly different statement holds when $n > m$ [Loi08]. Also, the construction which we will see later can be modified to get an MRD code in this case by using the map M and transposition of matrices.

In the remaining part of this section, we will consider only $[m \times n, k, d]$-rank metric codes with $n \leq m$. The construction of an MRD code is similar to the construction of Reed-Solomon codes but it needs a special class of polynomials.

Definition 9 (Linearized polynomials)**.** Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ be a field extension of degree m. A q-linearized polynomial over $\mathbb{F}_{q^m}$ is a polynomial in $\mathbb{F}_{q^m}[x]$ of the form

$$f(x) = a_0 x + a_1 x^q + \cdots + a_l x^{q^l}.$$

The set of q-linearized polynomial over $\mathbb{F}_{q^m}$ is denoted by $L_{q,m}[x]$ and the set of q-linearized polynomial of degree q^{k-1} at most is denoted by $L_{q,m}[x]_{<k}$.

The q-linearized polynomials form an interesting ring where the multiplication is the composition of polynomials. Moreover any $f(x) \in L_{q,m}[x]$ is an $\mathbb{F}_q$-linear map

$$f : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}, \quad x \mapsto f(x).$$

Having all of these, we can now explain the construction of MRD code.

Example 2 (Gabidulin codes). Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ be a finite field extension of degree m where the basis is given by $\{a_1, \cdots, a_m\}$. Let $n \leq m$. We define the following evaluation map

$$\begin{aligned} qev : \mathbb{F}_{q^m}[x] &\to \mathbb{F}_{q^m}^n \\ f(x) &\mapsto (f(a_1), \cdots, f(a_n)) \end{aligned}$$

The image $\mathcal{C} = qev\left(L_{q,m}[x]_{<k}\right)$ of $L_{q,m}[x]_{<k}$ by the evaluation map above is an MRD code. It is called a Gabidulin code. They were independently discovered by Gabidulin [Gab85] and Delsarte [Del78]. To see that it is MRD, we use the fact that a polynomial of degree q^{k-1} has q^{k-1} roots at most. Since q-linearized polynomials are $\mathbb{F}_q$-linear maps on $\mathbb{F}_{q^m}$, we can use the rank nullity theorem to show that $\mathbf{rank}\left(M((f(a_1), \cdots, f(a_n))\right)$ is at least $n - k + 1$.

As with the case of Reed-Solomon codes, Gabidulin codes have efficient decoding algorithms [Gab85, Loi06]. Furthermore, the notion of dual codes and parity check matrix is exactly the same as with the Hamming metric.

1.4 Cryptosystem based on linear codes

Linear codes have applications in data transmission, data storage, network coding. Apart from those, linear codes have also found applications in Cryptography. The use of Hamming metric codes were first suggested by McEliece for use in cryptography [McE78]. The cryptosystem is based on the difficulty of solving the following problem.

Coset weights problem (CWP): *Let w be a positive integer. Let $\mathcal{C}$ be a random linear code over a finite field $\mathbb{F}$ together with the Hamming distance $\mathbf{d}_h$. Given a vector $\mathbf{x} \in \mathbb{F}^n$. Find the codeword $\mathbf{c} \in \mathcal{C}$ such that $\mathbf{d}_h(\mathbf{x}, \mathbf{c}) \leq w$.*

We first defined the notion of hardness of a problem.

Definition 10. A problem $\mathcal{P}$ is said to be in NP if given a solution to $\mathcal{P}$, we can check in polynomial time (with respect to the size of the input) that it is indeed a solution.

Here, we give an equivalent definition of NP. Usually, NP problems are defined to be the problems which can be solved in polynomial time using a non-deterministic Turing machine.

Definition 11. A problem $\mathcal{P}$ is called NP-hard if it is proven that any problem in NP can be reduced to $\mathcal{P}$ in polynomial time. That means that any problem in NP can be transformed into an instance of the problem $\mathcal{P}$ with some polynomial time operations.

Definition 12. If a problem is both NP and NP-hard, then it is called NP-complete.

NP-complete problems are considered to be difficult to solve, i.e. it is believed that they require exponential time algorithm. An example of an NP-complete problem is the coset weights problem. This was shown in [BMvT78] and we can interpret the result as solving the coset weights problem is in general a hard problem. A somehow equivalent problem is the problem of finding the minimum distance of binary linear code. This was also shown to be hard in [Var97].

This property is used to construct one way functions which are difficult to invert. However, there are some particular classes of linear codes for which there is a decoding algorithm. Such decoding algorithms are used as a trapdoor for the one way function, thus enabling the construction of public key cryptosystems.

Here is a general description of the cryptosystem using linear Hamming metric codes.

Let q be a power of a prime. Suppose that $\mathcal{C}$ is an $[n, k, d]$-linear code over $\mathbb{F}_q$ with a metric defined by $\mathbf{d}_H$. Let $\mathbf{G} \in \mathbb{F}^{k\times n}$ be a generator matrix of $\mathcal{C}$ and we also use it to denote the encoding map for $\mathcal{C}$. Let $t = \lceil \frac{d}{2} \rceil$. We suppose that $\mathcal{D}$ is an efficient decoding algorithm of $\mathcal{C}$ which can correct any error vectors with Hamming weight less than t. We define the following public key cryptosystem. We want to encrypt a message $\mathbf{m} \in \mathbb{F}_q^k$ into a ciphertext $c \in \mathbb{F}_q^n$.

(I) **Public key:** $\mathbf{G}, t$

(II) **Secret key:** $\mathcal{D}$

(III) **Encryption:** Choose a random vector $\mathbf{e} \in \mathbb{F}_q^n$ with Hamming weight less than t. Then $\mathbf{c} = \mathbf{mG} + \mathbf{e}$.

(IV) **Decryption:** Compute $\mathbf{m}' = \mathbf{mG} = \mathcal{D}(\mathbf{c})$ using the decoding algorithm. $\mathbf{m} = \mathbf{m}'\mathbf{G}^{-1}$. Here $\mathbf{G}^{-1}$ means the inverse of the injective mapping $\mathbf{m}' \mapsto \mathbf{m}'\mathbf{G}^{-1}$. An invertible $(k \times k)$-submatrix of $\mathbf{G}$ can be used for this.

In general, we want to choose the code $\mathcal{C}$ such that the generator matrix $\mathbf{G}$ should not show an apparent structure of the code. In this regard, it should not be possible for an attacker to find a decoding algorithm $\mathcal{D}$ given the generator matrix $\mathbf{G}$. To hide any visible structure on the generator matrix $\mathbf{G}$, it was originally suggested to use a permutation matrix $\mathbf{P} \in \mathbb{F}_q^{n\times n}$ and an invertible matrix $\mathbf{S} \in \mathbb{F}_q^{k\times k}$, which are kept secret. The public generator matrix is therefore $\mathbf{G}' = \mathbf{SGP}$. To decrypt $\mathbf{mG}' + \mathbf{e} = \mathbf{mSGP} + \mathbf{e}$, we notice that multiplying by the secret $\mathbf{P}^{-1}$ does not change the Hamming weight. Thus the decoding algorithm applied on $\mathbf{mSG}+\mathbf{eP}^{-1}$ should produce $\mathbf{mSG}$. Then we just compute $\mathbf{mS}$ by using the inverse map $\mathbf{G}^{-1}$ and then apply $\mathbf{S}^{-1}$ to recover $\mathbf{m}$.

The first cryptosystem based on linear codes was the McEliece cryptosystem and it uses classical Goppa codes [McE78]. The construction of Goppa codes is given as follows, it was discovered by V. D. Goppa in [Gop70].

We define a polynomial $g(x)$ of degree t over a finite field $\mathbb{F}_{2^m}$. Let $a_1, \cdots, a_n$ be distinct elements of $\mathbb{F}_{2^m}$ such that $g(a_i) \neq 0$ for $1 \leq i \leq n$. The binary Goppa codes are defined to be the vector space $\mathcal{C} \subset \mathbb{F}_2^n$ such that

$$\mathcal{C} = \left\{ (c_1, \cdots, c_n) : \sum_{i=1}^{n} \frac{c_i}{x - a_i} \equiv 0 \mod g(x) \right\}.$$

Goppa codes have dimension $n - mt$ at least and if we choose $g(x)$ not to have multiple roots, then we can show that these codes have minimum distance $2t + 1$. This class of codes have an efficient decoding algorithm. For more information of Goppa codes, one can have a look at [Ber73].

Although there are methods to distinguish high rate Goppa codes from random codes, general Goppa codes have many properties similar to random codes [FGUaO+11]. This gives the original McEliece cryptosystem the strength to withstand attacks. Namely, after forty years of extensive research, the cryptosystem is still structurally secure. The original parameters suggested by McEliece were proven to be insecure as it is shown in [BLP08]. One way to design secure parameters for cryptosystems based on linear codes is to check its security against a general algorithm for decoding any linear codes. A reference for that is the optimized version of the information set decoding algorithm presented in [BLP08]. In that work, they have suggested the following parameters for the McEliece cryptosystem using Goppa codes. For 80-bit security, i.e. the fastest attack against the cryptosystem needs approximately $\mathcal{O}(2^{80})$ operations, the public-key size is 520 Kbits. For 128-bit security, the public-key size is 1537 Kbits. This is the main drawback of the McEliece cryptosystem. It requires the use of large codes and therefore the public key sizes are large. This makes it impractical compared to RSA and Elliptic curves based cryptosystems. Notice that the article was published 10 years ago so that further improvement of that algorithm should make the key sizes worse.

Due to this, several methods were suggested to reduce the key sizes. Niederreiter suggested the use of Reed-Solomon codes instead of Goppa codes [Nie86]. Since these codes have a larger error correcting capability, they can be used to reduce the key sizes. In Niederreiter's proposal, he uses an equivalent version of the cryptosystem by using the parity check matrix of the code instead of the generator matrix. The cryptosystem is described as follows.

Let q be a power of a prime. Suppose that $\mathcal{C}$ is an $[n, k, d]$-linear code over $\mathbb{F}_q$ with a metric defined by $\mathbf{d}_H$. Let $\mathbf{H} \in \mathbb{F}^{(n-k)\times n}$ be a parity check matrix of $\mathcal{C}$. Let $\mathbf{P} \in \mathbb{F}_q^{n\times n}$ be a permutation matrix and let $\mathbf{S} \in \mathbb{F}_q^{(n-k)\times(n-k)}$ be an irreducible matrix. Let $t = \lceil \frac{d}{2} \rceil$. We suppose that $\mathcal{D}$ is an efficient decoding algorithm of $\mathcal{C}$ which can correct error vectors with weight less than t. In fact $\mathcal{D}$ is an algorithm such that given a syndrome $\mathbf{He}^T$, we can uniquely compute $\mathbf{e}$ provided that its weight is small enough. We define the following public key cryptosystem. Messages are represented by error vectors $\mathbf{e}$ of weight t at most and we encrypt them into a ciphertext $c \in \mathbb{F}_q^{n-k}$.

(I) **Public key:** $\mathbf{H}' = \mathbf{SHP}, t$

(II) **Secret key:** $\mathcal{D}, \mathbf{S}, \mathbf{P}$

(III) **Encryption:** The ciphertext is the syndrome $\mathbf{c} = \mathbf{H}'\mathbf{e}^T$.

(IV) **Decryption:** Compute $\mathbf{c}' = \mathbf{S}^{-1}\mathbf{c} = \mathbf{HPe}^T$. The decoding algorithm $\mathcal{D}$ on $\mathbf{c}'$ should help us to recover the error $\mathbf{Pe}^T$. Finally using $\mathbf{P}^{-1}$ gives us the message $\mathbf{e}$.

Remark 3. Notice that in the previous cryptosystem, we also incorporated the matrices $\mathbf{S}$ and $\mathbf{P}$ for hiding the structure of $\mathbf{H}$.

It was shown in [LDW94] that in terms of security, the McEliece version is equivalent to the Niederreiter version. However, the Niederreiter version allows the use of parity check matrices in systematic form and thus reduces the key sizes further. Also, the Niederreiter version can be used to produce a signature scheme [CFS01].

It was again proven that the use of Reed-Solomon codes was not enough. Namely, Sidelnikov and Shestakov showed that it is possible to recover a decoding algorithm by recovering the structure of generator matrix of the codes [SS92]. For decades, several proposal were made to overcome the problem of large key sizes. We classify these into three main parts.

Alternative constructions of codes

(i) Alternative codes: Reed-Solomon codes [Nie86], Low (Medium) density parity-check codes – L(M)DPC [MRS00, BC07], Srivastava codes [Per12],

algebraic-geometric codes [JM96] etc... Most of these alternatives were shown to be insecure [SS92, CMCP14, MRS00, BC07]. However, using Srivastava codes is considered to be safe. That is also due to its construction similar to Goppa codes but then, it also suffers the same problem with the key sizes.

(ii) Subcodes of Generalized Reed-Solomon codes: One example of attack is by using the Schur product [Wie10].

(iii) Codes with symmetry: quasi-cyclic, quasi-dyadic (L(M)DPC codes, Goppa codes, Srivastava codes) [BCGO09, Per12, BC07, MB09]: Using symmetry on the generator matrix of the codes, only a part of the generator matrix needs to be published and thus reducing the key sizes. There are attacks on some of these constructions [OTD10, GJS16].

Alternative methods for hiding the structure of the codes

The idea of replacing the permutation matrix $\mathbf{P}$ by a more general matrix allows to hide the structure of the original codes [BBC+16, BGLK+17]. These methods were mainly considered with the use of Reed-Solomon codes. However some attacks are possible, still by exploiting the structure of Reed-Solomon codes [CGGU+14].

As we can see above, many of the cryptosystems above are using Reed-Solomon codes. That is mainly because Reed-Solomon codes are MDS, and thus it is optimal in a sense that it allows us to use linear codes with smaller length and thus reducing the key sizes. However, we may also notice that cryptosystems using Reed-Solomon codes were frequently shown to be insecure. The main reason is that Reed-Solomon codes have too much structure which can be used to distinguish it from random linear codes. This is explained by the following.

Definition 13. For two vectors $\mathbf{x} = (x_1, \cdots, x_n)$ and $\mathbf{y} = (y_1, \cdots, y_n)$ in $\mathbb{F}_q^n$, the Schur product $\mathbf{x} * \mathbf{y}$ is the vector $(x_1 y_1, \cdots, x_n y_n)$. The Schur product of two linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ is the vector space defined by

$$\mathcal{C}_1 * \mathcal{C}_2 = \mathrm{Span}_{\mathbb{F}_q} \left(\{ \mathbf{x} * \mathbf{y} : \mathbf{x} \in \mathcal{C}_1, \mathbf{y} \in \mathcal{C}_2 \} \right).$$

For a linear code $\mathcal{C}$, $\mathcal{C}^2 := \mathcal{C} * \mathcal{C}$ is called the square code of $\mathcal{C}$.

The Schur product of Reed-Solomon codes have the following property.

Proposition 1. *Let $\mathcal{C}$ be a Reed Solomon code of length n and dimension $k \leq n/2$, then the square code of $\mathcal{C}$ is a Reed Solomon code of dimension $2k - 1$.*

This is a particular property for Reed-Somon codes. As it was presented in [CGGU+14], this is not in general true for random codes. Namely, for a random linear code of dimension k and length n, the dimension of the square code is expected to be $\min\left(n, \binom{k+1}{2}\right)$.

Use of rank metric codes

Apart from using different classes of linear Hamming metric codes, it was also suggested to use a different metric. Namely, the rank metric codes offers advantages against the classical Hamming metric to reduce the large key sizes. This cryptosystem is exactly similar to the McEliece cryptosystem with the Hamming metric, except that we use rank metric codes in $\mathbb{F}_{q^m}^n$ and we measure the weight of the error vectors by using the rank distance $\mathbf{d}_R$. The linear codes are $\mathbb{F}_{q^m}$-subspaces of $\mathbb{F}_{q^m}^n$. Furthermore, instead of using permutation matrices to hide the structure of the generator matrix we can use invertible matrices in $\mathbb{F}_q^{n\times n}$. The analogy to the Hamming metric here is that, if in the Hamming metric the permutation matrices leave the Hamming weight invariant, in the rank metric setting, the invertible matrices in $\mathbb{F}_q^{n\times n}$ leave the rank weight invariant.

A proposal was given by Gabidulin et al. in [GPT91]. This cryptosystem is using Gabidulin codes [Gab85], which we have defined in Example 2. Since Gabidulin codes are analogous to the Reed-Solomon codes, it is probably not surprising for the reader that cryptosystems based on Gabidulin codes are mostly insecure. In analogy to Reed-Solomon codes, there is also a distinguisher for Gabidulin codes. We will see such property in Chapter 2, Lemma 7. Another idea to see the weakness is that when multiplying a generator matrix of a Gabidulin code with an invertible matrix in $\mathbb{F}_q^{n\times n}$, we still have a generator matrix of a Gabidulin code but with different support vector. To avoid an attack similar to the Sidelnikov-Shestakov attack in [SS92], a distortion matrix is used: the public generator matrix is of the form $\mathbf{G}' = \mathbf{S}([\mathbf{G}|\mathbf{X}])\mathbf{P}$. The distortion matrix $\mathbf{X}$ is used so that the algebraic structure of the generator matrix $\mathbf{G}$ is removed.

We describe the cryptosystem using Gabidulin codes in the following.

Let q be a power of a prime. Suppose that $\mathcal{C}$ is an $[n, k, d]$ Gabidulin code defined over the extension field $\mathbb{F}_{q^m}/\mathbb{F}_q$. Let $\mathbf{G} \in \mathbb{F}^{k\times n}$ be a generator matrix of $\mathcal{C}$. We use a distortion matrix $\mathbf{X} \in \mathbb{F}_{q^m}^{k\times t_1}$ with the following property: the columns of $\mathbf{X}$ generates an $\mathbb{F}_q$-vector space of dimension t_1. Let $\mathbf{P} \in \mathbb{F}_q^{(n+t_1)\times(n+t_1)}$ be an invertible matrix and let $\mathbf{S} \in \mathbb{F}_{q^m}^{k\times k}$ be another invertible matrix. Let $t = \frac{d}{2}$. We suppose that $\mathcal{D}$ is an efficient decoding algorithm of $\mathcal{C}$ which can error vectors with weight less than t. A message $\mathbf{m}$ is an element of $\mathbb{F}_{q^m}^k$ and we encrypt them into a ciphertext $c \in \mathbb{F}_{q^m}^{n+t_1}$.

(I) **Public key:** $\mathbf{G}' = \mathbf{S}([\mathbf{G}|\mathbf{X}])\mathbf{P}, t - t_1$

(II) **Secret key:** $\mathcal{D}, \mathbf{S}, \mathbf{P}$

(III) **Encryption:** Choose a random vector $e \in \mathbb{F}_q^n$ with rank weight less than $t - t_1$. Then $c = \mathbf{m}\mathbf{G}' + e$.

(IV) **Decryption:** Compute $c' = c\mathbf{P}^{-1} = \mathbf{m}\mathbf{S}\mathbf{G} + e_1 + e_2$. Where e_1 is the error of rank weight t_1 produced by the distortion matrix $\mathbf{X}$ and e_2 is produced by e and it has rank weight $t - t_1$. With the decoding algorithm, $\mathcal{D}(c') = \mathbf{m}\mathbf{S}\mathbf{G}$. We can finally recover $\mathbf{m}$ by multiplying the last result with the inverse of $\mathbf{G}$ and $\mathbf{S}$.

The distortion matrix $\mathbf{X}$ has the effect that intentional errors are initially introduced in the original codes. This has the consequence that the new code can correct less errors than the original code. However this helps in hiding the structure of the original code. Unfortunately the use of distortion matrices is not enough as attacks were presented whenever counter-attacks were given. For an extensive reading on this, see [Gib95, GRS16, Ksh07, Gab08, RGH10, HTMR18].

Another proposal is the use of the class of low rank parity-check (LRPC) codes [GRSZ14]. This is the rank metric version of the LDPC codes.

Again as in the case of Reed-Solomon codes, Gabidulin codes have too much algebraic structure. This explain the attacks cited above on the cryptosystems using Gabidulin codes. That particular structure is exploiting the Frobenius morphism $\phi : x \mapsto x^q$ which was used to construct the Gabidulin codes. Recall that in Example 2 and Definition 9, the construction of the code involves the use of monomials of the form x^{q^i}.

Definition 14. For a vector $\mathbf{x} = (x_1, \cdots, x_n) \in \mathbb{F}_{q^m}^n$, define its image by the Frobenius morphism as $\phi(\mathbf{x}) = (\phi(x_1), \cdots, \phi(x_n))$. If $\mathcal{C}$ is a rank metric code over the extension $\mathbb{F}_{q^m}/\mathbb{F}_q$, then we define the linear code $\phi(\mathcal{C})$ as

$$\phi(\mathcal{C}) = \{\phi(\mathbf{x}) : \mathbf{x} \in \mathcal{C}\}.$$

It was shown in [HTM17] that if $\mathcal{C}$ is an MRD code of dimension k, then $\dim_{\mathbb{F}_{q^m}} \mathcal{C} \cap \phi(\mathcal{C}) = k - 1$ if and only if $\mathcal{C}$ is a Gabidulin code. Thus we have an easy method to recognize a Gabidulin code. This can also be used (like in [HTMR18]), to mount an attack against cryptosystems based on Gabidulin codes.

Due to these distinguisher attacks, it is important to find alternative MDS and MRD codes which still have performance comparable to the Reed-Solomon and Gabidulin codes.

Remark 4. In this chapter, all the statements were given with some particular cases of Reed-Solomon codes and Gabidulin codes. However, with little modification if needed, these statements hold when Generalized Reed-Solomon codes and Generalized Gabidulin codes are considered.

1.5 Summary of the main results of this dissertation

Now that we have seen how linear codes are used in cryptography, let us give a brief overview of what this thesis offers.

This first chapter was mainly about some results which we will need later. We have defined linear codes, both in the Hamming and rank metric setting. We have seen how linear codes are used in cryptography. We have presented the known constructions and we have explained why most of these constructions are insecure. This leads us to work on the following chapters.

Reed-Solomon and Gabidulin codes were proposed to be used in code based cryptosystems. Their main advantage is that they are optimal in the sense that given their parameters, they have the largest possible minimum distance. Consequently, they help in constructing a public key cryptosystems where the public key size is smaller than in the cryptosystem using classical Goppa codes. However their main drawback is that they have too much algebraic structures, which can be exploited to mount some attack on the cryptosystems. Therefore we want to find alternative linear codes and possibly, ones which are still maximum rank distance codes or maximum distance separable codes. A step into this direction is the main result of Chapter 2 of this thesis. In that chapter, we will emphasize more on the side of rank metric codes. There, we show that when we consider a finite field extension $\mathbb{F}_{q^m}/\mathbb{F}_q$, where the fields are large enough, then most of the linear rank metric codes are MRD codes. In other words, if one generates a random linear rank metric code, then with high probability, this code is an MRD code. Moreover, among all of these MRD codes, we prove that only a tiny part of them are Gabidulin codes. To prove these results, we give two different methods. The first method is a probabilistic method, where we use the Schwarz-Zippel lemma. The second method is more geometric. We show that the set of Gabidulin (resp. MRD) codes contains a non-empty Zariski-open set and therefore they are generic sets. Therefore, theoretically, there are many alternative codes which are potential candidates for use in code based cryptosystems. However, first we need to find a general construction for such family of codes. Secondly, for any proposed code, we must find a decoding algorithm. To confirm our results about the existence of MRD codes which are not Gabidulin

codes, we give a new construction of such codes. And our new construction also confirms the hypothesis that we need to work with large finite fields. To get such a construction, we had to work with linear rank metric codes over the polynomial ring $\mathbb{F}_p[x]$ and then reduce all codewords in this code modulo some specific irreducible polynomial $f(x)$. The newly constructed rank metric code will be over the finite field $\mathbb{F}_p[x]/(f(x))$.

In his paper [She16], Sheekey has given a construction of new MRD codes other than Gabidulin codes. They are called twisted Gabidulin codes. However, as we have mentioned before, for linear codes to be used in practice, we need a decoding algorithm. In Chapter 3, we will give two different decoding algorithms for these new class of MRD codes. For the first algorithm, we will use an algorithm similar to the algorithm by Kötter-Kschichang. Their algorithm was designed to decode Gabidulin codes. Here, we want to have a decoding algorithm for twisted Gabidulin codes. With some modification on the Kötter-Kschichang algorithm, we get a decoding algorithm for twisted Gabidulin codes albeit, it works only for some specific parameters. The second decoding algorithm is completely different from the first algorithm. It is using the Berlekamp-Massey algorithm in a different way. First, we develop an algorithm which can be used for Gabidulin codes. This algorithm is different from any existing algorithm. The main step consists in interpolating the received message in order to get its polynomial representation. By the property of Gabidulin codes, we can easily find some coefficients of the polynomial representing the error vector. By using some generalization of linear-feedback shift register, we have shown that the previously known coefficients can be used to recover the whole polynomial representing the error vector. This algorithm also runs in quadratic time so that it is as fast as existing algorithms. With some further modifications, we were able to describe a way to extend this algorithm to the family of twisted Gabidulin codes.

When looking at the Hamming metric equivalent of the second decoding algorithm in Chapter 3, we got the idea of expanding the notion of Hamming distance into a more general set by using the notion of linear complexity. Namely, the notion of Hamming metric is somehow equivalent to the notion of linear complexity of sequences with some fixed period. So, in this setting, the linear complexity of sequences defines a metric. In Chapter 4, we have shown this by using a theorem of König-Rados. Then we consider the case where the sequences are no longer required to have a fixed period. In this case, we have shown that linear complexity still defines a metric on sequences of finite length. We therefore define a new metric using the linear complexity of sequences of finite length and we will develop a coding theory using this new metric. We will give an exact formula on the number of finite sequences given a bounded linear complexity and from this we get the exact formula for the number of sequences with a fixed linear complexity. That chapter provides the basis of the theory which we need

when we construct a new cryptosystem.

As we have seen in this chapter, once we have a metric, we can construct a cryptosystem. In Chapter 5, we will propose a general construction of a new cryptosystem using the finite sequences and their linear complexity. We will then give a particular instance together with some parameters. We will indeed show that this new metric helps in producing a cryptosystem which needs smaller key size than in the original McEliece cryptosystem. As a basis for the security of the cryptosystem, we will show that the problem of finding the closest sequence with respect to the linear complexity metric is difficult in general. Indeed, we will show that such problem is NP-complete. Finally, we will close this work with a short conclusion in Chapter 6. There, we recall the main results obtained in this thesis and we will end with a short description of a future work on the new cryptosystem.

Chapter 2

Classes of Maximum rank distance codes

The four first sections of this Chapter are based on a work I did together with Alessandro Neri, Anna-Lena Horlemann-Trautmann and Joachim Rosenthal [NHTRR18]. The section 2.5 is an independent work from myself.

2.1 Introduction

In this chapter we will always work on rank metric codes over the extension field $\mathbb{F}_{q^m}/\mathbb{F}_q$ for a fixed positive integer m and a prime power q. An identity matrix will be denoted by $\mathbf{I}_n$ if it belongs to $\mathbb{F}_q^{n\times n}$.

For an $\mathbb{F}_{q^m}$-linear rank metric code, instead of writing $[m \times n, k, d]$, we just use the notation $[n, k, d]$, where the field $\mathbb{F}_{q^m}$ is understood. As we have seen in Chapter 1, an $[n, k, d]$-linear rank metric code satisfies the following Singleton bound: $d \leq n - k + 1$. If this bound is met, i.e. $d = n - k + 1$, then the code is called maximum rank distance (MRD) code. The first construction of MRD codes were independently presented by Gabidulin [Gab85] and Delsarte [Del78]. These are the Gabidulin codes which we presented in Chapter 1. Several construction of non-Gabidulin MRD codes were given in [CMP16, dlCKWW16, She16]. We show that there are many MRD codes. In this chapter, we will show that the properties of being MRD and non-Gabidulin are generic. This implies that over a field extension with large degree, randomly choosing a generator matrix for a rank metric code will produce, with high probability, an MRD codes. Furthermore, we will see that among all these MRD codes, Gabidulin codes are just a fraction of them.

2.2 Preliminaries

2.2.1 Properties of MRD codes

Definition 15. Let $(a_1, \cdots, a_n)$ be elements of $\mathbb{F}_{q^m}$, linearly independent over $\mathbb{F}_q$. Let $\gcd(m, s) = 1$ and let $[i] = q^i$. The $(k \times n)$ s-Moore matrix is defined by

$$\mathbf{M}_{s,k} = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ a_1^{[s]} & a_2^{[s]} & \cdots & a_n^{[s]} \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{[s(k-1)]} & a_2^{[s(k-1)]} & \cdots & a_n^{[s(k-1)]} \end{pmatrix}$$

Using Definition 15, a Gabidulin code is therefore a linear rank metric code with a generator matrix of the form $\mathbf{M}_{1,k}$. A more general class is given when $s \neq 1$ [KGo05].

Definition 16 (Generalized Gabidulin codes)**.** For a field extension $\mathbb{F}_{q^m}/\mathbb{F}_q$ and an integer n. A Generalized Gabidulin code is a linear rank metric code with generator matrix $\mathbf{M}_{s,k}$ with $\gcd(m, s) = 1$.

Generalized Gabidulin codes are still MRD.

Lemma 4. *Any linear MRD code over $\mathbb{F}_{q^m}/\mathbb{F}_q$ of length n and dimension k has a generator matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ in systematic form i.e. $\mathbf{G} = [\mathbf{I}_k|\mathbf{X}]$. Moreover all the entries of $\mathbf{X}$ are not in $\mathbb{F}_q$.*

Definition 17. Let $\mathcal{C}$ be a linear code of length n over $\mathbb{F}_{q^m}/\mathbb{F}_q$. We define the dual code $\mathcal{C}^\perp$ as

$$\mathcal{C}^\perp = \{\mathbf{x} \in \mathbb{F}_{q^m}^n : \ \mathbf{x}\mathbf{c}^T = \mathbf{0}, \forall \mathbf{c} \in \mathcal{C}\}.$$

A generator matrix of $\mathcal{C}^\perp$ is called a parity check matrix of $\mathcal{C}$.

The properties of linear rank metric codes and their dual are related by the following proposition.

Proposition 2 ([Gab85])**.** *Let $\mathcal{C}$ be an MRD (resp. a generalized Gabidulin code) of dimension k over $\mathbb{F}_{q^m}/\mathbb{F}_q$. The dual code $\mathcal{C}^\perp$ is an MRD (resp. a generalized Gabidulin code) code of dimension $n - k$.*

To classify linear codes, we need properties to characterize their classes. For that, we want to define a notion of equivalence of linear rank metric codes.

Given a linear code $\mathcal{C}$ over $\mathbb{F}_{q^m}/\mathbb{F}_q$, let $\mathbf{c} = (c_1, \cdots, c_n) \in \mathcal{C}$. We define

$$\mathbf{c}^{[i]} = \left(c_1^{[i]}, \cdots, c_n^{[i]}\right).$$

We also set $\mathcal{C}^{[i]} = \{\mathbf{c}^{[i]} : \ \mathbf{c} \in \mathcal{C}\}$.

Definition 18. An isometry on $\mathbb{F}_{q^m}^n$ with respect to the rank metric is a map $\mathbb{F}_{q^m}^n \to \mathbb{F}_{q^m}^n$ which preserves the rank.

A particular classes of isometries on $\mathbb{F}_{q^m}^n$ is described by the following lemma.

Lemma 5 ([Mor14], Proposition 2)**.** *The semilinear $\mathbb{F}_q$-rank isometries on $\mathbb{F}_{q^m}^n$ are of the form*

$$(\lambda, \mathbf{A}, \sigma) \in \left(\mathbb{F}_{q^m}^* \times \mathbf{GL}_n(q)\right) \rtimes \left(\mathbb{F}_{q^m}\right),$$

defined by

$$\begin{aligned}(\lambda, \mathbf{A}, \sigma) : \mathbb{F}_{q^m}^n &\to \mathbb{F}_{q^m}^n \\ (c_1, \cdots, c_n) &\mapsto (\sigma(\lambda c_1), \cdots, \sigma(\lambda c_n))\mathbf{A}.\end{aligned}$$

The nice property of these isometries is that if we apply these semilinear isometries on a generator matrix of a generalized Gabidulin code, which is a Moore matrix, then we still get a Moore matrix. In other word, a code is semilinearly isometric to a generalized Gabidulin code if and only if it is itself a generalized Gabidulin code.

Now, let us characterize MRD codes.

Proposition 3 (MRD criterion,[HTM17])**.** *Let $\mathbf{G} \in \mathbb{F}_{q^m}^{k\times n}$ be the generator matrix of a linear rank metric code over $\mathbb{F}_{q^m}/\mathbb{F}_q$. Then $\mathcal{C}$ is an MRD code if and only if*

$$\mathbf{rank}\left(\mathbf{A}\mathbf{G}^T\right) = k.$$

for all $\mathbf{A} \in \mathbb{F}_q^{k\times n}$ such that $\mathbf{rank}\,\mathbf{A} = k$.

And for the Gabidulin codes we have the following theorem.

Theorem 3 (generalized Gabidulin criterion,[HTM17])**.** *Let $\mathcal{C}$ be an MRD code over $\mathbb{F}_{q^m}/\mathbb{F}_q$ of dimension k. $\mathcal{C}$ is a generalized Gabidulin code if an only if there exists s with $\gcd(s, m) = 1$ such that*

$$\dim\left(\mathcal{C} \cap \mathcal{C}^{[s]}\right) = k - 1.$$

2.2.2 The Zariski topology over finite fields

Let $\mathbb{F}_q[x_1, \cdots, x_r]$ be a polynomial ring in r variables over $\mathbb{F}_q$. Let $\overline{\mathbb{F}}_q$ be the algebraic closure of $\mathbb{F}_q$. Let $S \subset \mathbb{F}_q[x_1, \cdots, x_r]$ be a finite set of polynomials, then the algebraic set defined by S is defined by

$$V(S) = \left\{\mathbf{x} \in \overline{\mathbb{F}}_q^{\,r} : \ f(\mathbf{x}) = 0, \forall f \in S\right\}.$$

These algebraic sets form the closed sets of a topology in $\overline{\mathbb{F}}_q^{\,r}$. This is called the Zariski topology. As usual the complements of the Zariski-closed sets are called the Zariski-open sets.

Definition 19. A subset $G \subset \overline{\mathbb{F}_q^r}$ is called a generic set if G contains a non-empty Zariski-open set.

If one has an algebraic set $V(S)$ as we have defined above, then the number of $\mathbb{F}_{q^m}$-rational points defined through

$$V(S, \mathbb{F}_{q^m}) := \{\mathbf{x} \in \mathbb{F}_{q^m}^r : \ f(\mathbf{x}) = 0, \forall f \in S\},$$

becomes in proportion to the cardinality of the whole vector space $\mathbb{F}_{q^m}^r$ smaller, as the extension degree m increases. This is a consequence of the Schwartz-Zippel Lemma.

Lemma 6 (Schwartz-Zippel,[Sch80]). *Let $f \in \mathbb{F}_q[x_1, x_2, \cdots, x_r]$ be a non-zero polynomial of total degree $d \geq 0$. Let $\mathbb{F}_{q^m}$ be an extension field and let $F \subset \mathbb{F}_{q^m}$ be a finite set. Let $v_1, v_2, \cdots, v_r$ be selected at random independently and uniformly from F. Then the probability that f vanishes on $(v_1, \cdots, v_r)$ satisfies*

$$\Pr(f(v_1, \cdots, v_r) = 0) \leq \frac{d}{\sharp F}.$$

2.3 Topological results

We know by Lemma 4 that any linear MRD code in $\mathbb{F}_{q^m}^n$ of dimension k is uniquely represented by its generator matrix $\mathbf{G} \in \mathbb{F}_{q^m}^{k \times n}$ in systematic form $\mathbf{G} = [\mathbf{I}_k | \mathbf{X}]$. Thus to an MRD code $\mathcal{C}$ corresponds a unique matrix $\mathbf{X}$. To move to the Zariski topology, we need to work on the algebraic closure. We will show that the set of matrices fulfilling the MRD criterion of Proposition 3, and the subset of these matrices not fulfilling the generalized Gabidulin criterion of Theorem 3, are generic sets over the algebraic closure.

Let us first work on the MRD criterion of Proposition 3.

Theorem 4. *Let $1 \leq k \leq n-1$. The set S_{MRD} defined by*

$$\left\{ \mathbf{X} \in \overline{\mathbb{F}}_{q^m}^{\,k \times (n-k)} : \ \forall \mathbf{A} \in \mathbb{F}_q^{n \times k} \text{ of rank } k \text{ and } \det([\mathbf{I}_k | \mathbf{X}]\mathbf{A}) \neq 0 \right\},$$

is a generic subset of $\overline{\mathbb{F}}_{q^m}^{\,k \times (n-k)}$.

Proof. First S_{MRD} is non-empty since there are Gabidulin codes for every set of parameters. S^C will denote the complement of a set S. If we denote the entries of $\mathbf{X} \in \overline{\mathbb{F}}_{q^m}^{\,k \times (n-k)}$ as the variables $x_1, \cdots, x_{k(n-k)}$, then for a given $\mathbf{A} \in \mathbb{F}_q^{n \times k}$, we have

$$\det([\mathbf{I}_k | \mathbf{X}]\mathbf{A}) \in \mathbb{F}_q[x_1, \cdots, x_{k(n-k)}].$$

Thus,

$$\begin{aligned} S_{MRD} &= \bigcap_{\substack{\mathbf{A}\in\mathbb{F}_q^{n\times k}\\ \mathbf{rank}\,\mathbf{A}=k}} \{\mathbf{X}\in {\overline{\mathbb{F}}_{q^m}}^{k\times(n-k)} : \det([\mathbf{I}_k|\mathbf{X}]\mathbf{A})\neq 0\} \\ &= \bigcap_{\substack{\mathbf{A}\in\mathbb{F}_q^{n\times k}\\ \mathbf{rank}\,\mathbf{A}=k}} V(\det([\mathbf{I}_k|\mathbf{X}]\mathbf{A}))^C, \end{aligned}$$

i.e. it is a finite intersection of Zariski-open sets. Therefore, S_{MRD} is a Zariski-open set. □

Moving to the generalized Gabidulin codes. We have the following Criterion.

Lemma 7. *Let $\mathcal{C}$ be a linear code of length n and dimension k over $\mathbb{F}_{q^m}/\mathbb{F}_q$ with generator matrix $[\mathbf{I}_k|\mathbf{X}]$ and let $0<s<m$ with $\gcd(s,m)=1$. $\mathcal{C}$ is a generalized Gabidulin code with parameter s if and only if $\mathbf{rank}(\mathbf{X}^{[s]}-\mathbf{X})=1$.*

Proof. We know that

$$\begin{aligned} &\dim(\mathcal{C}\cap\mathcal{C}^{[s]}) = k-1 \\ &\Leftrightarrow \mathbf{rank}\left[\begin{array}{c|c} \mathbf{I}_k & \mathbf{X} \\ \mathbf{I}_k & \mathbf{X}^{[s]} \end{array}\right] = k+1 \\ &\Leftrightarrow \mathbf{rank}\left[\begin{array}{c|c} \mathbf{I}_k & \mathbf{X} \\ \mathbf{0} & \mathbf{X}^{[s]}-\mathbf{X} \end{array}\right] = k+1 \\ &\Leftrightarrow \mathbf{rank}(\mathbf{X}^{[s]}-\mathbf{X}) = 1. \end{aligned}$$

We complete the result with Theorem 3. □

Now, we show that the set of generator matrices which does not satisfy the properties in Lemma 7 is a generic set.

Theorem 5. *Let $1\leq k\leq n-1$ and $0<s<m$ be integers with $\gcd(s,m)=1$. Suppose that*

$$S_{Gab,s} := \{\mathbf{X}\in {\overline{\mathbb{F}}_{q^m}}^{k\times(n-k)} : \mathbf{rank}(\mathbf{X}^{[s]}-\mathbf{X})=1\}\cap S_{MRD}.$$

The set $S_{Gab,s}$ is a Zariski-closed subset of the Zariski-open set S_{MRD}.

Proof. Let $\mathbf{X}\in S_{Gab,s}$. Since $\mathbf{X}\in S_{MRD}$, by Lemma 4, $\mathbf{X}_{ij}\notin\mathbb{F}_q$ for $i=1,\cdots,k$ and $j=1,\cdots,n-k$. Then $\mathbf{rank}(\mathbf{X}^{[s]}-\mathbf{X})=1$ is equivalent to $\mathbf{rank}(\mathbf{X}^{[s]}-\mathbf{X})<2$ which is the same as all (2×2)-minors of $\mathbf{X}^{[s]}-\mathbf{X}$ are zero.

Again, we denote the entries of $\mathbf{X}$ by the variables $x_1, \cdots, x_{n(n-k)}$. The (2×2)-minors are again polynomials in $\mathbb{F}_q[x_1, \cdots, x_{n(n-k)}]$ and suppose that they form the set S'. Then

$$\begin{aligned} S_{Gab,s} &= \{\mathbf{X} \in \overline{\mathbb{F}}_{q^m}^{k\times(n-k)} : \ f(x_1, \cdots, x_{k(n-k)}) = 0, \ \forall f \in S'\} \cap S_{MRD} \\ &= V(S') \cap S_{MRD}. \end{aligned}$$

Hence it is a Zariski-closed subset of $S_{MRD} \subset \overline{\mathbb{F}}_{q^m}^{k\times(n-k)}$. □

By Theorem 5, the complement of $S_{Gab,s}$ in S_{MRD}, i.e the set of MRD but not Gabidulin codes, is a Zariski-open subset of S_{MRD}. Thus, if it is non-empty, it is a generic set. In the following section, we will give conditions on the non-emptiness of the set.

Translating the results above, we can say that when we randomly generate $\mathbf{X}$ over the algebraic closure, with probability 1 we get an MRD code and it is not a Gabidulin code. This suggest that if the field $\mathbb{F}_{q^m}$ is large enough, then generating $\mathbf{X}$ randomly should produce an MRD code which is not a Gabidulin code.

2.4 Probability

In this section, we want to confirm the results from Section 2.3 by computing some probability when we generate random linear rank metric code.

2.4.1 Probability for MRD codes

When generating random linear rank metric code, we want to know a bound on the probability that the code is MRD.

Theorem 6. *Let $\mathbf{X} \in \mathbb{F}_{q^m}^{k\times(n-k)}$ be randomly chosen. Then*

$$\Pr\left([\mathbf{I}_k|\mathbf{X}] \textit{ generates an MRD code}\right) \geq 1 - \frac{k\prod_{i=0}^{k-1}(q^n - q^i)}{q^m} \geq 1 - kq^{kn-m}.$$

Proof. By Proposition 3, $[\mathbf{I}_k|\mathbf{X}]$ generates a non-MRD code if and only if

$$f := \prod_{\substack{\mathbf{A}\in\mathbb{F}_q^{n\times k} \\ \mathbf{rank}\,\mathbf{A}=k}} \det([\mathbf{I}_k|\mathbf{X}]\mathbf{A}) = 0.$$

We consider f as polynomial with variables as the entries of $\mathbf{X}$ which we denote by $x_1, \cdots, x_{k(n-k)}$. Then $\det([\mathbf{I}_k|\mathbf{X}]\mathbf{A})$ is a polynomial of degree k at most.

Moreover the number of matrices in $\mathbb{F}_q^{n\times k}$ with rank k is $\prod_{i=0}^{k-1}(q^n - q^i) \leq q^{kn}$. Using Schwartz-Zippel lemma from Lemma 6, we get

$$\Pr\left([\mathbf{I}_k|\mathbf{X}] \text{ does not generate an MRD code}\right) \leq \frac{\deg f}{q^m}.$$

Thus we get,

$$\Pr\left([\mathbf{I}_k|\mathbf{X}] \text{ generates an MRD code}\right) \geq 1 - \frac{k\prod_{i=0}^{k-1}(q^n - q^i)}{q^m} \geq 1 - kq^{kn-m}.$$

□

To improve the previous bound, we we will use the set $\tau(k,n)$ given by

$$\{\mathbf{E} \in \mathbb{F}_q^{k\times n} : \mathbf{E} \text{ is in reduced row echelon form and } \mathbf{rank}\,\mathbf{E} = k\}.$$

We can reformulate Proposition 3 in the following way.

Proposition 4. *Let $\mathbf{G} \in \mathbb{F}_{q^m}^{k\times n}$ be a generator matrix of a linear rank metric code. Then $\mathcal{C}$ is an MRD code if and only if $\mathbf{rank}(\mathbf{E}\mathbf{G}^T) = k$ for all $\mathbf{E} \in \tau(k,n)$.*

Proof. This uses the fact that reducing a matrix to row echelon form is just multiplying by invertible matrix which does not affect the rank of a matrix. □

Following Proposition 4, we can replace the matrix $\mathbf{A}$ in the proof of Theorem 6 by matrices $\mathbf{E} \in \tau(k,n)$. For $\mathbf{E} \in \tau(k,n)$, let $g_{\mathbf{E}}$ and h be the polynomials

$$g_{\mathbf{E}}(x_1, \cdots, x_{k(n-k)}) := \det([\mathbf{I}_k|\mathbf{X}]\mathbf{E}^T)$$

and

$$h(x_1, \cdots, x_{k(n-k)}) := \mathrm{lcm}\{g_{\mathbf{E}}(x_1, \cdots, x_{k(n-k)}) : \mathbf{E} \in \tau(k,n)\}.$$

Proposition 5. *The set of linear non-MRD codes of dimension k in $\mathbb{F}_{q^m}^n$ is in one-to-one correspondence with the algebraic set*

$$V(\{h\}) = \{(v_1, \cdots, v_{k(n-k)}) \in \mathbb{F}_{q^m}^{k(n-k)} : h(v_1, \cdots, v_{k(n-k)}) = 0\}.$$

Proof. By Proposition 4, the set of linear non-MRD codes of dimension k in $\mathbb{F}_{q^m}^n$ is in one-to-one correspondence with the algebraic set

$$\begin{aligned} V &= \bigcup_{\mathbf{E}\in\tau(k,n)} \left\{(v_1, \cdots, v_{k(n-k)}) \in \mathbb{F}_{q^m}^{k(n-k)} : g_{\mathbf{E}}(v_1, \cdots, v_{k(n-k)}) = 0\right\} \\ &= \left\{(v_1, \cdots, v_{k(n-k)}) \in \mathbb{F}_{q^m}^{k(n-k)} : \prod_{\mathbf{E}\in\tau(k,n)} g_{\mathbf{E}}(v_1, \cdots, v_{k(n-k)}) = 0\right\} \end{aligned}$$

$$= \left\{(v_1, \cdots, v_{k(n-k)}) \in \mathbb{F}_{q^m}^{k(n-k)} : h(v_1, \cdots, v_{k(n-k)}) = 0\}\right\},$$

where the two last inequalities follow from the property of algebraic set that

$$V(\{f\}) \cup V(\{g\}) = V(\{fg\}) = V(\{\mathrm{lcm}(f, g)\})$$

□

Note that in the definition of an algebraic set, it suffices to use the square-free part of the defining polynomial(s). In the above definition of V however, h is already square-free, as we show in the following.

Lemma 8. *For every $\mathbf{E} \in \tau(k, n)$ the polynomial $g_{\mathbf{E}}$ is square-free. In particular, the polynomial h in Proposition 5 is square-free.*

Proof. Since every variable x_1 appears only in a unique row of $[\mathbf{I}_k|\mathbf{X}]\mathbf{E}^T$, the degree with respect to every variable is at most 1. Therefore, $g_{\mathbf{E}}$ cannot have multiple factor. □

Let us now look at an upper bound on the degree of the polynomial h.

Lemma 9. *Let $\mathbf{E} \in \tau(k, n)$ and let U be the subspace of $\mathbb{F}_q^n$ defined by*

$$U = \{(u_1, \cdots, u_n) : u_{k+1} = \cdots = u_n = 0\}.$$

Then

$$\deg g_{\mathbf{E}} = k - \dim(rs(\mathbf{E}) \cap U),$$

where $rs(\mathbf{E})$ is the row space of the matrix $\mathbf{E}$.

Proof. Let $r := k - \dim(rs(\mathbf{E}) \cap U)$ with $0 \leq r \leq k$. We can write

$$\mathbf{E} := \left[\frac{\mathbf{E}_1}{\mathbf{E}_2}\right],$$

where $\mathbf{E}_1 \in \mathbb{F}_q^{k \times k}$ and $\mathbf{E}_2 \in \mathbb{F}_q^{(n-k)} \times k$. Since $\dim(rs(\mathbf{E}) \cap U) = k - r$, we have $\mathbf{rank}\, \mathbf{E}_2 = r$. Therefore, there exists a matrix $\mathbf{M} \in \mathbf{GL}_k(q)$ such that the first r columns of $\mathbf{E}_2\mathbf{M}$ are linearly independent and the last $k - r$ columns are zero. Then

$$g_{\mathbf{E}} = \det(\mathbf{I}_k|\mathbf{X}]\mathbf{E}^T) = \det \mathbf{M}^{-1} \det(\mathbf{E}_1\mathbf{M} + \mathbf{X}\mathbf{E}_2\mathbf{M}).$$

The last $k - r$ columns of the matrix $\mathbf{X}\mathbf{E}_2\mathbf{M}$ are zero, i.e. the last $k - r$ columns of $\mathbf{E}_1\mathbf{M} + \mathbf{X}\mathbf{E}_2\mathbf{M}$ do not contain any of the variables x_i. On the other

hand, the entries of the first r columns are polynomials in $\mathbb{F}_q[x_1, \cdots, x_{k(n-k)}]$ of degree 1, since

$$\mathbf{E}_1\mathbf{M} + \mathbf{X}\mathbf{E}_2\mathbf{M} = \left(\sum_{l=1}^{n} (\mathbf{E}_1)_{il}\mathbf{M}_{lj} + \sum_{l=1}^{k}\sum_{l'=1}^{n} \mathbf{X}_{il'}(\mathbf{E}_2)_{ll'}\mathbf{M}_{lj} \right)_{ij}.$$

Hence we have $\deg g_{\mathbf{E}} \leq r$.

Now consider the matrix $\mathbf{E}_2\mathbf{M}$. We can write

$$\mathbf{E}_2\mathbf{M} = [\overline{\mathbf{E}}_2|\mathbf{0}]$$

where $\overline{\mathbf{E}}_2$ is an $(n-k) \times r$ matrix of rank r. Hence

$$\mathbf{X}\mathbf{E}_2\mathbf{M} = [\mathbf{X}\overline{\mathbf{E}}_2|\mathbf{0}].$$

Fix i with $1 \leq i \leq k$ and denote by $(\mathbf{X}\overline{\mathbf{E}}_2)_i$ the i-th row of the matrix $\mathbf{X}\overline{\mathbf{E}}_2$. The polynomials $(\mathbf{X}\overline{\mathbf{E}}_2)_{ij}$, for $j = 1, \cdots, r$, only involve the variables $x_{(i-1)(n-k)+1}, \cdots, x_{i(n-k)}$. The Jacobian of these polynomials is $\overline{\mathbf{E}}_2^T$, whose rows are linearly independent over $\mathbb{F}_q$. Therefore the elements in every row are algebraically independent over $\mathbb{F}_q$ ([Lef12], Chap. I). Moreover, different rows involve different variables, hence we can conclude that the entries of the matrix $\mathbf{X}\overline{\mathbf{E}}_2$ are algebraically independent over $\mathbb{F}_q$.

Now, let us look at the set of all $r \times r$ minors of $\mathbf{X}\overline{\mathbf{E}}_2$. These minors are all different and hence linearly independent over $\mathbb{F}_q$, otherwise a non-trivial linear combination of them that give 0 would produce a non-trivial polynomial relation between the entries of $\mathbf{X}\overline{\mathbf{E}}_2$. The degree r term of $g_{\mathbf{E}}$ is a linear combination of these minors. If we write

$$\mathbf{E}_1\mathbf{M} = [*|\overline{\mathbf{E}}_1],$$

where $\overline{\mathbf{E}}_1 \in \mathbb{F}_q^{k\times(k-r)}$, then the coefficients of this linear combination are given by the $(k-r)\times(k-r)$ minors of $\overline{\mathbf{E}}_1$, multiplied by $\det \mathbf{M}^{-1}$. Since $\mathbf{E}^T\mathbf{M}$ has rank k and the last $k-r$ columns of $\mathbf{E}_2\mathbf{M}$ are 0, it follows that the columns of $\overline{\mathbf{E}}_1$ are linearly independent, and hence at least one of the coefficients of the linear combination is non-zero. This proves that the degree r term of $g_{\mathbf{E}}$ is non-zero, and hence $\deg g_{\mathbf{E}} = r$. □

Using the previous results, we get the following theorem.

Theorem 7. *Let $\mathbf{X} \in \mathbb{F}_{q^m}^{k\times(n-k)}$ be a random matrix. Then*

$$\Pr([\mathbf{I}_k|\mathbf{X}] \text{ generates an MRD code }) \geq 1 - \sum_{i=0}^{k} i \begin{bmatrix} k \\ k-i \end{bmatrix}_q \begin{bmatrix} n-k \\ i \end{bmatrix}_q q^{i^2} q^{-m}.$$

Proof. For every $i = 0, \cdots, k$, we define the set

$$\tau_i = \{\mathbf{E} \in \tau(k,n) : \dim(U \cap rs(\mathbf{E})) = k - i\},$$

with

$$U := \{(u_1, \cdots, u_n) \in \mathbb{F}_q^n : u_{k+1} = \cdots = u_n = 0\}.$$

By Lemma 2,

$$\sharp\tau_i = \begin{bmatrix} k \\ k-i \end{bmatrix}_q \begin{bmatrix} n-k \\ i \end{bmatrix}_q q^{i^2}.$$

Furthermore, Lemma 9 says that if $\mathbf{E} \in \tau_i$, then $\deg g_{\mathbf{E}} = i$. Hence, by the definition of $h(x_1, \cdots, x_{k(n-k)})$, we have

$$\deg h \leq \sum_{\mathbf{E}\in\tau(k,n)} \deg g_{\mathbf{E}} = \sum_{i=0}^{k} \sum_{\mathbf{E}\in\tau(k,n)} \deg g_{\mathbf{E}} = \sum_{i=0}^{k} i \begin{bmatrix} k \\ k-i \end{bmatrix}_q \begin{bmatrix} n-k \\ i \end{bmatrix}_q q^{i^2}.$$

Finally, we use the Schwartz-Zippel Lemma 6 to get the result. □

Remark 5. If we set $m < k(n-k) + \log_q k$, then the lower bound of Theorem 7 is negative. In this regard, the bound is not tight. However Theorem 7 is an improvement on the bound in Theorem 6.

2.4.2 Probability for Gabidulin codes

Using Lemma 7, the class of Gabidulin codes can be expressed as the union of the set $\mathcal{G}(s)$, for $1 \leq s \leq m-1$ with

$$\mathcal{G}(s) := \{\mathbf{X} \in (\mathbb{F}_{q^m} \backslash \mathbb{F}_q)^{k\times(n-k)} : \mathbf{rank}(\mathbf{X}^{[s]} - \mathbf{X}) = 1\}.$$

A simple upper bound on the probability that a random linear code is a Gabidulin code is given by the following lemma.

Lemma 10. *Let $\mathbf{X} \in \mathbb{F}_{q^m}^{k\times(n-k)}$ be a random matrix. Then*

$$\begin{aligned} \Pr([\mathbf{I}_k|\mathbf{X}] \textit{ generates a Gabidulin code}) &\leq \sum_{\substack{0<s<m \\ \gcd(s,m)=1}} \Pr(\mathbf{X} \in \mathcal{G}(s)) \\ &= \sum_{\substack{0<s<m \\ \gcd(s,m)=1}} \frac{\sharp\mathcal{G}(s)}{q^{mk(n-k)}}. \end{aligned}$$

We define the following map for any integer s with $1 \leq s \leq m-1$ with $\gcd(s,m) = 1$.

$$\Phi_s : \mathbb{F}_{q^m}^{k\times(n-k)} \to \mathbb{F}_{q^m}^{k\times(n-k)}$$

$$\mathbf{X} \mapsto \mathbf{X}^{[s]} - \mathbf{X}.$$

For simplicity let

$$\begin{aligned}\mathcal{R}_1 &:= \left\{\mathbf{A} \in \mathbb{F}_{q^m}^{k\times(n-k)} : \mathbf{rank}\,\mathbf{A} = 1\right\}\\ {\mathcal{R}_1}^* &:= \left\{\mathbf{A} \in (\mathbb{F}_{q^m}^*)^{k\times(n-k)} : \mathbf{rank}\,\mathbf{A} = 1\right\}\\ \mathcal{K} &:= (\operatorname{Ker}\mathbf{Tr})^{k\times(n-k)}\end{aligned}$$

Lemma 11.

(i) Given a matrix $\mathbf{A} \in \mathbb{F}_{q^m}^{k\times(n-k)}$, there exists a matrix $\mathbf{X} \in \mathbb{F}_{q^m}^{k\times(n-k)}$ such that $\Phi_s(\mathbf{X}) = \mathbf{A}$ if and only if $\mathbf{A} \in \mathcal{K}$.

(ii) If $\mathbf{A} \in \mathcal{R}_1$, then

$$\sharp\Phi_s^{-1}(\mathbf{A}) = \begin{cases} 0 & \text{if } \mathbf{A} \notin \mathcal{K} \\ q^{n(n-k)} & \text{if } \mathbf{A} \in \mathcal{K}. \end{cases}$$

(iii) For every integer s coprime to m

$$\mathcal{G}(s) = \phi_s^{-1}({\mathcal{R}_1}^* \cap \mathcal{K}),$$

+ and

$$\sharp\mathcal{G}(s) = \sharp\mathcal{G}(s) = q^{k(n-k)}\sharp({\mathcal{R}_1}^* \cap \mathcal{K}).$$

Proof.

(i) This is a simple consequence of Lemma 2.

(ii) If $\mathbf{A} \notin \mathcal{K}$, then the first part implies that $\Phi_s^{-1}(\mathbf{A}) = \emptyset$. Else, each entry $\mathbf{A}_{i,j}$ is an element of $\operatorname{Im}\phi_s$. The converse is also true. Since ϕ_s is $\mathbb{F}_q$-linear, $\sharp\phi_s^{-1}(\mathbf{A}_{i,j}) = \sharp\operatorname{Ker}\phi_s$. Now, using Lemma 2 again,

$$\sharp\operatorname{Ker}\phi_s = \frac{\sharp\mathbb{F}_{q^m}}{\sharp\operatorname{Im}\phi_s} = q.$$

and $\mathbf{A}$ has $k(n-k)$ entries, we get the result.

(iii) We have that

$$\Phi_s^{-1}(\mathcal{R}_1) = \left\{\mathbf{X} \in \mathbb{F}_{q^m}^{k\times(n-k)} : \mathbf{rank}(\mathbf{X}^{[s]} - \mathbf{X}) = 1\right\}$$

and

$$\Phi_s^{-1}\left((\mathbb{F}_{q^m}^*)^{k\times(n-k)}\right) = (\mathbb{F}_{q^m}\backslash\mathbb{F}_q)^{k\times(n-k)}.$$

Since ${\mathcal{R}_1}^* = \mathcal{R}_1 \cap (\mathbb{F}_{q^m}^*)^{k\times(n-k)}$, then

$$\Phi_s^{-1}({\mathcal{R}_1}^*) = \mathcal{G}(s).$$

We can partition

$${\mathcal{R}_1}^* = ({\mathcal{R}_1}^* \cap \mathcal{K}) \cup ({\mathcal{R}_1}^* \cap \mathcal{K}^C).$$

Using the first part, we get

$$\mathcal{G}(s) = \Phi_s^{-1}({\mathcal{R}_1}^*) = \Phi_s^{-1}({\mathcal{R}_1}^* \cap \mathcal{K}).$$

The second part completes the result.

□

We now give the following bound on the probability that a randomly generated code is a Gabidulin code.

Theorem 8. *Let $\mathbf{X} \in \mathbb{F}_{q^m}^{k\times(n-k)}$ be a random matrix. Denote by $\phi(m)$ the Euler ϕ-function. Then*

$$\Pr([\mathbf{I}_k|\mathbf{X}] + \textit{ generates a gen. Gabidulin code}) \leq \phi(m)(2q^{1-m})^{\lfloor\frac{k}{2}\rfloor\lfloor\frac{n-k}{2}\rfloor}.$$

Proof. From the third part of Lemma 11, we have that $\mathcal{G}(s) = \mathcal{G}(1)$ for the appropriate value of s. Now for any $\mathbf{X} \in (\mathbb{F}_{q^m}\backslash\mathbb{F}_q)^{k\times(n-k)}$, the rank of $\mathbf{X}^q - \mathbf{X}$ is larger than zero. Thus, we have

$$\mathcal{G}(1) = \left\{\mathbf{X} \in (\mathbb{F}_{q^m}\backslash\mathbb{F}_q)^{k\times(n-k)} : \mathbf{rank}(\mathbf{X}^q - \mathbf{X}) \leq 1\right\}.$$

Since "$\mathbf{rank}(\mathbf{X}^q - \mathbf{X}) \leq 1$" is equivalent to "any (2×2)-minor of $\mathbf{X}^q - \mathbf{X}$ is zero", then a necessary condition is that any set of non-intersecting minors is zero. We have $\lfloor\frac{k}{2}\rfloor\lfloor\frac{n-k}{2}\rfloor$ many such non-intersecting minors, each of which has degree at most $2q$ if we see the entries of $\mathbf{X}$ as the variables $x_1, \cdots, x_k(n-k)$. With Lemma 6, we get for each minor M_{ij},

$$\Pr(M_{i,j} = 0) \leq 2q^{1-m}.$$

Since the non-intersecting minors are independent events, the probability that all of these are zero is at most

$$(2q^{1-m})^{\lfloor\frac{k}{2}\rfloor\lfloor\frac{n-k}{2}\rfloor}.$$

The statement follows from Lemma 10. □

We also want to improve this bound as we did in the previous section.

Lemma 12. *Let $\mathbf{T}_a(x) = \mathbf{Tr}(ax)$ for $a \in \mathbb{F}_{q^m}$. The set $\mathcal{R}_1{}^* \cap \mathcal{K}$ is in one-to-one correspondence with the set*

$$\begin{aligned} V_R &:= \left\{(\alpha, \beta) \in \mathbb{F}_{q^m}^k \times \mathbb{F}_{q^m}^{n-k-1} : \alpha_i, \alpha_i\beta_i \in \operatorname{Ker} \mathbf{Tr} \backslash \{0\}\right\} \\ &= \left\{(\alpha, \beta) \in \mathbb{F}_{q^m}^k \times \mathbb{F}_{q^m}^{n-k-1} : \alpha_i \in \operatorname{Ker} \mathbf{Tr} \backslash \{0\}, \beta_i \in \bigcap_{i=1}^{k} \operatorname{Ker} \mathbf{T}_{\alpha_i} \backslash \{0\}\right\} \end{aligned}$$

via the map $\psi : V_R \to \mathcal{R}_1{}^* \cap \mathcal{K}$, given by

$$(\alpha, \beta) \mapsto \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix} \left(1, \beta_1, \ldots, \beta_{n-k-1}\right),$$

and hence

$$\sharp(\mathcal{R}_1{}^* \cap \mathcal{K}) \leq (q^{m-1} - 1)^{n-1}.$$

Proof. The map ψ is well defined and it is trivial to show that it is a bijection. Now we have $q^{m-1} - 1$ choices for each α_i. For the β_i we cannot have more than $\sharp \operatorname{Ker} \mathbf{T}_{\alpha_1} \backslash \{0\} = q^{m-1} - 1$ choices. Therefore, in total, there can be at most $(q^{m-1} - 1)^{n-1}$ elements in V_R. □

We now have the following theorem.

Theorem 9. *Let $\mathbf{X} \in \mathbb{F}_{q^m}^{k \times (n-k)}$ be randomly chosen. Then*

$$\Pr([\mathbf{I}_k | \mathbf{X}] \text{ generates a gen. Gabidulin code}) \leq \phi(m) q^{-(m-1)(n-k-1)(k-1)},$$

Proof. We use Lemmas 10, 11 part 3 and 12 to get the result. □

2.4.3 Existence of non-Gabidulin MRD codes

Let $Q(k)$ be the function on $\mathbb{N}$ given by

$$Q(k) = \prod_{i=1}^{k} \left(1 - \frac{1}{2^i}\right).$$

Lemma 13. *$(Q(k))_{k \in \mathbb{N}}$ is a decreasing positive sequence converging to $C \simeq 0.2887$. In particular $Q(k) > \frac{1}{4}$.*

Proof. The approximation of the limit is found in [S+03]. □

Lemma 14. *Let j, k be two positive integers with $0 < k \leq j$. Then*

$$\binom{j}{k}_q \leq \frac{1}{Q(k)} q^{k(j-k)}.$$

Proof.

$$\binom{j}{k}_q = \prod_{i=0}^{k-1} \frac{q^{j-i}-1}{q^{k-i}-1} \leq \prod_{i=0}^{k-1} \frac{q^{j-i}}{q^{k-i} - \left(\frac{q}{2}\right)^{k-i}} = \frac{1}{Q(k)} q^{k(j-k)}.$$

□

Theorem 10. *Let q be a prime power and let k and n be two integers such that $1 < k < n-1$. If $m \geq k(n-k) + \lceil \log_q(4k+1) \rceil$, then there exists a k-dimensional linear MRD code in $\mathbb{F}_{q^m}^n$ that is not a generalized Gabidulin code.*

Proof. We will prove the statement for $n \geq 2k$. The other cases are proved just by using the dual code. Let

$$f(m) = \sum_{i=0}^{k} i \binom{k}{k-i}_q \binom{n-k}{i}_q q^{i^2} q^{-m} + (m-1) q^{-(m-1)(n-k-1)(k-1)}.$$

We first need to show that $f(m) < 1$ for $m \geq k(n-k) + \lceil \log_q(4k+1) \rceil$. First suppose that $n \geq 5$. We have

$$\begin{aligned} \sum_{i=0}^{k} i \binom{k}{k-i}_q \binom{n-k}{i}_q q^{i^2} q^{-m} &\leq k q^{-m} \sum_{i=0}^{k} \binom{k}{i}_q \binom{n-k}{i}_q q^{i^2} \\ &= k \binom{n}{k}_q q^{-m} \\ &< 4k q^{k(n-k)-m}, \end{aligned} \tag{2.1}$$

where the last inequality follows from Lemmas 13 and 14. Furthermore,

$$(m-1) q^{-(m-1)(n-k-1)(k-1)} = q^{-(m-1)(n-k-1)(k-1) + \log_q(m-1)}. \tag{2.2}$$

Since

$$n + k^2 + 1 - kn \leq 0 \Leftrightarrow n \geq \frac{k^2+1}{k-1} \Leftrightarrow n \geq k + 1 + \frac{2}{k-1},$$

and this is satisfied when $k \leq n/2$ and $n \geq 5$, we have

$$\Big((1-m)(k(n-k) - n + 1) + \log_q(m-1)\Big) - (k(n-k) - m)$$

$$\begin{aligned} &= m(n+k^2-kn)+1+\log_q(m-1)-n \\ &\leq m(n+k^2-kn+1)-n \\ &\leq 0. \end{aligned}$$

Together with Equations (2.1) and (2.2), this gives us

$$f(m) < 4kq^{k(n-k)-m} + q^{-(m-1)(n-k-1)(k-1)+\log_q(m-1)} \leq (4k+1)q^{k(n-k)-m}.$$

So, we get that for $m \geq k(n-k) + \lceil \log_q(4k+1) \rceil$, it holds that $f(m) < 1$ which means there exists a non-Gabidulin MRD code.

For $n = 4$. This implies that $k = 2$ and $m \geq 4 + \lceil \log_1 9 \rceil$. For these fixed values of k and n, we consider $f(m) = g(q,m)$ as a function of q and m. Thus

$$g(q,m) = \frac{2q^4+q^3+2q^2+q}{q^m} + \frac{m-1}{q^{m-1}},$$

which is a decreasing function in both q and m. If we fix q, we have that $g\left(q, 4+\lceil \log_q 9 \rceil\right) \geq g(q,m)$, for every $m \geq 4 + \lceil \log_q 9 \rceil$. So we need to show that $g\left(q, 4+\lceil \log_q 9 \rceil\right) < 1$ for every prime power q. We have

$$\begin{aligned} g\left(q, 4+\lceil \log_1 9 \rceil\right) &= \frac{2q^4+q^3+2q^2+q}{q^{4+\lceil \log_q 9 \rceil}} + \frac{3+\lceil \log_q 9 \rceil}{q^{3+\lceil \log_q 9 \rceil}} \\ &\leq \frac{2q^4+q^3+2q^2+q}{9q^4} + \frac{3+\lceil 2\log_q 3 \rceil}{9q^3} =: K(q). \end{aligned}$$

We can check that $K(q)$ is a decreasing function in q. Thus,

$$g\left(q, 4+\lceil \log_q 9 \rceil\right) \leq K(q) \leq K(2) = \frac{4}{9} < 1.$$

So now, we have that $f(m) < 1$. From Theorem 7, we know that

$$\Pr([\mathbf{I}_k|\mathbf{X}] \text{ generates an MRD code }) \geq 1 - \sum_{i=0}^{k} i \begin{bmatrix} k \\ k-i \end{bmatrix}_q \begin{bmatrix} n-k \\ i \end{bmatrix}_q q^{i^2} q^{-m}.$$

Thus

$$\Pr([\mathbf{I}_k|\mathbf{X}] \text{ generates an MRD code }) \geq 1 - f(m) + (m-1)q^{-(m-1)(n-k-1)(k-1)}.$$

Using the fact that $f(m) < 1$, we have

$$\Pr([\mathbf{I}_k|\mathbf{X}] \text{ generates an MRD code }) > (m-1)q^{-(m-1)(n-k-1)(k-1)}.$$

Furthermore $\phi(m) \leq m-1$. So using Theorem 9, we get

$$\begin{aligned}&\Pr([\mathbf{I}_k|\mathbf{X}] \text{ generates an MRD code })\\ &>\\ &\Pr([\mathbf{I}_k|\mathbf{X}] \text{ generates a gen. Gabidulin code}).\end{aligned}$$

Therefore, we see that the proportion of MRD codes is larger than the proportion of Gabidulin codes. In other words, some of the MRD codes are non-Gabidulin codes. □

2.5 Construction of Non-Gabidulin MRD codes

In this section, we will give a construction of Non-Gabidulin MRD codes when we work in a large field. This further confirms the results of the previous sections. To do that we need a generalization of the concept of a linearized polynomial.

2.5.1 Differential operators

In this section, all notions are defined with arbitrary characteristic unless otherwise specified.

Definition 20 (Derivations)**.** Let R be a ring. A derivation of R is a map $\delta : R \longrightarrow R$ such that, for all x and y in R,

$$\delta(x+y) = \delta x + \delta y, \quad \delta(ab) = \delta(a)b + a\delta(b).$$

We usually write $x' = \delta x$ or $\delta(x)$ to avoid confusion.

For a non-negative integer n, the n-th successive derivation is denoted by δ^n and we also write $\delta^n x = x^{(n)}$. Notice that $\delta^0 = Id$.

Definition 21 (Differential ring)**.** A differential ring is a tuple (R, δ) of a ring R and a fixed derivation defined on it. If the derivation is understood, then we simply denote the differential ring by R. If R is a field, then we call it a differential field.

Definition 22. Let (R, δ) be a differential ring (resp. field). The set of elements x of R such that $\delta x = 0$ form a subring (resp. subfield) of R and it is called the ring (resp. field) of constants of (R, δ).

Definition 23 (Differential operators)**.** Let $(\mathbb{F}, \delta)$ be a differential field. A differential operator with coefficients in $\mathbb{F}$ is a polynomial in δ , i.e. of the form,

$$\mathcal{L} = \sum_{i=0}^{n} a_i \delta^i, \quad n \in \mathbb{N}, \quad a_i \in \mathbb{F}.$$

The set of differential operators form a non-commutative ring where the multiplication is defined by $\delta x = x' + x\delta$. We denote it by $\mathbb{F}\langle\delta\rangle$. In the above notation, n is called the order of $\mathcal{L}$ and we write $\operatorname{ord}\mathcal{L} = n$.

It is well known that the ring of differential operators admits an analogue of the Euclidean division and therefore we have the following proposition.

Proposition 6 ([MMo9]). *Let $(\mathbb{F}, \delta)$ be a differential field. The ring $\mathbb{F}\langle\delta\rangle$ admits a left and a right division. In other words, for $\mathcal{L}_1$ and $\mathcal{L}_2$ in $\mathbb{F}\langle\delta\rangle$, there are $\mathcal{Q}, \mathcal{R}, \mathcal{Q}', \mathcal{R}' \in \mathbb{F}\langle\delta\rangle$ such that,*

$$\mathcal{L}_2 = \mathcal{Q}\mathcal{L}_1 + \mathcal{R} \text{ and } \operatorname{ord}\mathcal{R} \leq \operatorname{ord}\mathcal{L}_1$$
$$\mathcal{L}_2 = \mathcal{L}_1\mathcal{Q}' + \mathcal{R}' \text{ and } \operatorname{ord}\mathcal{R}' \leq \operatorname{ord}\mathcal{L}_1$$

The operators $\mathcal{Q}, \mathcal{R}, \mathcal{Q}', \mathcal{R}' \in \mathbb{F}\langle\delta\rangle$ are unique.

A differential operator $\mathcal{L} \in \mathbb{F}\langle\delta\rangle$ defines a differential equation $\mathcal{L}y = 0$. If $y \in \mathbb{F}$ satisfy such equation, then we say that y is a solution of the differential equation. The set of solutions of such equations defines a subspace of $\mathbb{F}$ over the field of constants of $(\mathbb{F}, \delta)$.

Rational functions and Formal derivatives

From now on, everything will be in characteristic p, for a fixed prime p.

Let $\mathbb{F}_p$ be a finite field with p elements, p prime. The polynomial ring over $\mathbb{F}_p$ is denoted by $\mathbb{F}_p[T]$ and the field of rational functions over $\mathbb{F}_p$ is denoted by $\mathbb{F}_p(T)$. We can define a derivation on $\mathbb{F}_p[T]$ by the following,

$$\delta(a_0 + a_1T + \cdots + a_nT^n) = a_1 + 2a_2T + \cdots + na_nT^{n-1}.$$

This derivation can be extended to $\mathbb{F}_p(T)$ by the following formula.

Let $P, Q \in \mathbb{F}_p[T]$, then

$$\delta\left(\frac{P}{Q}\right) = \frac{P'Q - Q'P}{Q^2}.$$

With this derivation, we have that the ring of constants of $(\mathbb{F}_p[T], \delta)$ is equal to $\mathbb{F}_p[T^p]$ and the field of constants of $(\mathbb{F}_p(T), \delta)$ is equal to $\mathbb{F}_p(T^p)$.

It is well known that the degree of the extension $\mathbb{F}_p(T)/\mathbb{F}_p(T^p)$ is equal to p.

From now on, everything will be as in the above example. Let $K\langle\delta\rangle$, be the corresponding ring of differential operators, where $K = \mathbb{F}_p(T)$.

Let $\mathcal{L}y = 0$ be a differential equation defined over K i.e $\mathcal{L} \in K\langle\delta\rangle$. We have seen above that the set of solution of such equation is a subspace of $\mathbb{F}_p(T)$ as a vector space over $\mathbb{F}_p(T^p)$. One may ask what is the dimension of such solution space. This is given by the following theorem.

Proposition 7. *Let $(\mathbb{F}_p(T), \delta)$ be a differential field with field of constants $\mathbb{F}_p(T^p)$ and let $P_1, \cdots, P_n$ be n distinct elements of $\mathbb{F}_p(T)$. Then, $P_1, \cdots, P_m$ are linearly independent over $\mathbb{F}_p(T^p)$ if and only if the Wronskian W is invertible, with*

$$W_m = \begin{pmatrix} P_1 & P_2 & \dots & P_m \\ \delta P_1 & \delta P_2 & \dots & \delta P_m \\ \vdots & \vdots & \ddots & \vdots \\ \delta^{m-1} P_1 & \delta^{m-1} P_2 & \dots & \delta^{m-1} P_m \end{pmatrix}.$$

Proof. It is easy to see that if the P_i's are linearly dependent over $\mathbb{F}_p(T^p)$, then the matrix W has linearly dependent columns and thus it is singular. We will prove the converse by induction. The statement is obvious for $m = 1$. Assume that $P_1, \cdots, P_{m-1}$ are linearly independent, and that the corresponding Wronskian matrix W_{m-1} is non-singular. Let P_m be a rational function which is not in the vector space generated by $P_1, \cdots, P_{m-1}$ over $\mathbb{F}_p(T^p)$. If W_m is singular, then one of its columns is a linear combination of the other columns. W.l.o.g. (assuming that $\alpha_m = -1$), we have that

$$\delta^j(P_m) = \sum_{i=1}^{m-1} \alpha_i \delta^j(P_i), \quad 0 \leq j \leq m-1. \tag{2.3}$$

Applying δ to Equation (2.3), we get that for $0 \leq j \leq m-2$,

$$\delta^{j+1}(P_m) = \sum_{i=1}^{m-1} \delta(\alpha_i)\delta^j(P_i) + \alpha_i \delta^{j+1}(P_i)$$
$$\delta^{j+1}(P_m) = \sum_{i=1}^{m-1} \alpha_i \delta^{j+1}(P_i).$$

Using these two equations, we get

$$0 = \sum_{i=1}^{m-1} \delta(\alpha_i)\delta^j(P_i), \quad 0 \leq j \leq m-2.$$

Since W_{m-1} is non-singular, then $\delta(\alpha_i) = 0$ for $1 \leq i \leq m-1$. Thus $\alpha_i \in \mathbb{F}_p(T^p)$. But, with $j = 0$ in Equation (2.3), we would get that P_m is a linear combination of $P_1, \cdots, P_{m-1}$ over $\mathbb{F}_p(T^p)$. Thus we get a contradiction. Therefore W_m is also non-singular. □

Recall that from Remark 1, even if we are working with infinite field, we can still define the notion of maximum rank distance (MRD) code by the condition $k = n - d + 1$, where k is the dimension of the linear code, n its length and d its minimum distance.

The following theorem is an extension to infinite fields Proposition 3.

Theorem 11. *Let $\mathbf{L}$ and $\mathbf{K}$ be arbitrary fields (not necessarily finite) and suppose that $\mathbf{L}/\mathbf{K}$ is a field extension of finite degree. Let $\mathbf{G} \in \mathbf{L}^{k\times n}$ be the generator matrix of a $\mathbf{L}$-linear rank metric code $\mathcal{C}$ over the extension $\mathbf{L}/\mathbf{K}$. Then, $\mathcal{C}$ is a maximum rank distance code if and only if for any $(n \times k)$-matrix $\mathbf{M}$ over $\mathbf{K}$ of rank k, $\mathbf{GM}$ is invertible.*

Proof. Let $\mathcal{C}$ be a maximum rank distance $[n, k, n-k+1]$-linear code. Let $\mathbf{M}$ be an $(n \times k)$-matrix $\mathbf{M}$ over $\mathbf{K}$ of rank k. If $\mathbf{GM}$ is not invertible, then there is a vector $\mathbf{x} = (x_1, \cdots, x_k)$ such that $\mathbf{xGM} = \mathbf{0}$. We know that $\mathbf{c} = \mathbf{xG}$ is a codeword in $\mathcal{C}$ so that $\mathbf{rank}\, \mathbf{c} \geq n-k+1$.

Now, consider the $\mathbf{K}$-linear map

$$\begin{aligned} \phi_{\mathbf{c}} : \mathbf{K}^n &\to \mathbf{L} \\ (m_1, \cdots, m_k) &\mapsto \mathbf{c}(m_1, \cdots, m_k)^T \end{aligned}$$

It is easy to see that the rank of $\phi_{\mathbf{c}}$ is equal to $\mathbf{rank}\, \mathbf{c}$. But we have that $\mathbf{cM} = \mathbf{0}$ and $\mathbf{M}$ is of rank k, therefore the dimension of the kernel of $\phi_{\mathbf{c}}$ is at least k. By the rank nullity theorem, $k \leq n - \mathbf{rank}\, \mathbf{c}$. Thus $\mathbf{rank}\, \mathbf{c} \leq n-k$ which gives a contradiction. Therefore $\mathbf{GM}$ is invertible.

Conversely, suppose for any $(n \times k)$-matrix $\mathbf{M}$ over $\mathbf{K}$ of rank k, $\mathbf{GM}$ is invertible. We want to show that the code generated by $\mathbf{G}$ is a maximum rank distance code. If not, then we have a codeword $\mathbf{xG}$ with rank weight at most $n-k$. W.l.o.g we suppose that $\mathbf{xG} = (c_1, \cdots, c_{n-k}, \cdots, c_n)$ with c_i $\mathbf{K}$-linear combination of $(c_1, \cdots, c_{n-k})$ for $n-k < i \leq n$. Then we can find some $((n-k) \times k)$-matrix $\mathbf{A}$ such that

$$\mathbf{xG} \begin{bmatrix} \mathbf{A} \\ \mathbf{I}_k \end{bmatrix} = \mathbf{0}.$$

This is in contradiction with our hypothesis. □

Using Proposition 7 and 11, we can construct an MRD code over the extension $\mathbb{F}_p(T)/\mathbb{F}_p(T^p)$. For that we need the next theorem.

Theorem 12. *Let $(\mathbb{F}_p(T), \delta)$ be a differential field with field of constants $\mathbb{F}_p(T^p)$ and let $\{P_0, \cdots, P_{n-1}\}$ be a basis of the extension $\mathbb{F}_p(T)/\mathbb{F}_p(T^p)$, with $n = p-1$ and $P_i = T^i$. Then the matrix $\mathbf{G}$ is a generator matrix of an MRD code over $\mathbb{F}_p(T)/\mathbb{F}_p(T^p)$ with parameters $[n, k]$, where*

$$\mathbf{G} = \begin{pmatrix} P_0 & P_1 & \dots & P_{n-1} \\ \delta P_0 & \delta P_1 & \dots & \delta P_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \delta^{k-1} P_0 & \delta^{k-1} P_1 & \dots & \delta^{k-1} P_{n-1} \end{pmatrix}.$$

Proof. By using Theorem 11, it is enough to show that for any $(n \times k)$-matrix $\mathbf{M}$ over $\mathbb{F}_p(T^p)$ of rank k, $\mathbf{GM}$ is invertible over $\mathbb{F}_p(T)$. But we can easily check that for any $(n \times k)$-matrix $\mathbf{M}$ over $\mathbb{F}_p(T^p)$ of rank k, $\mathbf{GM}$ is an invertible Wronskian matrix by using Proposition 7. □

We have now given a construction of MRD codes over the extension of finite degree $\mathbb{F}_p(T)/\mathbb{F}_p(T^p)$. Of course this code is not really useful for application as we have infinite symbols. However, in the following, we will show that we can get an MRD code over finite fields from this code.

From Theorem 12, if we choose $P_i = T^i$, then the matrix $\mathbf{G}$ is

$$\mathbf{G} = \begin{pmatrix} 1 & T & T^2 & T^3 & \cdots & T^{k-1} & \cdots & T^{p-1} \\ 0 & 1 & 2T & 3T^2 & \cdots & (k-1)T^{k-2} & \cdots & (p-1)T^{p-2} \\ 0 & 0 & 2 & 6T & \cdots & (k-1)(k-2)T^{k-3} & \cdots & (p-1)(p-2)T^{p-3} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & (k-1)! & \ddots & \prod_{i=1}^{k-1}(p-i)T^{p-k} \end{pmatrix} \tag{2.4}$$

Since for any $(n \times k)$-matrix $\mathbf{M}$ over $\mathbb{F}_p(T^p)$ of rank k, $\mathbf{GM}$ is invertible over $\mathbb{F}_p(T)$. In particular for any $(n \times k)$-matrix $\mathbf{M}$ over $\mathbb{F}_p$ of rank k, $\mathbf{GM}$ is invertible over $\mathbb{F}_p(T)$. The entries of $\mathbf{GM}$ have the following property: the entries in the i-th row are polynomial of degree $p - i$.

Theorem 13. *Let $\mathbf{G}$ be the matrix defined in Equation (2.4). If Q is an irreducible polynomial $\mathbb{F}_p[T]$ such that $\deg Q = r = pk - \frac{k(k+1)}{2} + 1$. Then taking $\overline{\mathbf{G}} = \mathbf{G} \mod Q$ (meaning that all entries are taken modulo Q). The linear code over the extension $\mathbb{F}_{p^r}/\mathbb{F}_p$ with generator matrix $\overline{\mathbf{G}}$ is an MRD code of parameters $[p, k]$.*

Proof. We have seen that for any $(n \times k)$-matrix $\mathbf{M}$ over $\mathbb{F}_p$ of rank k, $\mathbf{GM}$ is invertible over $\mathbb{F}_p(T)$. Therefore the determinants of these matrices are non-zero polynomials in T of degree $pk - \frac{k(k+1)}{2}$ at most. By the choice of Q, these polynomials when reduced modulo Q gives non-zero elements of $\mathbb{F}_{p^r}$. Thus by applying Theorem 11 on $\overline{\mathbf{G}}$ over the field extension $\mathbb{F}_{p^r}/\mathbb{F}_p$, we get an MRD code. □

We have thus constructed a maximum rank distance code over the field extension $\mathbb{F}_{p^r}/\mathbb{F}_p$ with parameters $[p, k]$. For us to construct an MRD code of length p we need and extension of degree $r = pk - \frac{k(k+1)}{2} + 1$. This is too large for practical consideration. However, the point here is to show that we can construct MRD codes when the field extension is large. By using the Gabidulin criterion from Theorem 3, it is not hard to produce an example of our construction and show that it is not a Gabidulin code by using a computer.

Remark 6. In fact, we do not necessarily need to choose $r = pk - \frac{k(k+1)}{2} + 1$. We can just choose a smaller degree polynomial Q such that all the determinant of the maximal submatrices modulo Q are still non-zero. We can also restrict the choice of matrices in Theorem 11 to the matrices in reduce column echelon form.

Chapter 3

Decoding algorithms for rank metric codes

The second section of this Chapter is based on a work together with Joachim Rosenthal [RR17]. The third section is is based on my work [Ran17].

3.1 Twisted Gabidulin codes

Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ be a finite field extension of degree m. Let $n \leq m$ be a positive integer. All linear codes considered in this section are of length n.

As we have seen before, the key of the construction of Gabidulin codes is that we use linearized polynomials of degree q^{k-1} at most so that the polynomial can have q^{k-1} zeroes at most. However, we still can bound the number of roots of the polynomial even if we take polynomials of larger degree, namely q^k. For that we have the following proposition.

Proposition 8. *Let $f(x) = f_0x + f_1x^q + \cdots + f_kx^{q^k} \in L_{q,m}[x]$ be a linearized polynomial over $\mathbb{F}_{q^m}$. If $f(x)$ has a kernel of dimension k as an $\mathbb{F}_q$-linear map on $\mathbb{F}_{q^m}$, then there is a non-zero $z \in \mathbb{F}_{q^m}$ such that $f_0z = (-1)^k f_k^q z^q$.*

Proof. Suppose that the kernel V of f has dimension k. Let $x_0, \cdots, x_{k-1}$ be a basis of the kernel of f. We define the linearized polynomial $h(x) = \sum_i h_i x^{q^i}$ by

$$h(x) = \begin{vmatrix} x & x^q & \dots & x^{q^k} \\ x_0 & x_0^q & \dots & x_0^{q^k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k-1} & x_{k-1}^q & \dots & x_{k-1}^{q^k} \end{vmatrix}.$$

It is obvious that $h(x)$ also vanishes on V. Therefore, we have that $h_i = f_i z$ for some non-zero $z \in \mathbb{F}_{q^m}$. Furthermore, by computing the above determinant, we have $h_0 = (-1)^k h_k^q$. The result follows. □

Now we can construct the new class of MRD codes from [She16].

Definition 24. Let n, k, r, m be integers and suppose that $m > k, n > 0$. Let $\{a_1, \cdots, a_n\}$ be n distinct element of $\mathbb{F}_{q^m}$ linearly independent over $\mathbb{F}_q$. We define the set

$$\mathcal{G}(\eta, r) = \{f_0 x + f_1 x^q + \cdots + f_{k-1} x^{q^{k-1}} + \eta f_0^{q^r} x^{q^k} \ : \ f_i \in \mathbb{F}_{q^m}\}.$$

Let qev be the evaluation map defined in Example 2. If $\mathbf{N}(\eta) \neq (-1)^{kn}$, then $\mathcal{C}(\eta, r) = qev(\mathcal{G}(\eta, r))$ is an MRD code called a twisted Gabidulin code.

To see that this defines an MRD code, first we use the fact that the dimension of the kernel is at most k. But it cannot be k, by the choice of η and by using Theorem 8. And finally, as in the case of Gabidulin codes, we use the rank nullity theorem to complete the result.

Remark 7.

(i) The original Gabidulin codes corresponds to the case where $\eta = 0$.

(ii) When $r \neq 0$, then we have an MRD code which is linear only over $\mathbb{F}_q$.

(iii) This construction can be generalized by replacing the Frobenius map x^q by x^{q^s} with $\gcd(m, s) = 1$.

It was shown in [She16] that the class of twisted Gabidulin codes contains codes which are not equivalent to any generalized Gabidulin code.

In Chapter 1, we mentioned that the set of linearized polynomials $L_{q,m}[x]$ is a polynomial ring when we take the product between two linearized polynomials as their composition. Namely $(L_{q,m}[x], +, \circ)$ is a non commutative ring with $+$ being the usual addition of two polynomials and

$$ax^{q^i} \circ bx^{q^j} = ab^{q^i} x^{q^{i+j}}, \ i, j \in \mathbb{N} \text{ and } a, b \in \mathbb{F}_{q^m}.$$

The ring of linearized polynomials was studied by Ore in [Ore33]. It admits a left/right Euclidean division and this division will later be used in the decoding algorithm.

3.2 A Kötter-Kschischang-like decoding algorithm for twisted Gabidulin codes

Rank metric codes have application in network coding. In [KK08], a construction of subspace code, which is used in random network coding, was presented. It is similar to Gabidulin codes as the construction also involves evaluations of linearized polynomial. A decoding algorithm for the constructed subspace code was presented. That algorithm can be easily modified to decode Gabidulin codes. Here we want to modify the algorithm to apply it to the decoding of some particular cases of twisted Gabidulin codes.

Let $f \in \mathcal{G}(\eta, r)$ be the message polynomial. It is encoded into the codeword

$$(f(\alpha_1), f(\alpha_2), \cdots, f(\alpha_n)).$$

This codeword is sent and we assume that the word $(r_1, r_2, \cdots, r_n)$ was received. We define the error vector

$$\mathbf{e} = (f(\alpha_1) - r_1, f(\alpha_2) - r_2, \cdots, f(\alpha_n) - r_n)$$

Under the assumption $t < \frac{n-k+1}{2}$ rank errors happened, we seek the unique codeword $(f(\alpha_1), f(\alpha_2), \cdots, f(\alpha_n)), f \in \mathcal{G}(\eta, r)$ such that $\mathbf{e}$ has rank weight t.

$\mathbf{e} = (e_1, \cdots, e_n)$ now defines an $\mathbb{F}_q$-endomorphism

$$\begin{aligned} \mathbb{F}_q^n &\longrightarrow \mathbb{F}_{q^m} \\ (b_1, \cdots, b_n) &\longmapsto \sum_i b_i e_i. \end{aligned}$$

By the definition of $\mathbf{rank}$, we see that the rank of this map is $\mathbf{rank}(e)$. Therefore, by the rank nullity theorem, the kernel V of this map is a subspace of dimension $n - t$.

We have

$$V = \left\{ (b_1, \cdots, b_n) : \sum_i b_i f(\alpha_i) = \sum_i b_i r_i \right\}. \tag{3.1}$$

Assume that we have two linearized polynomials P_1 and P_2, with degree at most q^{n-t} and q^{n-t-k} respectively. Assume furthermore that P_1 and P_2 satisfy

$$P_1(\alpha_i) - P_2(r_i) = 0 \quad \forall i, 1 \leq i \leq n. \tag{3.2}$$

Choose $(b_1, \cdots, b_n) \in V$, we thus have that

$$P_1(\sum_i b_i \alpha_i) - P_2(\sum_i b_i r_i)$$

$$\begin{aligned}&=P_1(\sum_i b_i\alpha_i) - P_2(\sum_i b_i f(\alpha_i))\\ &=P_1(\sum_i b_i\alpha_i) - P_2(f(\sum_i b_i\alpha_i)).\end{aligned}$$

Therefore $P_1 - P_2 \circ f$ vanishes on a subspace $W \simeq V$ of dimension $n-t$.

First, assume that P_1 and P_2 are of the form

$$\begin{aligned}P_1(x) &= a_0x + a_1x^q + \cdots + a_{n-t-1}x^{q^{n-t-1}} + a_{n-t}x^{q^{n-t}}.\\ P_2(x) &= b_0x + b_1x^q + \cdots + b_{n-t-k}x^{q^{n-t-k}}.\end{aligned}$$

and of course

$$f(x) = f_0x + f_1x^q + \cdots + f_{k-1}x^{q^{k-1}} + \eta f_0^{q^r}x^{q^k}.$$

With these forms, we see that $P_1 - P_2 \circ f$ is of degree q^{n-t} at most where the two extreme monomials are

$$A = (a_0 - b_0f_0)x,$$

and

$$B = \left(a_{n-t} - b_{n-t-k}\eta^{q^{n-t-k}} f_0^{q^{r+n-t-k}}\right) x^{q^{n-t}}.$$

Now we are ready to explain the decoding algorithm. First, we want to solve the system of linear equations (3.2). This is a system of n independent equations at most. We have $2n - 2t - k + 2$ unknown. Since we consider that the error has rank, $t < \frac{n-k+1}{2}$, then we see that, the system (3.2) is underdetermined. Namely the solution space of this system of equation is of dimension 2 at least. This can be solved in polynomial time using the Gaussian elimination.

As we have seen, we now have two polynomials $P_1(x)$ and $P_2(x)$ such that $(P_1 - P_2 \circ f)$ is of q-degree $n-t$ but also has $n-t$ zeros. If one of $a_0 - b_0f_0$ and $a_{n-t} - b_{n-t-k}\eta^{q^{n-t-k}} f_0^{q^{r+n-t-k}}$ is equal to 0, then we must have $P_1 - P_2 \circ f = 0$ as polynomial. Therefore, we have two possibilities. In the first case we use a division algorithm to recover the polynomial f by computing P_1/P_2. Notice that the division is done in the linearized polynomial ring. If that does not work, then the second possibility is that as in the proof of Proposition 8, we must have a polynomial $h(x) = h_0x + \cdots + h_{n-t}x^{q^{n-t}}$ such that h and $P_1 - P_2 \circ f$ have the same zeroes with

$$h_0 = (-1)^{n-t}h_{n-t}^q,$$

and

$$h_{n-t} = (-1)^{n-t}\begin{vmatrix} x_0 & x_0^q & \cdots & x_0^{q^{n-t-1}}\\ x_1 & x_1^q & \cdots & x_1^{q^{n-t-1}}\\ \vdots & \vdots & \ddots & \vdots\\ x_{n-t-1} & x_{n-t-1}^q & \cdots & x_{n-t-1}^{q^{n-t-1}}\end{vmatrix}$$

Now the polynomial $P_1 - P_2 \circ f$ and h differ only by a constant multiple so that, for any solution $\{a_i, b_i\}$ of the system of equations (3.2)

$$\frac{h_0}{h_{n-t}} = \frac{a_0 - b_0 f_0}{a_{n-t} - b_{n-t-k}\eta^{q^{n-t-k}} f_0^{q^{r+n-t-k}}}. \tag{3.3}$$

So we will use the above relation to compute f_0 and we will thus get

$$(g(\alpha_1^q) - r_1, g(\alpha_2^q) - r_2, \cdots, g(\alpha_n^q) - r_n).$$

where

$$g(x) = f_1 x + \cdots + f_{k-1} x^{q^{k-2}}$$

Finally, a decoding algorithm of Gabidulin codes allows us to recover r_i and thus we can recover the original message.

So, what remains is how do we find f_0 from the relation (3.3). Suppose we have two linearly independent solutions $\{a_i, b_i\}$ and $\{a'_i, b'_i\}$ of the Equation (3.2). Then they form two polynomials with the same roots as $h(x)$. Thus

$$\begin{aligned} &\frac{a_0 - b_0 f_0}{a_{n-t} - b_{n-t-k}\eta^{q^{n-t-k}} f_0^{q^{r+n-t-k}}} \\ =&\frac{a'_0 - b'_0 f_0}{a'_{n-t} - b'_{n-t-k}\eta^{q^{n-t-k}} f_0^{q^{r+n-t-k}}}. \end{aligned}$$

In some case, we can solve this to recover f_0. For example, if $r + n = t + k \mod m$, then, since $x^{q^m} = x$, this becomes a polynomial equation of degree two. This can be easily solved and we try out the two values of f_0 for decoding. This condition is needed here as the above equation cannot be simply solved when the degree is large. Moreover, in that case there may be many possibilities for the solution f_0 of the equations and this will render the algorithm impracticable.

We summarize the decoding algorithm in the following. We suppose that $r + n = t + k \mod m$. And the message polynomial is

$$f(x) = f_0 x + f_1 x^q + \cdots + f_{k-1} x^{q^{k-1}} + \eta f_0^{q^r} x^{q^k}.$$

Algorithm

We are given a received codeword $(r_1, \cdots, r_n)$.

(I) Solve the system of linear equations in the variables b_j and a_j,

$$P_1(\alpha_i) - P_2(r_i) = 0, \quad \forall i, 1 \le i \le n$$

where

$$P_1(x) = a_0 x + \cdots + a_{n-t} x^{q^{n-t}},$$

and

$$P_2(x) = b_0 x + \cdots + b_{n-t-k} x^{q^{n-t-k}}.$$

Get two linearly independent solutions $\{a_i, b_i\}, \{a'_i, b'_i\}$.

(II) For the above solutions, compute P_1/P_2 as a polynomial division in the linearized polynomial ring. One can easily check if the quotient is the original message by comparing its rank distance to the received codeword.

(III) If the previous step does not give the original message, then solve for f_0 in the equation

$$\frac{a_0 - b_0 f_0}{a_{n-t} - b_{n-t-k} \eta^{q^{n-t-k}} f_0} = \frac{a'_0 - b'_0 f_0}{a'_{n-t} - b'_{n-t-k} \eta^{q^{n-t-k}} f_0}$$

This is equivalent to a polynomial equation of degree 2.

(IV) We now recover f_0, and remove the contribution of $f_0 x + \eta f_0^{q^r} x^{q^k}$ from the received word. Any decoding algorithm of Gabidulin codes can now be used to recover

$$g(x) = f_1 x + \cdots + f_{k-1} x^{q^{k-2}}$$

For the complexity, we give a brief explanation of why it is polynomial. Step I is solving a system of linear equations and thus it can be done in polynomial time. In fact finding two linearly independent solutions can be done in $\mathcal{O}(n^3)$ field operations. For Step II, the polynomial division in the linearized polynomial ring as it was described in [KK08] can also be done in polynomial time. This is $\mathcal{O}(n^2)$. For Step III, solving a polynomial system of degree 2 is easy. Finally, for Step IV, we just need to use any existing decoding algorithm of Gabidulin codes. There are plenty of those and they can be executed in polynomial time [Loi06, RP04]. To conclude, this decoding algorithm is dominated by the first step and thus the overal complexity if $\mathcal{O}(n^3)$ field operations. If one uses the the *Interpolate* procedure in [KK08], the overall complexity can even be reduced to $\mathcal{O}(n^2)$ field operations.

Remark 8. We have considered only the case where the code is the original Gabidulin code. The algorithm can be easily modified for any type of generalized Gabidulin code where we use $x^{q^{si}}$ instead of x^{q^i}.

3.3 A New decoding algorithm for MRD codes

Most of the decoding algorithms for Gabidulin codes are using syndrome computation, extended Euclidean algorithm, Berlekamp-Massey algorithm. In this section, we will use the Berlekamp-Massey algorithm for a rank metric code again, but in a different way. First, we give the new decoding algorithm for Gabidulin codes. Then we will show how to modify the algorithm to get it to work with general twisted Gabidulin code. In this section we will only work with rank metric code of full length i.e. a linear code in $(\mathbb{F}_{q^n})^n$. The rank of a linearized polynomial is the rank of the polynomial as an $\mathbb{F}_q$-linear map $\mathbb{F}_{q^n} \to \mathbb{F}_{q^n}$.

For reference, let us first show a brief description of some known decoding algorithm so that we can see the difference between them and our new algorithm. Suppose that $\mathbf{r}$ is the received vector.

(1) Compute the syndrome vector $\mathbf{r}\mathbf{H}^T$, where $\mathbf{H}$ is a parity check matrix of the code given by $h_{i,j} = h_j^{q^i}$. The entries s_i of this vector define a linearized polynomial $S(x)$.

(2) Determine two linearized polynomials $L(x)$ and $F(x)$ such that $F(x) = L(x) \circ S(x) \mod x^{q^{d-1}}$. Here, there are two methods: Use Berlekamp Massey [RP04] or use the extended Euclidean algorithm [Gab85].

(3) Find a basis $\{e_i\}$ of the kernel of $L(x)$.

(4) Compute the α_i's from $\sum_i e_i \alpha_i^{q^j} = s_j$.

(5) Find a matrix $\mathbf{Y}$, with $\alpha_j = \sum_i \mathbf{Y}_{j,i} h_i$.

(6) Finally the error vector is $\mathbf{e}Y$, where the entries of $\mathbf{e}$ are e_i.

(7) output the message as $\mathbf{r} - \mathbf{e}$.

3.3.1 Decoding algorithm for Gabidulin codes

Let us first look at some properties of linearized polynomials. Using Lemma 1, we get the following proposition.

Proposition 9. *Let λ be an element of $\mathbb{F}_{q^n}$ and let $\lambda\mathbb{F}_q$ be the $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^n}$ generated by λ. Then any linear map $\mathbb{F}_{q^n} \to \lambda\mathbb{F}_q$ has the form $\lambda\mathbf{Tr}(ax)$ for some $a \in \mathbb{F}_{q^n}$.*

The above representation in the proposition is of course not unique. As a consequence of the previous proposition, we have the following theorem. This theorem appears already in [LQ12] but here, we give a different proof.

Theorem 14. *Let f be a linearized polynomial of rank r, then there are two subsets of $\mathbb{F}_{q^n}$ $S_1 = \{a_1, \cdots, a_r\}$ and $S_2 = \{b_1, \cdots, b_r\}$ such that they are both linearly independent over $\mathbb{F}_q$ and that*

$$f(x) = a_1\mathbf{Tr}(b_1x) + \cdots a_r\mathbf{Tr}(b_rx).$$

Proof. Since f is of rank r, then we choose $\{a_1, \cdots, a_r\}$ to be a generator of the image of f as a linear map. By Proposition 9, each projection of f onto the subspace $a_i\mathbb{F}_q = \langle a_i \rangle$ has the form $a_i\mathbf{Tr}(b_ix)$. Thus we get the desired form of f. What remains to be shown is the linear independence of the b_i's. Without loss of generality, say $b_1 = \mathbf{m}_2b_2 + \cdots + \mathbf{m}_rb_r$, with $\mathbf{m}_i \in \mathbb{F}_q$. Then

$$\begin{aligned} f(x) &= a_1\left((\mathbf{Tr}((\mathbf{m}_2b_2 + \cdots + \mathbf{m}_rb_r)x))\right) + a_2\mathbf{Tr}(b_2x) + \cdots + a_r\mathbf{Tr}(b_rx) \\ &= a_1\mathbf{Tr}(\mathbf{m}_2b_2x) + \cdots + a_1\mathbf{Tr}(\mathbf{m}_rb_rx) + a_2\mathbf{Tr}(b_2x) + \cdots + a_r\mathbf{Tr}(b_rx) \\ &= (a_2 + a_1\mathbf{m}_2)\mathbf{Tr}(b_2x) + \cdots + (a_r + a_1\mathbf{m}_r)\mathbf{Tr}(b_rx). \end{aligned}$$

Thus rank of f is at most $r-1$ which is a contradiction. □

From Theorem 14, we get the following corollary.

Corollary 1. *Let $f(x)$ be a linearized polynomial of rank r over the field extension $\mathbb{F}_{q^n}/\mathbb{F}_q$ such that*

$$f(x) = f_0x + f_1x^q + \cdots + f_{n-1}x^{q^{n-1}}.$$

Then there are two subsets of $\mathbb{F}_{q^n}$ $S_1 = \{a_1, \cdots, a_r\}$ and $S_2 = \{b_1, \cdots, b_r\}$ such that they are both linearly independent over $\mathbb{F}_q$, and for all integer i such that $0 \leq i \leq n-1$,

$$f_i = \sum_{j=1}^{r} b_j^{q^i} a_j.$$

Definition 25. Let $f(x) = f_0x + f_1x^q + \cdots + f_{n-1}x^{q^{n-1}}$ be a linearized polynomial. The Dickson matrix associated to $f(x)$ is the matrix

$$\mathbf{M} = \begin{pmatrix} f_0 & f_{n-1}^q & \cdots & f_1^{q^{n-1}} \\ f_1 & f_0^q & \cdots & f_2^{q^{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n-1} & f_{n-2}^q & \cdots & f_0^{q^{n-1}} \end{pmatrix}.$$

Recall that given $\{a_1, \cdots, a_k\} \subset \mathbb{F}_{q^n}$, the Moore matrix associated to the a_i's is the matrix

$$\begin{pmatrix} a_1 & a_2 & \ldots & a_k \\ a_1^q & a_2^q & \ldots & a_k^q \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{q^{k-1}} & a_2^{q^{k-1}} & \ldots & a_k^{q^{k-1}} \end{pmatrix}$$

It is well known that the above Moore matrix is invertible if and only if the a_i's are linearly independent over $\mathbb{F}_q$.

As a consequence of Corollary 1, we have the following theorem.

Theorem 15. *Let $f(x)$ be a linearized polynomial of rank r over the field extension $\mathbb{F}_{q^n}/\mathbb{F}_q$ such that*

$$f(x) = f_0 x + f_1 x^q + \cdots + f_{n-1} x^{q^{n-1}}.$$

Let $\mathbf{M}_1, \cdots, \mathbf{M}_n$ be the rows of the matrix $\mathbf{M}$ as in Definition 25. Then we have the following property:

(i) The matrix $\mathbf{M}$ is of rank r.

(ii) Any r successive rows $\mathbf{M}_i, \cdots, \mathbf{M}_{i+r}$ are linearly independent and the other rows are linear combinations of them.

(iii) All $(r \times r)$-matrices $(M_{i,j})_{(i \mod n, j \mod n)}$; $l_1 \leq i \leq l_1 + r$, $l_2 \leq j \leq l_2 + r$ with $0 \leq l_i \leq n-1$ are invertible.

Proof. From Corollary 1, one sees that $\mathbf{M} = \mathbf{BA}$, where

$$\mathbf{B} = \begin{pmatrix} b_1 & b_2 & \dots & b_r \\ b_1^q & b_2^q & \dots & b_r^q \\ \vdots & \vdots & \ddots & \vdots \\ b_1^{q^{n-1}} & b_2^{q^{n-1}} & \dots & b_r^{q^{n-1}} \end{pmatrix} \text{ and } \mathbf{A} = \begin{pmatrix} a_1 & a_1^q & \dots & a_1^{q^{n-1}} \\ a_2 & a_2^q & \dots & a_2^{q^{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ a_r & a_r^q & \dots & a_r^{q^{n-1}} \end{pmatrix}$$

Since the a_i's are linearly independent over $\mathbb{F}_q$ and the same for the $b_i's$, we see that each r successive rows of B and any r successive columns of A constitute invertible matrices. All statements of the theorem follow from these facts. □

This theorem is important to us. It enables us to build a new decoding algorithm.

We are now ready to explain the decoding algorithm. It consists of two steps. The first part is to interpolate the received message to construct the polynomial $f(x) + g(x)$, where $f(x)$ is the message polynomial and $g(x)$ is the error polynomial. Since $f(x)$ is of degree q^{k-1} at most, we should know the coefficient of x^{q^i} in $g(x)$, $\forall i \geq k$. We will show that these coefficients are enough to recover the whole polynomial $g(x)$ with some condition on the rank of $g(x)$.

3.3.1.1 Polynomial interpolation

First of all, we need to do some interpolation to get a linearized polynomial form. Recall that the encoding was just the evaluation

$$\begin{aligned} qev : L_{q,n}[x] &\to (\mathbb{F}_{q^n})^n \\ f(x) &\mapsto (c_1, c_2, \cdots, c_n), \end{aligned}$$

where $c_i = f(a_i)$ and $\{a_1, a_2, \cdots, a_n\}$ is a fixed basis of $\mathbb{F}_{q^n}/\mathbb{F}_q$.

We assume that an error $\mathbf{e} = (e_1, e_2, \cdots, e_n)$ was added to the original codeword and suppose that $\mathbf{rank}(\mathbf{e}) = t < \frac{n-k+1}{2}$. Therefore $(r_1, r_2, \cdots, r_n)$ was received with $r_i = c_i + e_i$.

Let $\mathbf{U}$ be the Moore matrix

$$\mathbf{U} = \begin{pmatrix} a_1 & a_1^q & \dots & a_1^{q^{n-1}} \\ a_2 & a_2^q & \dots & a_2^{q^{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ a_n & a_n^q & \dots & a_n^{q^{n-1}} \end{pmatrix}$$

Then

$$\mathbf{U} \begin{pmatrix} f_0 + g_0 \\ \vdots \\ f_{n-1} + g_{n-1} \end{pmatrix} = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$

where $g(x) = \sum g_i x^{q^i}$ is the error polynomial corresponding to $\mathbf{e}$ i.e. $g(a_i) = e_i$. Obviously, $g(x)$ as an $\mathbb{F}_q$-linear map has rank $t < \frac{n-k+1}{2}$.

Thus, we may compute $\mathbf{U}^{-1}$ in advance and then compute

$$\mathbf{U}^{-1} \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}.$$

This gives us $f_0 + g_0, \cdots, f_{n-1} + g_{n-1}$. Since $f_i = 0, \forall i \geq k$, we now know the values of $g_k, \cdots g_{n-1}$. In the next step, we will use these coefficients to recover the other coefficients of $g(x)$.

3.3.1.2 Polynomial reconstruction

Let us have a look at the matrix $\mathbf{M}$ in Theorem 15 corresponding to the error polynomial $g(x)$. We consider its submatrix

$$\mathbf{W} = \begin{pmatrix} g_0 & g_{n-1}^q & \cdots & g_{k+t-1}^{q^{n-(k+t-1)}} & \cdots & g_k^{q^{n-k}} & \cdots & g_1^{q^{n-1}} \\ g_1 & g_0^q & \cdots & g_{k+t}^{q^{n-(k+t-1)}} & \cdots & g_{k+1}^{q^{n-k}} & \cdots & g_2^{q^{n-1}} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ g_{t-1} & g_{t-2}^q & \cdots & g_{k+2t-2}^{q^{n-(k+t-1)}} & \cdots & g_{k+t-1}^{q^{n-k}} & \cdots & g_t^{q^{n-1}} \\ g_t & g_{t-1}^q & \cdots & g_{k+2t-1}^{q^{n-(k+t-1)}} & \cdots & g_{k+t}^{q^{n-k}} & \cdots & g_{t+1}^{q^{n-1}} \end{pmatrix}.$$

We know that the t last rows are linearly independent and that the first row should be a linear combination of the t last rows.

This gives us an equation of the form

$$(\lambda_0, \cdots, \lambda_t) \begin{pmatrix} g_{k+t-1}^{q^{n-(k+t-1)}} & \cdots & g_k^{q^{n-k}} \\ g_{k+t}^{q^{n-(k+t-1)}} & \cdots & g_{k+1}^{q^{n-k}} \\ \vdots & \ddots & \vdots \\ g_{k+2t-2}^{q^{n-(k+t-1)}} & \cdots & g_{k+t-1}^{q^{n-k}} \\ g_{k+2t-1}^{q^{n-(k+t-1)}} & \cdots & g_{k+t}^{q^{n-k}} \end{pmatrix} = \mathbf{0} \tag{3.4}$$

where we may assume that $\lambda_0 = 1$.

After interpolation we know the coefficients $g_k, \cdots, g_{n-1}$. Since we suppose that $t < \frac{n-k+1}{2}$, then $k + 2t - 1 \leq n - 1$. Thus we know all the coefficients $g_k, \cdots, g_{k+2t-1}$. And thus, by Theorem 15, this equation has unique solution in $\boldsymbol{\lambda}$ which we can compute. This can be done for example by using matrix inversion but that will take $\mathcal{O}(t^3)$ operations.

Similarly to the case of Reed-Solomon codes, we can do better. Namely, we have here a Toeplitz-like matrix. And this can be solved by using a Berlekamp-Massey-like algorithm from [RP04]. To see this let $u_i = g_i^{q^{n-i}}$. Therefore, Equation (3.4) becomes.The algorithm was first discovered in [PT91] and then rediscovered in [RP04] without proof. For a proof, one can have a look at [VGM11].

$$(\lambda_0, \cdots, \lambda_t) \begin{pmatrix} u_{k+t-1} & \cdots & u_k \\ u_{k+t}^q & \cdots & u_{k+1}^q \\ \vdots & \ddots & \vdots \\ u_{k+2t-2}^{q^{t-1}} & \cdots & u_{k+t-1}^{q^{t-1}} \\ u_{k+2t-1}^{q^t} & \cdots & u_{k+t}^{q^t} \end{pmatrix} = \mathbf{0} \tag{3.5}$$

We want to solve Equation (3.5) i.e. we want to find $\lambda_1, \cdots, \lambda_t$ from the sequence $(u_{k+2t-1}, \cdots, u_{k+t}, \cdots, u_k)$. Equation (3.5) is exactly the form of recurrence shown in [RP04]. In that paper, they gave an algorithm for solving Equation (3.5). We will give the algorithm in Algorithm 1. We set $d = n-k+1$.

Algorithm 1 Berlekamp-Massey

procedure BERLEKAMP-MASSEY($s_0, \cdots, s_{2t-1}$)
  $L \leftarrow 0$
  $\Lambda^{(0)}(x) \leftarrow x$
  $B^{(0)}(x) \leftarrow x$
  $i \leftarrow 0$
  while $i \leq d-2$ **do**
    $\Delta_i \leftarrow s_i + \sum_{j=1}^{L} \lambda_j^{(i)} s_{i-j}^{q^j}$
    $\Lambda^{(i+1)} \leftarrow \Lambda^{(i)} - \Delta_i x^q \circ B^{(i)}(x)$
    if $\Delta_i == 0$ **then**
      $B^{(i+1)}(x) \leftarrow x^q \circ B^{(i)}(x)$
    else
      if $2L > i$ **then**
        $B^{(i+1)}(x) \leftarrow x^q \circ B^{(i)}(x)$
      else
        $B^{(i+1)}(x) \leftarrow \Delta_i^{-1} \Lambda^{(i)(x)}$
        $L \leftarrow i + 1 - L$
      end if
    end if
    $i \leftarrow i + 1$
  end while
  return $\Lambda^{(i)}(x)$
end procedure

In this algorithm $\Lambda^{(i)}(x) = \sum_j \lambda_j^{(i)} x^{q^j}$ and at the end of the algorithm, we will just collect the coefficient of the $\Lambda^{(i)}(x)$ to get our λ_i. Notice that on input we take $(s_0, \cdots, s_{2t-1}) = (u_{k+2t-1}, \cdots, u_k)$.

We summarize our decoding algorithm with the following steps in Algorithm 2. Suppose $(r_1, r_2, \cdots, r_n)$ was received with an error of rank $t < \frac{n-k+1}{2}$. We already know the matrix $\mathbf{U}^{-1}$ in advance.

Remark 9. The Step (3) in Algorithm 2 is similar to some step of a decoding algorithm of some classes of linear Hamming metric codes presented in [Bla79]. The step is called Syndrome extension while the whole algorithm is called Transform domain decoding. Transform domain decoding was extended to rank metric in [LSS14]. However, our algorithm as a whole is different as we use interpolation

Algorithm 2 Decoding algorithm

Input: $(r_1, \cdots, r_n)$

(1) Compute

$$\begin{pmatrix} f_0 + g_0 \\ \vdots \\ f_{k-1} + g_{k-1} \\ g_k \\ \vdots \\ g_{n-1} \end{pmatrix} = \mathbf{U}^{-1} \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$$

(2) Use the Berlekamp-Massey-like Algorithm 1 to get the λ_i's.

(3) Use the fact that the first row of the matrix $\mathbf{W}$ is a linear combination of the remaining rows, using the λ_i's, to recursively compute the remaining coefficients of $g(x)$. This is just like recursively computing elements of a sequence but the difference with linear-feedback shift register is that the steps also involves raising to some power of q.

(4) Output the message as $(f + g)(x) - g(x)$.

instead of syndrome computation in the first step.

3.3.1.3 Complexity and comparison with other algorithms

All the three first steps of Algorithm 2 have quadratic complexity i.e they can be done in $\mathcal{O}(n^2)$ operations in $\mathbb{F}_{q^n}$. The last step is a linear operation. Thus in general we have an algorithm with $\mathcal{O}(n^2)$ operations in $\mathbb{F}_{q^n}$.

We already saw two decoding algorithms at the beginning of Section 3.3. As we can see, there is a difference in the first steps of these algorithms and our algorithm. Instead of using an $((n-k) \times n)$ matrix for computing the syndromes, we use an $(n \times n)$ matrix to interpolate $f + g$. So in the first step, we have some $\mathcal{O}(nk)$ extra multiplications. The second steps are more or less the same as they are either Berlekamp-Massey or extended Euclidean algorithm. The last steps are where we may get the advantage as we directly use a linear recurrence to recover the error polynomial. For the other algorithms at the beginning of this Section 3.3, one first needs to compute the roots of some polynomials (error locator polynomial) before one can reconstruct the error vectors using some relations.

3.4 Extension to twisted Gabidulin codes

In this section, we will explain that our algorithm can also be modified to get a decoding algorithm for twisted Gabidulin codes. And in contrast to the algorithm in Section 3.2, we can do it for any parameters. We assume that the original message was given by

$$f(x) = f_0 x + \cdots + f_{k-1} x^{q^{k-1}} + \eta f_0^{q^r} x^{q^k}.$$

After the interpolation step, we get the polynomial $f(x) + g(x)$. In opposite to the case of Gabidulin codes, we do not know the value of g_k from this. However the problem we are faced remains similar. We want to find a linear relation between the rows of

$$\mathbf{W} = \begin{pmatrix} g_0 & g_{n-1}^q & \cdots & g_{k+t-1}^{q^{n-(k+t-1)}} & \cdots & g_k^{q^{n-k}} & \cdots & g_1^{q^{n-1}} \\ g_1 & g_0^q & \cdots & g_{k+t}^{q^{n-(k+t-1)}} & \cdots & g_{k+1}^{q^{n-k}} & \cdots & g_2^{q^{n-1}} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ g_{t-1} & g_{t-2}^q & \cdots & g_{k+2t-2}^{q^{n-(k+t-1)}} & \cdots & g_{k+t-1}^{q^{n-k}} & \cdots & g_t^{q^{n-1}} \\ g_t & g_{t-1}^q & \cdots & g_{k+2t-1}^{q^{n-(k+t-1)}} & \cdots & g_{k+t}^{q^{n-k}} & \cdots & g_{t+1}^{q^{n-1}} \end{pmatrix}.$$

where we know the values $g_{k+1}, \cdots, g_{n-1}$ and $\eta g_0^{q^r} - g^{q^k}$. We will see that we still can solve this problem. We have an equation of the form

$$(\lambda_0, \cdots, \lambda_t) \begin{pmatrix} g_{k+t}^{q^{n-(k+t)}} & g_{k+t-1}^{q^{n-(k+t-1)}} & \cdots & g_k^{q^{n-k}} \\ g_{k+t+1}^{q^{n-(k+t)}} & g_{k+t}^{q^{n-(k+t-1)}} & \cdots & g_{k+1}^{q^{n-k}} \\ \vdots & \ddots & \ddots & \vdots \\ g_{k+2t-1}^{q^{n-(k+t)}} & g_{k+2t-2}^{q^{n-(k+t-1)}} & \cdots & g_{k+t-1}^{q^{n-k}} \\ g_{k+2t}^{q^{n-(k+t-1)}} & g_{k+2t-1}^{q^{n-(k+t-1)}} & \cdots & g_{k+t}^{q^{n-k}} \end{pmatrix} = \mathbf{0}.$$

Notice that we introduce one more column in the equation. Again, by assumption, we have $2t < n - k + 1$. Thus $k + 2t \leq n$. If $k + 2t < n$, then a Berlekamp-Massey algorithm using the columns of the previous matrix except the last column is enough to compute the λ_i's. If $k + 2t = n$, then the equation becomes,

$$(\lambda_0, \cdots, \lambda_t) \begin{pmatrix} g_{k+t}^{q^{n-(k+t)}} & g_{k+t-1}^{q^{n-(k+t-1)}} & \cdots & g_k^{q^{n-k}} \\ g_{k+t+1}^{q^{n-(k+t)}} & g_{k+t}^{q^{n-(k+t-1)}} & \cdots & g_{k+1}^{q^{n-k}} \\ \vdots & \ddots & \ddots & \vdots \\ g_{n-1}^{q^{n-(k+t)}} & g_{n-2}^{q^{n-(k+t-1)}} & \cdots & g_{k+t-1}^{q^{n-k}} \\ g_0^{q^{n-(k+t)}} & g_{n-1}^{q^{n-(k+t-1)}} & \cdots & g_{k+t}^{q^{n-k}} \end{pmatrix} = \mathbf{0}, \tag{3.6}$$

where two entries in terms of g_k and g_0 are unknown.

If we use the columns of the matrix except the first and last columns, then we should have an underdetermined system of linear equations whose solution space is of dimension two. We assume that two linearly independent solutions are $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}'$. They can be found using the Berlekamp-Massey like algorithm again. Thus a solution of equation (3.6) is of the form $\boldsymbol{\lambda} + A\boldsymbol{\lambda}'$ for some $A \in \mathbb{F}_{q^n}$. Using this with the first column and the last column, we get two equations. Furthermore, we also know $\eta g_0^{q^r} - g_k$. So in total we get a system of three equations with three unknowns,

$$\begin{cases} h_0 + h_1 A + (h_2 + h_3 A) g_0^{q^{n-(k+t)}} & = 0 \\ h_4 + h_5 A + (h_6 + h_7 A) g_k^{q^{n-k}} & = 0\,. \\ h_8 + \eta g_0^{q^r} - g_k & = 0 \end{cases} \tag{3.7}$$

In this system, we know all the h_i and the variables g_0, g_k, A are unknown. Notice that any solution of the system of Equation (3.7) is a solution of the decoding algorithm. By the unique decoding property, there can only be one solution of this system.

To solve the system, we use the third equation in the two first equations and we get

$$\begin{cases} s_0 + s_1 A + (s_2 + s_3 A) g_0^{q^i} & = 0 \\ s_4 + s_5 A + (s_6 + s_7 A) g_0^{q^j} & = 0 \end{cases} \tag{3.8}$$

with the s_i's known. If $s_2 + s_3 A = 0$ or $s_6 + s_7 A = 0$, then we can use this to get the value of A. Else the first equation in Equation (3.8) gives us

$$g_0^{q^i} = \frac{s_0 + s_1 A}{s_2 + s_3 A}$$

And using this with the second equation in Equation (3.8), we get a one variable equation of the form, for some integer l,

$$\frac{s_0 + s_1 A}{s_2 + s_3 A} = \frac{s'_4 + s'_5 A^{q^l}}{s'_6 + s'_7 A^{q^l}}.$$

We want to point out that this form of equation was also obtained in Section 3.2. However, in the case here, we are sure that any solution would give us the closest codeword to the received message. We have now reduced the problem to solving the polynomial equation of the form

$$P(A) = u_0 + u_1 A + u_2 A^{q^l} + A^{q^l+1} = 0.$$

We distinguish three cases:

(I) If $u_0 = u_1 u_2$, then we can factor $P(A) = (A^{q^l} + u_1)(A + u_2)$.

(II) If $u_1 = u_2^q$, then

$$\begin{aligned} P(A) &= u_0 + u_2^{q^l} A + u_2 A^{q^l} + A^{q^l+1} \\ &= u_0 - u_2^{q^l} u_2 + (A + u_2)^{q^l+1} \end{aligned}$$

(III) If $u_0 \neq u_1 u_2$ and $u_1 \neq u_2^q$, then, from [Blu04], by a change of variable $y = (u_2 u_1 - u_0)(u_1 - u_2^{q^l})^{-1} A - u_2$, we will get a polynomial equation of the form

$$Q(y) = y^{q^l+1} - vy + v = 0$$

with $v = (u_1 - u_2^{q^l})^{q^l+1} / (u_0 - u_2 u_1)^{q^l}$.

First of all, it is easy to show that if we get A from $P(A)$, then we can use Equation (3.8) to get g_0. And we use Equation (3.7) to get g_k. These will give us the error polynomial $g(x)$ with the recurrence relation from Equation (3.6). So, normally, there should be only one unique solution for A. Now the question is how do we solve the equation $P(A) = 0$? Any of the three cases which produce multiple solutions should be ruled out. The first case of $P(A)$ is easy to solve. The two last cases reduce to polynomials of the form

$$P(X) = X^{q^l+1} + aX + b.$$

The number of roots of such polynomials was studied in [Blu04]. Here we will give a method to find these roots.

Suppose that y_2 is a root of $P(X)$. Then set $b = -y_2^{q^l+1} - ay_2$ and choose $y_1 = -a - y_2^{q^l}$. thus $b = y_2 y_1$. We get

$$\begin{aligned} (x^{q^l} - y_1 x) \circ (x^{q^l} - y_2 x) &= x^{q^{2l}} - y_2^{q^l} x^{q^l} - y_1 x^{q^l} + y_1 y_2 x \\ &= x^{q^{2l}} + a x^q + bx. \end{aligned}$$

The converse is also true. So, to get the root of $P(X)$, we just need to factor the linearized polynomial $x^{q^{2l}} + ax^q + bx$. In case this polynomial admits a root x_0 in $\mathbb{F}_{q^n}$ then we just take $y_2 = x_0^{q-1}$. Otherwise, we will need to use a factorization algorithm like in [Gie98].

Once A is computed, we can compute g_0 and g_k. Then we continue the decoding algorithm with the same methods as with the Gabidulin codes.

Remark 10. These algorithms can be easily modified to get a decoding algorithm for generalized (twisted) Gabidulin codes. Namely instead of working with the field automorphism x^q, we work with automorphisms of the form x^{q^s}.

3.5 Conclusion

In this chapter we have given two algorithms for decoding rank metric codes. The first algorithm was for twisted Gabidulin codes but it only works for some parameters. For the second algorithm, we provided a new decoding algorithm for Gabidulin codes and then we have shown how to modify it to get an algorithm which works with any parameters of twisted Gabidulin codes. In this second algorithm, instead of computing syndromes, we do some polynomial interpolation. Our algorithm requires more computations in this first step but we can compensate this in the last steps. Namely, there is no need to find roots of some "error locator polynomial". We just need to use a recurrence relation to recover the "error polynomial" after using a Berlekamp-Massey-like algorithm. We gave a brief analysis on the complexity and a comparison of our algorithm to some existing decoding algorithms for Gabidulin codes. Furthermore, we show that our algorithm can be modified to get a general decoding algorithm for twisted Gabidulin codes.

We have seen that our algorithm involves factoring linearized polynomial of degree q^2. It is well known that factoring a regular polynomial of degree 2 can be done by computing the discriminant of the polynomial. The algorithm presented in [Gie98] gives a factorization for linearized polynomials of general degree, we could further simplify our algorithm if we would have a discriminant like method to factorize a degree q^2 linearized polynomial.

Finally, we think that it is possible to get a version of our algorithm for Reed-Solomon codes. Namely we can use an equivalent of the Dickson matrix. In the case of Reed-Solomon codes, we have a circulant matrix. And a theorem of König-Rados gives a relation between the number of non-zero roots of a polynomial and the rank of some circulant matrix, see [LN96], Chapter 6, Section 1. It is known that the most expensive steps in the decoding of Reed-Solomon codes is finding roots of the error locator polynomials. This can be avoided in our algorithm. In the next chapter, we will see that there is a relation between the rank of the circulant matrix and the codewords it defines. The rank of circulant matrix in turn is equivalent to some linear complexity of finite sequence. This prompts us to study the property of finite sequences and their linear complexity.

Chapter 4

Coding Theory using Linear Complexity of Finite Sequences

This Chapter is based on my work in [Ran18].

4.1 Motivation

As we will explain in this section, the notion of weight of vectors is closely related to the notion of linear complexity of sequences. This motivates us to study the linear complexity of sequences as a new metric. For us to see this relation, let us first recall the construction of Reed-Solomon codes as we saw in Chapter 1, Example 1.

Let $\mathbb{F}_q$ be a finite field of size q. We consider Reed-Solomon codes of length $q-1$. So let $n = q-1$ and let $\alpha = (\alpha_1, \cdots, \alpha_n)$ be a vector where its elements are the non-zero elements of $\mathbb{F}_q$. We define the evaluation map as

$$\begin{aligned} ev_\alpha : \mathbb{F}_q[x] &\to \mathbb{F}_q^n \\ f(x) &\mapsto (f(\alpha_1), \cdots, f(\alpha_n)) \end{aligned}$$

Let $\mathbb{F}_q[x]_{<k}$ be the vector space of all polynomials of degree at most $k-1$. Then the image $\mathcal{C} = ev_\alpha\left(\mathbb{F}_q[x]_{<k}\right)$ is an MDS code. This comes from the fact that a polynomial of degree at most $k-1$ can have at most $k-1$ roots. The code we described is called Reed-Solomon codes.

It is this relation between the property of the roots of polynomial which is interesting for us. Let us see the following theorem of König-Rados. For a proof of this theorem, one can have a look at Chapter 6 of [LN96].

Theorem 16 (König-Rados). *Let $f(x) = a_0 + a_1 x + \cdots + a_{q-2}x^{q-2}$ be a polynomial over $\mathbb{F}$. Define the following matrix*

$$\mathbf{A} = \begin{pmatrix} a_0 & a_1 & \dots & a_{q-2} \\ a_1 & \iddots & \iddots & a_0 \\ \vdots & \iddots & \iddots & \vdots \\ a_{q-2} & a_0 & \dots & a_{q-3} \end{pmatrix}.$$

Suppose that the rank of $\mathbf{A}$ is equal to r. Then the number of roots of $f(x)$ in $\mathbb{F}_q^*$ is given by $q-1-r$.

The matrix $\mathbf{A}$ in the above theorem is a circulant matrix. It is easy to see that if its rank is equal to r, then the first r rows of $\mathbf{A}$ are linearly independent and the other rows are linear combination of them. Furthermore, this tells us that the coefficients of $f(x)$ satisfy the following property.

$$a_{i+r} = \sum_{j=0}^{r-1} c_j a_{i+j}, \quad \forall i \in \mathbb{N}.$$

Note that the coefficients $a_0, \cdots, a_{q-2}$ satisfy a recurrence relation of order r. Using the definitions which we will see in Section 4.2, we say that the coefficients of the polynomials $f(x)$ can be generated by a linear-feedback shift register (LFSR) of order r and this is the minimum possible for r. We say that $(a_0, \cdots, a_{q-2})$ has linear complexity r. Moreover, our sequence gives a periodic sequence with period $q-1$. To summarize, we have the following theorem, which is a direct consequence of the theorem of König-Rados.

Theorem 17. *Let $f(x) = a_0 + a_1 x + \cdots + a_{q-2}x^{q-2}$ be a polynomial over $\mathbb{F}_q$. If $f(x)$ has $q-1-r$ roots, then $(a_0, \cdots, a_{q-2})$ has linear complexity r and the evaluation $(f(\alpha_1), \cdots, f(\alpha_n))$ has weight r.*

A more general version of Theorem 16 (or equivalently Theorem 17) is known as Blahut's theorem. It is for example stated in [Bla79] where they use some notion of discrete Fourier transform. In this thesis, we will use the version as stated in Theorems 16 and 17.

Through Theorem 17, we can relate the linear complexity of a periodic sequence with the weight of a vector. However, we have only periodic sequences. This raises the following question: What happens if we study any type of sequence i.e. we don't require the LFSR to be a periodic sequence with fixed period. We will answer this question in the next sections. First, in Section 4.2, we will introduce the notion of linear-feedback shift register. In Section 4.3, we will give a coding theory for finite sequences. We will use Section 4.4 for a

separate study on the number of finite sequences which can be generated by an LFSR of given order. Finally, we will conclude with Section 4.5 and give some future work.

4.2 Linear-feedback shift register

Let $\mathbb{F}_q$ be a finite field with q elements.

Definition 26. Let $\mathbb{F}_q$ be a field. A linear feedback shift register (LFSR) sequence over $\mathbb{F}_q$ is an infinite sequence (a_i) over $\mathbb{F}_q$ such that there are fixed $c_j \in \mathbb{F}_q$ with,

$$a_{i+l} = \sum_{j=0}^{l-1} c_j a_{i+j}, \quad \forall i \in \mathbb{N}.$$

We also say that the sequence (a_i) is generated by an LFSR of order l. The vector $(a_0, \cdots, a_{l-1})$ is the initial state of the LFSR and the c_i's are its coefficients.

The feedback polynomial associated to (a_i) is

$$f(z) = z^l - \sum_{j=0}^{l-1} c_j z^j.$$

Definition 27. Let (a_i) be a sequence generated by an LFSR over $\mathbb{F}_q$. The generating function $A(z)$ associated to (a_i) is the formal power series

$$A(z) = \sum_{i=0}^{\infty} a_i z^i.$$

One can show (Chapter 8, [LN96]) that for some polynomial $g(z)$ of degree $l-1$ at most, we have

$$A(z) = \frac{g(z)}{f^*(z)},$$

where f^* is the reciprocal polynomial given by

$$f^*(z) = z^l f\left(\frac{1}{z}\right).$$

Definition 28. Given a non-zero finite sequence $(a_i) = (a_0, \cdots, a_{n-1}) \in \mathbb{F}_q^n$, the linear complexity $\mathfrak{L}(a_i)$ of the sequence is the smallest l such that

$$a_{i+l} = \sum_{j=0}^{l-1} c_j a_{i+j}, \quad \forall i,\ 0 \leq i \leq n-1-l,$$

for some fixed $c_j \in \mathbb{F}_q$. In other words, it is the order of the smallest LFSR which can generate (a_i).

For a zero sequence, we set the linear complexity to be equal to zero.

Given a finite sequence, we can compute the shortest LFSR producing this sequence. This can be done using the Berlekamp-Massey algorithm in $\mathcal{O}(n^2)$ field operations in $\mathbb{F}_q$ (Chapter 8, [LN96]). Furthermore if the linear complexity is $n/2$, then n successive terms of the sequence are enough to uniquely find the shortest shift register. We present the algorithm in Algorithm 3. On input, we have a sequence $s_0, \cdots, s_{n-1}$ of length n. On output, the algorithm generates the order and the feedback polynomial $f(z)$ of the shortest LFSR generating $s_0, \cdots, s_{n-1}$.

Algorithm 3 Berlekamp-Massey

procedure BERLEKAMP-MASSEY($s_0, \cdots, s_{n-1}$)
  $f(z) \leftarrow 1$, $A(z) \leftarrow 1$,
  $L \leftarrow 0$, $m = -1$, $e \leftarrow 1$
  for i from 0 to $n-1$ **do**
    $d \leftarrow s_i + \sum_{j=1}^{L} f_j s_{i-j}$
    if $d \neq 0$ **then**
      $B(z) \leftarrow f(z)$
      $f(z) \leftarrow f(z) - (d/e)A(z)z^{i-m}$
      if $2L \leq i$ **then**
        $L \leftarrow i + 1 - L$
        $m \leftarrow i$
        $A(z) \leftarrow B(z)$
        $e \leftarrow d$
      end if
    end if
  end for
  return L and $f(z)$
end procedure

Proposition 10. *Let $(a_i) = (a_0, \cdots, a_{n-1})$ be a finite sequence of length n over $\mathbb{F}_q$. Then the linear complexity $\mathfrak{L}(a_i)$ satisfies $\mathfrak{L}(a_i) \leq n$. Furthermore the only sequences attaining the upper bound n are of the form $(0, \cdots, 0, a)$, with $a \in \mathbb{F}_q^*$.*

Proof. We can just use an LFSR with $(a_0, \cdots, a_{n-1})$ as initial state so that the maximum linear complexity is at most n. It is obvious that $(0, \cdots, 0, a)$ has linear complexity n. Finally, if $(a_i) = (a_0, \cdots, a_{n-1})$ is such that $a_j \neq 0$ for some j with $0 \leq j \leq n-2$, then take $c_i = 0$ for $i \neq j$. Define $c_j = a_{n-1}/a_j$. We prove that $a_{n-1} = \sum_{j=0}^{n-2} c_j a_j$ so that the linear complexity is at least $n-1$. □

The key property of the linear complexity of sequences which will be used later is the following known theorem.

Theorem 18. *Let (a_i) and (b_i) be two finite sequences. If $(c_i) = (a_i) + (b_i)$, then*

$$\mathfrak{L}(c_i) \leq \mathfrak{L}(a_i) + \mathfrak{L}(b_i).$$

Proof. Suppose that the generating function of the LFSR generating (a_i) and (b_i) are respectively

$$\frac{g_a(z)}{f_a^*(z)}, \quad \text{and } \frac{g_b(z)}{f_b^*(z)}.$$

Then the generating function of the LFSR generating (c_i) is

$$\frac{g_a(z)f_b^*(z) + g_b(z)f_a^*(z)}{f_a^*(z)f_b^*(z)}.$$

And therefore, (c_i) can be generated by an LFSR with feedback polynomial $f_a(z)f_b(z)$. Therefore the linear complexity is at most $\mathfrak{L}(a_i) + \mathfrak{L}(b_i)$. □

4.3 A coding theory for finite sequences using the linear complexity

Let $\mathbb{F}_q$ be a finite field and let n be a positive integer. We will consider sets of sequences of length n.

Definition 29. Let $(a_i) = (a_0, \cdots, a_{n-1}) \in \mathbb{F}_q^n$ and $(b_i) = (b_0, \cdots, b_{n-1}) \in \mathbb{F}_q^n$ be two finite sequences of n elements of $\mathbb{F}_q$ each. Then we define a distance on $\mathbb{F}_q^n$ by the following,

$$\mathbf{d}((a_i), (b_i)) = \mathfrak{L}((a_i) - (b_i)),$$

where $\mathfrak{L}(0) = 0$.

This map defines indeed a distance:

(i) By definition, $\mathbf{d}((a_i), (b_i)) = 0 \Leftrightarrow (a_i) = (b_i)$.

(ii) By definition of $\mathfrak{L}$, $\mathfrak{L}(a_i) \geq 0$.

(iii) $\mathbf{d}((a_i), (b_i)) = \mathbf{d}((b_i), (a_i))$.

(iv) For the triangular inequality,

$$\begin{aligned}
\mathbf{d}((a_i), (b_i)) &= \mathfrak{L}((a_i) - (b_i)) \\
&= \mathfrak{L}((a_i) - (c_i) + (c_i) - (b_i)) \\
&\leq \mathfrak{L}((a_i) - (c_i)) + \mathfrak{L}((c_i) - (b_i)), \text{ by Theorem 18} \\
&= \mathbf{d}((a_i), (c_i)) + \mathbf{d}((c_i), (b_i)).
\end{aligned}$$

Like in coding theory, we can define a subset of $\mathbb{F}_q^n$ and define the metric $\mathbf{d}$ on this set. We will derive basic coding results for this context.

Definition 30. Let S be a subset of $\mathbb{F}_q^n$. The minimum distance d of S is the minimum of $\mathbf{d}((a_i),(b_i))$ for distinct $(a_i),(b_i) \in S$. We will describe the parameters of S as $[n, \sharp S, d]$. In case S is a k-dimensional subspace of $\mathbb{F}_q^n$, then, by additivity, d is the minimum linear complexity of the non-zero sequences in S and we will write $[n,k,d]$.

For the next steps we want to have a look at the bounds on a $[n, \sharp S, d]$-subset of $\mathbb{F}_q^n$.

Theorem 19 (Singleton bound)**.** *Let $\mathbb{F}_q$ be a finite field of size q. Let $S \subset \mathbb{F}_q^n$ be a set of finite sequence over $\mathbb{F}_q$ of length n, with minimum distance d. Then $\sharp S \leq q^{n-d+1}$.*

Proof. Let us define the following linear map P as

$$P : \mathbb{F}_q^n \to \mathbb{F}_q^{n-d+1}$$

$$(a_0, \cdots, a_{n-1}) \to \begin{pmatrix} 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} a_0 & \dots & a_{n-d} \\ \vdots & \ddots & \vdots \\ a_{d-1} & \dots & a_{n-1} \end{pmatrix}$$

The restriction of this map must be injective on S. Otherwise if two sequences (a_i) and (b_i) are mapped to the same image, then $(a_i) - (b_i)$ is mapped to zero. But this would imply that $\mathfrak{L}((a_i) - (b_i)) \leq d-1$. This is in contradiction with the minimum distance of S. By the injectivity, we must have that $\sharp S \leq \sharp(\mathbb{F}_q^{n-d+1})$. □

Note that in this proof, instead of using $\begin{pmatrix} 1 & \dots & 1 \end{pmatrix}$, we can use any vector with 1 as last entry. These operations are equivalent to the puncturing operation on codes. Namely using $\begin{pmatrix} 0 & \dots & 0 & 1 \end{pmatrix}$ is just puncturing at the first $d-1$ positions.

Remark 11. In case S is linear of dimension k over $\mathbb{F}_q$, then $k \leq n-d+1$.

Definition 31 (Optimal set of sequences - OSS)**.** We call a set of sequences S optimal if the minimum distance of S reaches the bound of the previous theorem i.e. if S has elements of length n and minimum distance d and $\sharp S = q^{n-d+1}$.

Example 3. Let S be the set of sequences of length n over a finite field $\mathbb{F}_q$ defined by

$$S = \{(0, \cdots, 0, a_1, \cdots, a_k) : a_i \in \mathbb{F}_q\}.$$

Then, S is an optimal set of linear sequences of dimension k. That is because the sequences cannot be generated by an LFSR of order smaller than $n-k+1$ except when it is the zero sequence.

The nice property of using the set of sequences with the linear complexity as metric is that, in opposite to maximum distance separable codes, we can have optimal set of sequences for any parameters. We can make the construction, even for the binary field.

Decoding of OSS

The decoding of OSS given in Example 3 is straightforward. First let us look at the unique decoding property.

Proposition 11. *Suppose that S is an $[n, M, d]$ set of sequences. Suppose that $\mathbf{y} \in \mathbb{F}_q$ is equal to $\mathbf{x} + \mathbf{e}$, where $\mathbf{x} \in S$ and $\mathfrak{L}(\mathbf{e}) < \frac{d}{2}$. Then, the decomposition $\mathbf{x} + \mathbf{e}$ is unique.*

Proof. If $\mathbf{y} = \mathbf{x}_1 + \mathbf{e}_1 = \mathbf{x}_2 + \mathbf{e}_2$, then $\mathbf{x}_1 - \mathbf{x}_2 = \mathbf{e}_2 - \mathbf{e}_1$. Therefore $\mathbf{d}(x_1, x_2) = \mathfrak{L}(\mathbf{e}_2 - \mathbf{e}_1)$. By Theorem 18, $d(x_1, x_2) \leq \mathfrak{L}(\mathbf{e}_2) + \mathfrak{L}(\mathbf{e}_1) < d$. This is in contradiction with the minimum distance of S. □

Let S, of dimension k, be the OSS in Example 3. Suppose that we know $\mathbf{y} = \mathbf{x} + \mathbf{e}$ with $\mathbf{x} \in S$ and $\mathfrak{L}(\mathbf{e}) < \frac{n-k+1}{2}$. By Proposition 11, we know that $\mathbf{e}$ is unique. Since the $n - k$ first entries of $\mathbf{x}$ are equal to zero. Then we know the first $n - k$ entries of $\mathbf{e}$. Now, since $\mathfrak{L}(\mathbf{e}) < \frac{n-k+1}{2}$, then we can uniquely recover the LFSR generating $\mathbf{e}$ by using the Berlekamp-Massey algorithm on the first $n - k$ entries of e. We are therefore able to produce the whole $\mathbf{e}$ and then we compute $\mathbf{x} = \mathbf{y} - \mathbf{e}$.

Remark 12. We can modify the above decoding algorithm to get a decoding algorithm for the Reed-Solomon code in Section 4.1. The extra step is just that we need to interpolate a received codewords first to get a polynomial $f(x)$ of degree $q - 2$ at most. After this we apply the decoding algorithm for the OSS we gave above on the coefficients of this polynomial $f(x)$. Notice that the Berlekamp-Massey in this case is applied to the last coefficients of the polynomial $f(x)$.

4.4 Number of finite sequences generated by a linear feedback shift register with a given order

LFSR already has applications in cryptography. For instance, it is used when one wants to generate random keys. As we have seen, one can compute the linear complexity of a sequence using the Berlekamp-Massey algorithm. Thus, if

a sequence has small linear complexity, one can easily find an LFSR generating this sequence. Due to this fact, we usually want to have sequences with large linear complexity. Therefore, one important question is to know how many finite sequences have large linear complexity. Another motivation for this section is also that knowing the number of sequences with a given linear complexity is important for the security aspect of a code-based cryptosystem using linear complexity as metric.

Lemma 15. *Let (a_i) be an infinite sequence. If (a_i) can be generated by an LFSR of order n, then (a_i) can be generated by an LFSR of order i, for any $i \geq n$.*

Proof. For a proof of this, if $c_1, \cdots, c_n$ are the coefficients of the LFSR of order n, then $0, \cdots, 0, c_1, \cdots, c_i$ are the coefficients of the larger LFSRs. □

By Lemma 15, we can just study the number of sequences which can be generated by an LFSRs of order r to know the number of sequences which has linear complexity smaller or equal to r. Studying sequences which can be generated by an LFSRs of order r can be in turn translated to studying some matrix $\mathbf{A}$ of the form

$$\mathbf{A} = \begin{pmatrix} a_0 & a_1 & \dots & \dots & a_{n-r-1} \\ a_1 & \iddots & \iddots & a_{n-r-1} & a_{n-r} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ a_r & a_{r+1} & \dots & \dots & a_{n-1} \end{pmatrix}$$

We just need the condition that the last row is a linear combination of the previous rows. The matrices with the form of $\mathbf{A}$ are called *Hankel* matrices when they are square matrices. A similar problem, which was counting the number of singular Hankel matrices, were solved by observing the Berlekamp-Massey algorithm [CK12]. However, we will use another method. We give a general definition for rectangular matrices.

Definition 32. An $(m \times n)$ matrix $\mathbf{A}$ is called a Hankel matrix if there is a sequence $(a_0, a_1, \cdots, a_{m+n-2})$ such that $\mathbf{A}_{i,j} = a_{i+j-2}$ for $1 \leq i \leq m$ and $1 \leq j \leq n$.

In [Day60], Daykin called the general rectangular matrices *persymmetric matrices*. To go further with our counting, we will need the following reduction method as used by Daykin in [Day60].

Fix and integer u such that $0 \leq u < \min(r, n-r-1)$. We define the following set

$$\mathcal{A}_u = \{(0, \cdots, 0, a_u, \cdots, a_{n-1}) : a_u \neq 0\}.$$

Then for $(a_i) \in \mathcal{A}_u$, we recursively define θ_i, $i = 0, \cdots, n-u-1$ by

$$\begin{cases} a_u\theta_0 = 1 \\ \sum_{l=0}^{i} a_{u+l}\theta_{i-l} = 0 \end{cases}.$$

Now define the following matrices

$$\mathbf{U} = \begin{pmatrix} \theta_0 & 0 & 0 & \dots & 0 \\ \theta_1 & \theta_0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \theta_r & \dots & \dots & \theta_1 & \theta_0 \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} \theta_0 & \theta_1 & \dots & \dots & \theta_{n-r-1} \\ 0 & \theta_0 & \theta_1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \theta_1 \\ 0 & \dots & 0 & 0 & \theta_0 \end{pmatrix}$$

$$\mathbf{X} = \begin{pmatrix} 0 & \dots & 0 & 0 & \theta_0 \\ \vdots & \iddots & 0 & \theta_0 & \theta_1 \\ 0 & \iddots & \iddots & \iddots & \vdots \\ 0 & \theta_0 & \iddots & \iddots & \vdots \\ \theta_0 & \theta_1 & \dots & \dots & \theta_u \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} \theta_{u+2} & \theta_{u+3} & \dots & \dots & \theta_{n-r} \\ \theta_{u+3} & \iddots & \iddots & \theta_{n-r-1} & \theta_{n-r+1} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ \vdots & \theta_{r+1} & \iddots & \iddots & \theta_{n-u-2} \\ \theta_{r+1} & \theta_{r+2} & \dots & \theta_{n-u-2} & \theta_{n-u-1} \end{pmatrix}$$

Lemma 16. *For a fixed u with $0 \leq u < \min(r, n-r-1)$, there is a bijection between $\mathcal{A}_u$ and the set $\{(\theta_0, \cdots, \theta_{n-u-1}) : \theta_0 \in \mathbb{F}_q^*, \theta_i \in \mathbb{F}_q, 1 \leq i \leq n-u-1\}$ given by*

$$\mathbf{UAV} = \begin{pmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & -\mathbf{Y} \end{pmatrix}.$$

Proof. First let us show that

$$\mathbf{UAV} = \begin{pmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & -\mathbf{Y} \end{pmatrix}.$$

We know that $\mathbf{A}_{i,k} = a_{i+k}$ for $0 \leq i \leq r$ and $0 \leq k \leq n-r-1$. For the matrix $\mathbf{V}$, the entries satisfies $\mathbf{V}_{j,k} = 0$ if $j < k$ and $\mathbf{V}_{j,k} = \theta_{k-j}$ if $k \leq j$. And for the matrix $\mathbf{U}$, $\mathbf{U}_{i,k} = 0$ if $k > i$ and $\mathbf{U}_{i,k} = \theta_{i-k}$ if $k \leq i$. Thus

$$(\mathbf{UAV})_{i,j} = \sum_{k=0}^{i} \theta_{i-k} \left[\sum_{l=0}^{j} a_{k+l}\theta_{j-l} \right], \quad \begin{cases} 0 \leq i \leq r \\ 0 \leq j \leq n-r-1 \end{cases}.$$

We are now going to look at three different cases:

- Suppose that $i \leq u$, Since $a_0 = a_1 = \cdots = a_{u-1} = 0$, then

$$(\mathbf{UAV})_{i,j} = \sum_{k=0}^{i} \theta_{i-k} \left[\sum_{l=u-k}^{j} a_{k+l}\theta_{j-l} \right], \quad \begin{cases} 0 \leq i \leq r \\ 0 \leq j \leq n-r-1 \end{cases}.$$

After a change of variable

$$(\mathbf{UAV})_{i,j} = \sum_{k=0}^{i} \theta_{i-k} \left[\sum_{l=0}^{j-u+k} a_{u+l}\theta_{j-u+k-l} \right], \quad \begin{cases} 0 \leq i \leq r \\ 0 \leq j \leq n-r-1 \end{cases}.$$

By the recurrence relation on the θ_i's, we know that

$$\sum_{l=0}^{j-u+k} a_{u+l}\theta_{j-u+k-l} = \begin{cases} 1, & \text{if } j-u+k=0 \\ 0, & \text{otherwise} \end{cases}$$

And thus

$$(\mathbf{UAV})_{i,j} = \begin{cases} \theta_{i+j-u}, & \text{if } 0 \leq u-j \leq i \\ 0, & \text{otherwise} \end{cases}$$

- Now, suppose that $i > u$ and $j \leq u$. Since, $j \leq u$, then we can use the expression

$$(\mathbf{UAV})_{i,j} = \sum_{l=0}^{j} \theta_{j-l} \left[\sum_{k=0}^{i} a_{k+l}\theta_{i-k} \right], \quad \begin{cases} 0 \leq i \leq r \\ 0 \leq j \leq n-r-1 \end{cases}.$$

We use the same transformation as before to get

$$(\mathbf{UAV})_{i,j} = \sum_{l=0}^{j} \theta_{j-l} \left[\sum_{k=0}^{i-u+l} a_{u+k}\theta_{i-u+l-k} \right], \quad \begin{cases} 0 \leq i \leq r \\ 0 \leq j \leq n-r-1 \end{cases}.$$

and

$$\sum_{k=0}^{i-u+l} a_{u+k}\theta_{i-u+l-k} = \begin{cases} 1, & \text{if } i-u+l=0 \\ 0, & \text{otherwise.} \end{cases}$$

Since $u < i$, then the first case is never possible, therefore $(\mathbf{UAV})_{i,j}$ is always zero.

- Finally, suppose that $i > u$ and $j > u$. We have

$$\begin{aligned} (\mathbf{UAV})_{i,j} &= \sum_{l=0}^{j} \theta_{j-l} \left[\sum_{k=0}^{i} a_{k+l}\theta_{i-k} \right] \\ &= (\mathbf{UAV})_{i,u} + \sum_{l=u+1}^{j} \theta_{j-l} \left[\sum_{k=0}^{i} a_{k+l}\theta_{i-k} \right] \\ &= \sum_{l=u+1}^{j} \theta_{j-l} \left[\sum_{k=u-l}^{i} a_{k+l}\theta_{i-k} - \sum_{k=u-l}^{-1} a_{k+l}\theta_{i-k} \right]. \end{aligned}$$

The last equality contains the subtraction because by starting k with $u-l$, we have some negative value for the index k, so we have to remove them. Finally, we have

$$\begin{aligned}(\mathbf{UAV})_{i,j} &= -\sum_{l=u+1}^{j} \theta_{j-l} \sum_{k=u-l}^{-1} a_{k+l}\theta_{i-k} \\ &= -\sum_{k=u-j}^{-1} \theta_{i-k} \sum_{l=u-k}^{j} a_{k+l}\theta_{j-l} \\ &= -\sum_{k=u-j}^{-1} \theta_{i-k} \sum_{l=0}^{j-u+k} a_{u+l}\theta_{j-u+k-l}.\end{aligned}$$

By the recurrence relation on the θ_i's, we have

$$(\mathbf{UAV})_{i,j} = -\theta_{i+j-u}.$$

For the bijection, suppose that $\mathbf{A}$ and $\mathbf{A}'$ both give the same θ_i's, then $\mathbf{UAV} = \mathbf{UA'V}$, but since $\mathbf{U}$ and $\mathbf{V}$ are invertible, then $\mathbf{A} = \mathbf{A}'$. We have an injection between two sets of the same size, therefore it is a bijection. □

Lemma 17. *Suppose that $\mathbf{A}$ is the matrix corresponding to the sequence $(a_i) \in \mathcal{A}_u$, and it corresponds to the matrices $\mathbf{U}, \mathbf{V}, \mathbf{X}, \mathbf{Y}$. Then the last row of $\mathbf{A}$ is a linear combination of its other rows if and only if the last row of $\mathbf{Y}$ is a linear combination of its other rows.*

Proof. We know that

$$\mathbf{AV} = \mathbf{U}^{-1}\begin{pmatrix}\mathbf{X} & \mathbf{0} \\ \mathbf{0} & -\mathbf{Y}\end{pmatrix}$$

Thus if $(\lambda_0, \cdots, \lambda_{r-1}, 1)\mathbf{AV} = \mathbf{0}$, then

$$(\lambda_0, \cdots, \lambda_{r-1}, 1)\mathbf{U}^{-1}\begin{pmatrix}\mathbf{X} & \mathbf{0} \\ \mathbf{0} & -\mathbf{Y}\end{pmatrix} = \mathbf{0}$$

Therefore, there is some non-zero μ_r with,

$$(\mu_0, \cdots, \mu_{r-1}, \mu_r)\begin{pmatrix}\mathbf{X} & \mathbf{0} \\ \mathbf{0} & -\mathbf{Y}\end{pmatrix} = \mathbf{0}.$$

Since $\mu_r \neq 0$, then the last row of $\mathbf{Y}$ is a linear combination of its previous row. The converse can be proven by going backward. □

Definition 33. Define $B(n,r,u)$ to be the set of non-zero sequences (a_i) of length n with linear complexity r at most such that u is the smallest index i such that a_i is non-zero. We also define $B(n,r)$ to be the set of all sequences (a_i) of length n with linear complexity at most r. Therefore $B(n,r) = \left(\cup_{u=0}^{n-1} B(n,r,u)\right) \cup \{\mathbf{0}\}$. We set $b(n,r,u) = \sharp B(n,r,u)$ and $b(n,r) = \sharp B(n,r)$.

Now, suppose that $r+1 \leq n-r$. If $u < r$, then we use the above method for reduction. Otherwise if $u \geq r$, then the first r elements of (a_i) are 0 and therefore we can only get the zero sequence.

Next, if $n-r \leq r$, then again we use the above reduction method for $u < n-r-1$. If $r > u \geq n-r-1$, then for any choice of the remaining coefficients $a_{u+1}, \cdots, a_{n-1}$, it is always possible to generate it using an LFSR of order r. If $u \geq r$, then there is no LFSR of order at most r which can generate the sequence.

These, together with Lemmas 15 and 17 allow us to get the next theorem.

Theorem 20.

(i) If $r+1 \leq n-r$ and $u \geq r$, then $b(n,r,u) = 0$.

(ii) If $n-r \leq r$ and $r > u \geq n-r-1$, then $b(n,r,u) = q^{n-u-1}(q-1)$.

(iii) If $n-r \leq r$ and $u \geq r$, then $b(n,r,u) = 0$.

(iv) If $r+1 \leq n-r$ and $u < r$, or $n-r \leq r$ and $u < n-r-1$ then $b(n,r,u) = q^{u+1}(q-1)b(n-2u-2, r-u-1)$.

Proof. Lemma 15 tells us all elements of $B(n,r)$ can be generated by an LFSR of order r. Hence, we study only matrices in the form of $\mathbf{A}$. For (i), one can just write down the matrix $\mathbf{A}$ and see that it has a triangular shape where the last row is never a linear combination of the previous row. For (ii), we again look at the form of the matrix $\mathbf{A}$, we will see that some first non-zero rows of $\mathbf{A}$ make an invertible matrix and thus the last row is always a linear combination of the rows of that invertible matrix whatever the choice of the coefficients we choose after a_u. For (iii), looking at the form of the matrix will also give the result. For (iv), we use the bijection in Lemma 16 and Lemma 17. □

Summing all the possibilities in Theorem 20, we get the following corollaries.

Corollary 2. *Given two integers $r \leq n$, the number of finite sequence of length n with linear complexity at most r is equal to*

(i) If $r = 0$, $b(n,0) = 1$.

(ii) If $r+1 \leq n-r$,

$$b(n,r) = 1 + \sum_{u=0}^{r-1} q^{u+1}(q-1)b(n-2u-2, r-u-1).$$

(iii) If $n-r \leq r$,

$$b(n,r) = 1 + \sum_{u=0}^{n-r-2} q^{u+1}(q-1)b(n-2u-2, r-u-1) + \sum_{u=n-r-1}^{r-1} (q-1)q^{n-u-1}.$$

Corollary 3. *Given two integers* $r \leq n$, *the number of finite sequences of length* n *with linear complexity at most* r *is equal to* 1 *if* $r = 0$ *and if* $0 < r \leq n$,

$$b(n,r) = 1 - q + q^2 b(n-2, r-1).$$

Proof. Suppose that $r+1 \leq n-r$. Then,

$$\begin{aligned}
b(n,r) &= 1 + \sum_{u=0}^{r-1} q^{u+1}(q-1)b(n-2u-2, r-u-1) \\
&= 1 + q(q-1)b(n-2, r-1) \\
&\quad + \sum_{u=1}^{r-1} q^{u+1}(q-1)b(n-2u-2, r-u-1) \\
&= 1 + q(q-1)b(n-2, r-1) \\
&\quad + q\sum_{u=0}^{r-2} q^{u+1}(q-1)b(n-2u-4, r-u-2) \\
&= 1 + q(q-1)b(n-2, r-1) + qb(n-2, r-1) - q \\
&= 1 - q + q^2 b(n-2, r-1)
\end{aligned}$$

Now suppose that $n-r \leq r$. Then,

$$\begin{aligned}
b(n,r) &= 1 + \sum_{u=n-r-1}^{r-1} (q-1)q^{n-u-1} \\
&\quad + \sum_{u=0}^{n-r-2} q^{u+1}(q-1)b(n-2u-2, r-u-1) \\
&= 1 + \sum_{u=n-r-1}^{r-1} (q-1)q^{n-u-1} + q(q-1)b(n-2, r-1) \\
&\quad + q\sum_{u=1}^{n-r-2} q^{u}(q-1)b(n-2u-2, r-u-1)
\end{aligned}$$

$$
\begin{aligned}
&= 1 + \sum_{u=n-r-1}^{r-1} (q-1)q^{n-u-1} + q(q-1)b(n-2, r-1) \\
&\quad + q \sum_{u=0}^{n-r-3} q^{u+1}(q-1)b(n-2u-4, r-u-2) \\
&= 1 + \sum_{u=n-r-1}^{r-1} (q-1)q^{n-u-1} + q(q-1)b(n-2, r-1) \\
&\quad + q \left[b(n-2, r-1) - 1 - \sum_{u=n-r-2}^{r-2} (q-1)q^{n-u-3} \right] \\
&= 1 - q + q^2 b(n-2, r-1) + \sum_{u=n-r-1}^{r-1} (q-1)q^{n-u-1} \\
&\quad - q \sum_{u=n-r-2}^{r-2} (q-1)q^{n-u-3} \\
&= 1 - q + q^2 b(n-2, r-1) \\
&\quad + (q-1) \left[\sum_{u=n-r-1}^{r-1} q^{n-u-1} - \sum_{u=n-r-2}^{r-2} q^{n-u-2} \right] \\
&= 1 - q + q^2 b(n-2, r-1) \\
&\quad + (q-1) \left[\sum_{u=n-r-1}^{r-1} q^{n-u-1} - \sum_{u=n-r-1}^{r-1} q^{n-u-1} \right] \\
&= 1 - q + q^2 b(n-2, r-1).
\end{aligned}
$$

□

As a consequence of the corollaries, we get the following theorem.

Theorem 21. *Given two integers $r \leq n$, the number of finite sequences of length n with linear complexity at most r is*

(i) If $r = 0$, $b(n, 0) = 1$.

(ii) If $r + 1 \leq n - r$,

$$b(n, r) = \frac{q^{2r+1} + 1}{q + 1}.$$

(iii) If $n - r \leq r$,

$$b(n, r) = \frac{1 - q^{2(n-r)}}{1 + q} + q^n.$$

Proof. Suppose that $r+1 \leq n-r$. From the previous corollary,

$$\sum_{i=0}^{r-1} q^{2i} b(n-2i, r-i) = (1-q)\sum_{i=0}^{r-1} q^{2i} + \sum_{i=0}^{r-1} q^{2(i+1)} b(n-2(i+1), r-(i+1)).$$

Thus

$$\sum_{i=0}^{r-1} q^{2i} b(n-2i, r-i) = (1-q)\sum_{i=0}^{r-1} q^{2i} + \sum_{i=1}^{r} q^{2i} b(n-2i, r-i).$$

Therefore,

$$b(n,r) = (1-q)\sum_{i=0}^{r-1} q^{2i} + q^{2r} = q^{2r} + \frac{1-q^{2r}}{1+q}.$$

And we get the result.

If $n-r \leq r$, then $r \geq \frac{n}{2}$. So using this,

$$\sum_{i=0}^{n-r-1} q^{2i} b(n-2i, r-i) = (1-q)\sum_{i=0}^{n-r-1} q^{2i} + \sum_{i=0}^{n-r-1} q^{2(i+1)} b(n-2(i+1), r-(i+1)).$$

Therefore

$$b(n,r) = (1-q)\sum_{i=0}^{n-r-1} q^{2i} + q^{2(n-r)} b(2r-n, 2r-n),$$

Since $B(2r-n, 2r-n) = \mathbb{F}^{2r-n}$, then

$$b(n,r) = (1-q)\frac{1-q^{2(n-r)}}{1-q^2} + q^n.$$

And we get our result.

□

Using the previous theorem, we can compute the number of finite sequences with a fixed linear complexity.

Theorem 22. *Let $r \leq n$ be positive integers. Then, the number of sequences of length n and linear complexity r over a finite field $\mathbb{F}$ of size q is*

$$\begin{cases} 1 & \text{if } r = 0, \\ q^{2r-1}(q-1) & \text{if } r \leq \lfloor \frac{n}{2} \rfloor, \\ q^{2(n-r)}(q-1) & \text{if } r > \lfloor \frac{n}{2} \rfloor. \end{cases}$$

Proof. The case $r = 0$ is clear. For $r = 1$, we get that the number of sequences of length n and linear complexity r over a finite field $\mathbb{F}$ of size q is

$$\frac{q^3 - q}{q + 1} = q(q - 1).$$

Now, suppose that $r = \lceil \frac{n}{2} \rceil$. Then the number we want is given by

$$q^n - \frac{q^{2(n-r)} + q^{2r-1}}{q+1} = \begin{cases} q^{2r-1}(q-1) & \text{if } n \text{ is even,} \\ q^{2(n-r)}(q-1) & \text{if } n \text{ is odd.} \end{cases}$$

It is easy to check that if $r \leq \lceil \frac{n}{2} \rceil - 1$, then the number is

$$q^{2r-1}(q-1),$$

and if $\lceil \frac{n}{2} \rceil + 1$, the number is

$$q^{2(n-r)}(q-1).$$

Furthermore $\{r \leq \lceil \frac{n}{2} \rceil - 1\}$ and $\{r = \lceil \frac{n}{2} \rceil, n \text{ even}\}$ are the same as $\{r \leq \lfloor \frac{n}{2} \rfloor\}$. Finally $\{r \geq \lceil \frac{n}{2} \rceil + 1\}$ and $\{r = \lceil \frac{n}{2} \rceil, n \text{ odd}\}$ are the same as $\{r > \lfloor \frac{n}{2} \rfloor\}$. □

Since we also know the size of balls with respect to the linear complexity from Theorem 21, we can give a formula for the Sphere packing bound.

Theorem 23 (Sphere packing bound)**.** *Let S be a set of sequences of length n and with minimum distance d. Then*

$$\sharp S \leq \begin{cases} \frac{q^n(q+1)}{q^{2\lfloor \frac{d-1}{2} \rfloor}+1} & \text{if } 2\lfloor \frac{d-1}{2} \rfloor \leq n-1, \\ \frac{q^n(q+1)}{1-q^{2(n-\lfloor \frac{d-1}{2} \rfloor)}+(1+q)q^n} & \text{if } 2\lfloor \frac{d-1}{2} \rfloor > n-1. \end{cases}$$

Proof. This is a direct consequence of Theorem 21 and using the fact that the union of the spheres of radius $\lfloor \frac{d-1}{2} \rfloor$ centered at the sequences in S is a disjoint union. □

4.5 Conclusion and future work

We have seen how the notion of weight of vectors can be extended to the notion of linear complexity of finite sequences. Using the new metric defined by the linear complexity, we developed a coding theory for finite sequences. We gave the Singleton bound and we presented a construction for an optimal set of sequences

reaching this bound. Then we computed an exact formula for the number of finite sequences which can be generated by an LFSR of a fixed order.

LFSR have been extensively studied [Rue86]. It is widely used in the generation of random secret key in symmetric cryptography. Our main goal however is to use the LFSR and linear complexity to get a new protocol for asymmetric public key cryptography.

In 1978, McEliece proposed a new cryptosystem using binary linear codes (Goppa codes) with the Hamming metric [McE78]. After 40 years of cryptanalysis, the cryptosystem is still considered to be structurally secure. However, the cryptosystem requires the use of public keys with large size. This makes it impractical for daily use. The advantage of using linear codes is that cryptosystem based on them are in general safe against quantum computers. Namely, there is no general algorithm which can decode a random linear code in polynomial time.

The strength of the McEliece cryptosystem is that Goppa codes look like random linear codes and it is considered to be a difficult problem to decode a random linear code. To solve the problem with the key size, it was suggested to use different family of linear codes. For instance, Niederreiter proposed a new cryptosystem using Reed-Solomon codes [Nie86]. However, cryptosystems using Reed-Solomon codes were proven to be insecure [SS92]. Several types of codes were suggested to get a secure cryptosystem. Another suggestion was that, instead of using the classical Hamming metric on the linear code, one can use the rank metric. For instance, a new cryptosystem based on the Gabidulin codes were proposed [GPT91]. This system was still proven to be insecure [Ove08].

Recently, this increased the interest in the search of linear codes with good properties which can be used in cryptography both in Hamming and rank metric. There is another cryptosystem which are also using a set and a metric on the set. The lattice based cryptosystem is the scheme where the metric is the Euclidean distance [Ajt96]. This particular cryptosystem is also resistant against attacks from quantum computers.

Motivated by all of this, we may think of a cryptosystem using the linear complexity as metric. We will see this in the next chapter. Finally we all know that Hamming metric codes are good for error correcting in a q-ary symmetric channel. For rank metric codes, they have good applications in network coding [KK08, SKK08]. It is our hope that the presented coding framework will also be of use for some particular channel.

Chapter 5

A new public key cryptosystem based on linear complexity of finite sequences

5.1 Introduction

The most popular public key encryption scheme used today are based on either the difficulty of factoring integers (RSA for example) or the difficulty of computing discrete logarithms in some group (elliptic curves for example). However, with the design of quantum computers, these schemes will soon be considered insecure by the use of Shor's algorithm [Sho94] when such computers are built. Fortunately, there are some public key encryption schemes which still resist the quantum computer attack. One alternative is the multivariate cryptosystems [DGS06]. Three other alternatives are the McEliece cryptosystem [McE78], the variants of GPT cryptosystem [GPT91] and the lattice based cryptosystem [Ajt96]. The multivariate cryptosystem is based on the fact that it is generally difficult to solve a non-linear system of polynomial equation in several variables. The common aspect of the other cryptosystems is that they use a module over some ring together with a metric. The security of the cryptosystem is based on the difficulty of some problem, mainly finding the closest vector with respect to the metric. For lattice based cryptosystem, they are using lattices in $\mathbb{R}^n$ and the metric is the Euclidean distance. For the McEliece cryptosystem, linear codes i.e. subspaces of $\mathbb{F}_q^n$ are used together with the Hamming metric. For the GPT cryptosystem, they use linear codes but the metric is the rank metric [Gab85].

In Chapter 4, we have introduced a new metric on the vector space $\mathbb{F}_q^n$ by using the linear complexity of finite sequences. In a paper [Ran18], we made a coding theory study on the vector space $\mathbb{F}_q^n$ with the linear complexity as metric.

There, it was already mentioned that using the linear complexity is somehow a generalization of the Hamming metric. In fact when considering periodic sequences with the fixed period, they are the same notion. However, in a more general setting, we have considered sequences without condition on the periods. From this, the idea of replacing the Hamming metric by the linear complexity to construct a McEliece-like cryptosystem comes up. Hence we present this work about a cryptosystem using linear complexity as metric. Linear feedback-shift registers have already some application in symmetric cryptography. A good reference for that is the book [Eli18]. However the application that we present here is completely new. First in Section 5.2, we will recall the notion of linear feedback-shift register, linear complexity and we will also restate some results from Chapter 4, which we will need in this Chapter. In Section 5.3, we will present a general construction for the cryptosystem. Then we will be more precise in Section 5.4. There, we will present a particular construction for the cryptosystem. In Section 5.5, we will show how secure the general scheme is by reducing the coset weight problem of [BMvT78] to the linear complexity coset weight problem. Finally, we will conclude in Section 5.6.

5.2 Linear-feedback shift register sequence and linear complexity

Many of the results in this section were taken from [Kle13] and Chapter 4 [Ran18]. Most of the statements here are given without proof. We refer the user to the previous chapter or the previously mentioned references if proofs are needed. Unless otherwise specified, we will always be working on a finite field $\mathbb{F}_q$ with q elements.

Recall from Definition 28 that, given a finite sequence $(a_i) = (a_1, \cdots, a_n)$, the linear complexity $\mathfrak{L}(a_i)$ of the sequence is the smallest integer l such that

$$a_{i+l} = \sum_{j=0}^{l-1} c_j a_{i+j}, \quad 0 \leq i \leq n-l-1,$$

for some fixed $c_j \in \mathbb{F}_q$. In case (a_i) is the zero sequence, then we define $\mathfrak{L}(a_i) = 0$.

As we have seen in Algorithm 3, Chapter 4, the Berlekamp-Massey algorithm can be used to find the shortest LFSR, i.e. the LFSR with the smallest order, generating a finite sequence. This can be done in $\mathcal{O}(n^2)$ field operations in $\mathbb{F}_q$, where n is the length of the sequence ([Kle13], Chapter 2, Section 4).

The key property of the linear complexity of sequences which will be used later is the following result from Theorem 18 of Chapter 4.

If (a_i) and (b_i) are two finite sequences and if $(c_i) = (a_i) + (b_i)$, then

$$\mathfrak{L}(c_i) \leq \mathfrak{L}(a_i) + \mathfrak{L}(b_i).$$

This is a triangular inequality. And as we have seen in the previous chapter, this triangular inequality leads to the construction of a metric.

Definition 34. Let $(a_i) = (a_1, \cdots, a_n) \in \mathbb{F}_q^n$ and $(b_i) = (b_1, \cdots, b_n) \in \mathbb{F}_q^n$ be two finite sequences of n elements of $\mathbb{F}_q$ each. Then we define a distance on $\mathbb{F}_q^n$ by the following,

$$\mathbf{d}((a_i), (b_i)) = \mathfrak{L}\left((a_i) - (b_i)\right),$$

where $\mathfrak{L}(\mathbf{0}) = 0$.

Like in classical coding theory, we can take a subset of $\mathbb{F}_q^n$ and then define the metric d on this set. For application in cryptography, we will be interested in subspaces of $\mathbb{F}_q^n$ only.

Definition 35. Let S be a subspace of $\mathbb{F}_q^n$. The minimum distance d of S is the minimum of $\mathbf{d}((a_i), (b_i))$ for any $(a_i), (b_i) \in S$ such that $(a_i) \neq (b_i)$. If the dimension of S as a subspace of $\mathbb{F}_q^n$ is equal to k, then we write $[n, k, d]$ to describe S.

The Singleton bound is given by the following theorem.

Theorem 24 ([Ran18]). *Let $\mathbb{F}_q$ be a finite field with q elements. Let $S \subset \mathbb{F}_q^n$ be an $[n, k, d]$-subspace of finite sequences. Then $k \leq n - d + 1$. If $k = n - d + 1$, then S is said to be an optimal set of sequences (OSS).*

Optimal sets of sequences can be constructed for any parameters q, n, k. To do this, let S be the subspace of $\mathbb{F}_q^n$ defined by

$$S = \{(0, \cdots, 0, a_1, \cdots, a_k) : a_i \in \mathbb{F}_q\}.$$

This is an optimal set of sequences.

Furthermore, we will also need the following property of finite sequences.

Theorem 25 ([Ran18]). *Let $\mathbb{F}_q$ be a finite field with q elements. Let $r \leq n$ be two non-negative integers. Let $B(n, r)$ be the set of all finite sequences (a_i) of length n over $\mathbb{F}_q$ such that $\mathfrak{L}(a_i) \leq r$ and let $b(n, r) = \sharp B(n, r)$. Then*

$$b(n, r) = \begin{cases} 1 & \text{if } r = 0, \\ \frac{q^{2r+1}+1}{q+1} & \text{if } r \leq \frac{n-1}{2}, \\ \frac{1-q^{2(n-r)}}{1+q} + q^n & \text{if } r \geq \frac{n}{2}. \end{cases}$$

As a consequence of Theorem 25, we have the following corollary.

Corollary 4 ([Ran18]). *For two positive integers $r \leq n$, the number of sequences of length n with linear complexity r over a finite field $\mathbb{F}_q$ of size q is*

$$\begin{cases} 1 & \text{if } r = 0, \\ q^{2r-1}(q-1) & \text{if } r \leq \left\lfloor \frac{n}{2} \right\rfloor, \\ q^{2(n-r)}(q-1) & \text{if } r > \left\lfloor \frac{n}{2} \right\rfloor, \end{cases}$$

Apart from the triangular inequality in Theorem 18, we also have a similar statement with respect to the multiplication between two sequences.

Definition 36. Let (a_i) and (b_i) be two sequences. We define the product $(a_i) * (b_i)$ as the sequence (c_i) such that $c_i = a_i b_i$, for all integer i.

Similar to Theorem 18, we have that the linear complexity of the product of two linear feedback shift register sequences is at most the product of the linear complexities of the two sequences. A proof of this fact can for example be found in [Kle13], Chapter 3, Section 4. Using this property we have the following theorem when we switch to sequences of finite length.

Theorem 26. *Let (a_i) and (b_i) be two sequences of finite length over a finite field $\mathbb{F}_q$. Then $\mathfrak{L}((a_i) * (b_i)) \leq \mathfrak{L}(a_i)\mathfrak{L}(b_i)$.*

We are now ready to move the construction of the cryptosystem.

5.3 A new cryptosystem based on LFSR

Similarly to the case of codes with the Hamming metric, we have the following problem on which our cryptosystem will be based.

Linear complexity coset weight problem: *Given a random subspace S of $\mathbb{F}_q^n$, a finite sequence $(b_i) \in \mathbb{F}_q^n$ and a positive integer w, find the sequence $(a_i) \in S$ such that $\mathbf{d}((a_i), (b_i)) \leq w$.*

If the set S has parameters $[n, k, d]$, then using the triangular inequality for d, it is easy to show that the solution of the linear complexity coset weight problem (if it exists) is unique when $w < \frac{d}{2}$.

We will later prove in Section 5.5 that for random subspace S, this problem is difficult to solve. However, there are some instances of S, where it is easy to solve this problem. Recall the following construction of optimal set of sequences S we have seen in Section 5.2.

$$S = \{(0, \cdots, 0, a_1, \cdots, a_k) : a_i \in \mathbb{F}_q\}.$$

Here we have that $d = n - k + 1$. So assume that $\mathfrak{L}(b_i) < \frac{d}{2}$. Suppose also that we know the sequence $(a_i) + (b_i)$ such that $(a_i) \in S$. We are therefore asked to recover (a_i) from $(a_i)+(b_i)$. However, by the property of S, we directly know the first $n-k$ elements of (b_i). And since, $\mathfrak{L}(b_i) < \frac{n-k+1}{2}$, we can use the Berlekamp-Massey algorithm to recover the linear feedback-shift register which generates (b_i). And the solution is unique with the property that $\mathfrak{L}(a_i) < \frac{n-k+1}{2}$. Using this linear feedback-shift register, we can recover the whole sequence (b_i). Once we recover (b_i), we can compute (a_i) at the end.

Let us now describe the cryptosystem. let S be a vector space of finite sequences over $\mathbb{F}_q$ with parameters $[n, k, d]$. Suppose that we are in possession of an algorithm for solving the linear complexity coset weight problem for S when $\mathfrak{L}(b_i) < \frac{d}{2}$. Then we set the following cryptosystem.

The public key is a basis of S as a subspace of $\mathbb{F}_q^n$. This is generally represented by a generator matrix $\mathbf{G}$, and an integer $t < \frac{d}{2}$. Notice that $\mathbf{G}$ is a $(k \times n)$-matrix of rank k over $\mathbb{F}_q$. The secret key is the algorithm for finding the closest sequence i.e. for solving the linear complexity coset problem for the particular sequence S.

The encryption is done as follow:

(i) The message $\mathbf{m} \in \mathbb{F}_q^k$ is encoded as an element $(a_i) = \mathbf{mG}$ of S.

(ii) The user chooses a random sequence $(b_i) \in \mathbb{F}_q^n$ such that $\mathfrak{L}(b_i) \leq t$.

(iii) Then $(c_i) = (a_i) + (b_i)$ is sent as ciphertext.

The decryption is just finding the element (a_i) of S which is closest to (c_i) using the secret algorithm. Since $\mathbf{G}$ is of rank k, and thus defines an injective map, we can invert it to recover $\mathbf{m}$ from (a_i).

Remark 13. The integer t is chosen in such a way that the decryption always gives a unique solution. This is similar to the unique decoding problem.

The security of the new cryptosystem we give is based on the hardness of solving the linear complexity coset weight problem. However we are facing the following challenge. First we need to know a set of linear sequences S with a given algorithm. Secondly, we also need to make sure that knowing S, it should not be possible for an attacker to find an efficient decoding algorithm. For instance a direct use of the optimal set of sequences we have seen previously is not possible since we know that the first elements of the ciphertext directly comes from the error introduced in the encryption and therefore we can decode it using the Berlekamp-Massey algorithm. So, what we need to do is to transform $\mathbf{G}$ to make it look like a random matrix. For that we give a modification of the previous cryptosystem.

The secret keys are an $[n, k, d]$ set of sequences S given by a generator matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ and an $(n \times n)$ invertible matrix $\mathbf{X}$ over $\mathbb{F}_q$. Here we can still take the previously constructed optimal set of sequence. However, we will disguise it by using the matrix $\mathbf{X}$. Hence the public keys are the product $\mathbf{GX}$ and an integer t.

The encryption is done as follow:

(i) The message $\mathbf{m} \in \mathbb{F}_q^k$ is encoded as $(a_i) = \mathbf{mGX}$.

(ii) Then the user chooses a random sequence (b_i) with $\mathfrak{L}(b_i) \leq t$.

(iii) then $(c_i) = (a_i) + (b_i)$ is sent as ciphertext.

For the decryption: first compute $(c_i)\mathbf{X}^{-1}$. Then we look for $\mathbf{mG}$ which is the closest sequence to $\mathbf{mG} + (b_i)\mathbf{X}^{-1}$.

However, in order to solve the linear complexity coset weight problem in the set S with input $\mathbf{mG} + (b_i)\mathbf{X}^{-1}$, we need to make sure that our decoding algorithm will work. Therefore, we also need that $\mathfrak{L}\left((b_i)\mathbf{X}^{-1}\right)$ is still smaller than $\frac{n-k+1}{2}$. Thus, we need to choose $\mathbf{X}$ to have some property. Furthermore, $\mathbf{X}$ should also be chosen so that knowing $\mathbf{GX}$, it should not be easy to recover $\mathbf{G}$ or a decoding algorithm.

5.4 A particular construction

As we have seen in the previous section, directly using finite sequences starting with many zeroes is not a good idea. However, to be able to decode, we still want to use these sequences. Let $n > k$ be two positive integers and let $\mathbb{F}_q$ be a field of size q. All objects will always have entries over $\mathbb{F}_q$.

Let $\mathbf{G}_1$ be a random invertible $(k \times k)$ matrix and $\mathbf{0}$ be the $(k \times (n-k))$ zero matrix. We also randomly generate a $(k \times (n-k))$ matrix $\mathbf{G}_2$. Using these matrices, we construct a matrix $\mathbf{G} = [\mathbf{0}|\mathbf{G}_1|\mathbf{G}_2]$.

Define S to be the vector space of finite sequences

$$(a_i) = (a_1, \cdots, a_n, \cdots, a_{2n-k}),$$

of length $2n-k$ generated by the rows of $\mathbf{G}$. We have that $a_i = 0$ for all $1 \leq i \leq n-k$.

We fix $t < \frac{n-k+1}{4}$. We define the $((2n-k) \times (2n-k))$ invertible matrix $\mathbf{M}$ such that

$$\mathbf{M}^{-1} = \begin{pmatrix} u_1 & 0 & \dots & 0 & m_{1,1} & \dots & m_{1,n-k} \\ u_2 & u_1 & \ddots & \ddots & m_{2,1} & \dots & m_{2,n-k} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \ddots & \vdots \\ \vdots & \ddots & u_2 & u_1 & m_{n,1} & \ddots & m_{n,n-k} \\ \vdots & \ddots & \ddots & u_2 & \vdots & \ddots & \vdots \\ u_{n-k+1} & \ddots & \ddots & \vdots & \vdots & \dots & \vdots \\ 0 & u_{n-k+1} & \ddots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & u_{n-k+1} & m_{2n-k,1} & \dots & m_{2n-k,n-k} \end{pmatrix} \tag{5.1}$$

We choose the u_i and the $m_{i,j}$ so that $\mathbf{M}^{-1}$ is invertible and hence $\mathbf{M}$ exists. We choose a sequence $(h_1, \cdots, h_{2n-k})$ with linear complexity exactly 2. And we defined the diagonal matrix $\mathbf{H} = \mathbf{diag}(h_1, \cdots, h_{2n-k})$. We also need $\mathbf{H}$ to be invertible and thus we choose $(h_1, \cdots, h_{2n-k})$ not to contain any zero element. This also implies that we have to work with field larger than $\mathbb{F}_2$. We now define $\mathbf{X}^{-1} = \mathbf{H}\mathbf{M}^{-1}$.

The secret key is now $\mathbf{G}$ and $\mathbf{X}^{-1}$. The public key is $\mathbf{G}' = \mathbf{G}\mathbf{X}$ and a positive integer t. We defined the rowspace of $\mathbf{G}'$ as S'. Encryption and decryption is done similarly as in the previous section:

(i) The message $\mathbf{m} \in \mathbb{F}_q^n$ is encoded as $(a_i) = \mathbf{m}\mathbf{G}' \in S'$.

(ii) Then the user chooses a random sequence (e_i) with $\mathfrak{L}(e_i) \leq t$

(iii) Then $(c_i) = (a_i) + (e_i)$ is sent as ciphertext.

Remark 14. Notice that we encrypt messages of length k into a ciphertext of length $2n - k$. The encryption rate is therefore $\frac{k}{2n-k}$.

The next step is to show that the decryption works i.e. we can solve the linear complexity coset weight problem uniquely in S'. To decrypt, first we compute $(c_i)\mathbf{X}^{-1}$. Thus we get $\mathbf{m}\mathbf{G} + (e_i)\mathbf{X}^{-1}$. We now need to solve the linear complexity coset weight problem in S to get $\mathbf{m}\mathbf{G}$ and finally recover $\mathbf{m}$ from this. So, what remains to show is that we can recover $\mathbf{m}\mathbf{G}$ uniquely.

Let us have a look at $(e_i)\mathbf{X}^{-1}$ with $\mathfrak{L}(e_i) \leq t$. We have $(e_i)\mathbf{X}^{-1} = (e_i)\mathbf{H}\mathbf{M}^{-1}$. So if we set $(b_i) = (e_i)\mathbf{H}$, then by Theorem 26, we have that $\mathfrak{L}(b_i) \leq 2\mathfrak{L}(e_i) \leq 2t$. Now we have $(e_i)\mathbf{X}^{-1} = (b_i)\mathbf{M}^{-1}$.

We denote the vector made with the n first elements of $(b_i)\mathbf{M}^{-1}$ by $\mathbf{v}$. We have

$$\mathbf{v}^T = \begin{pmatrix} b_1u_1 + b_2u_2 + \cdots + b_{n-k+1}u_{n-k+1} \\ b_2u_1 + b_3u_2 + \cdots + b_{n-k+2}u_{n-k+1} \\ \vdots \\ b_nu_1 + b_{n+1}u_2 + \cdots + b_{2n-k}u_{n-k+1} \end{pmatrix}$$

So $\mathbf{v}$ is the sum of the row of the following matrix.

$$\begin{pmatrix} b_1u_1 & b_2u_1 & \dots & b_nu_1 \\ b_2u_2 & b_3u_2 & \dots & b_{n+1}u_2 \\ \vdots & \vdots & \ddots & \vdots \\ b_{n-k+1}u_{n-k+1} & b_{n-k+2}u_{n-k+1} & \dots & b_{2n-k}u_{n-k+1} \end{pmatrix}$$

Notice that each one of the rows of the previous matrix can be generated by the same linear feedback-shift register of order $\mathfrak{L}(b_i)$. Therefore, $\mathbf{v}$ can also be generated by that same linear feedback-shift register of order $\mathfrak{L}(b_i)$, i.e. $\mathfrak{L}(\mathbf{v}) \leq \mathfrak{L}(b_i) \leq 2t$. Now, we have that the n first elements of $(c_i)\mathbf{X}^{-1}$ are

$$(c_i)' = \mathbf{m}[\mathbf{0}|\mathbf{G}_1] + \mathbf{v}.$$

Since $t < \frac{n-k+1}{4}$ and $\mathfrak{L}(\mathbf{v}) \leq 2t$, then $\mathfrak{L}(\mathbf{v}) < \frac{n-k+1}{2}$ so that we can use the decoding algorithm on the set of sequences generated by the rows of $[\mathbf{0}|\mathbf{G}_1]$ to get $\mathbf{m}[\mathbf{0}|\mathbf{G}_1]$. At the end, we use the inverse of $\mathbf{G}_1$ to recover $\mathbf{m}$.

Let us explain why we choose the above construction. Since we cannot choose $[\mathbf{0}|\mathbf{G}_1]$ as public key, we added more columns from $\mathbf{G}_2$. Then we scrambles all the columns together via $\mathbf{M}$ in such a way that $\mathbf{GM}$ looks random. Also, the choices of $\mathbf{M}$ was done in such a way that at the end, we are able to make a unique decoding on the set of sequences generated by $\mathbf{0}|\mathbf{G}_1]$.

The use of matrix $\mathbf{H}$ is also important. Namely, if we suppose that $\mathbf{H}$ is only the identity matrix, then,

$$\mathbf{G}' \begin{pmatrix} c_1 & 0 & \dots & 0 \\ c_2 & c_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & c_2 & c_1 \\ \vdots & \ddots & \ddots & c_2 \\ c_{n-k+1} & \ddots & \ddots & \vdots \\ 0 & c_{n-k+1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & c_{n-k+1} \end{pmatrix} = [\mathbf{0}|\mathbf{G}_1]. \tag{5.2}$$

Then we can just use this equation to find an alternative value for the u_i's. This will help in recovering an alternative secret key for the decryption. This is easy to solve as we have linear equations. However, as we will see in the next section, the use of $\mathbf{H}$ introduces non-linearity and thus helps to avoid the recovering of the secret key by simple linear algebra.

5.4.1 Cryptanalysis on the cryptosystem

Key recovery attack

One possible attack is given by the following. It consists in trying to recover a possible alternative for the secret key.

We suppose that the attacker knows $\mathbf{G}' = \mathbf{GX}$, where $\mathbf{X}^{-1} = \mathbf{HM}^{-1}$ such that $\mathbf{M}^{-1}$ has the form as in Equation (5.1), $\mathbf{H} = \mathbf{diag}(h_1, \cdots, h_{2n-k})$ and $\mathbf{G} = [\mathbf{0}|\mathbf{G}_1|\mathbf{G}_2]$.

Now, $\mathbf{X}^{-1}$ is of the form $\mathbf{HM}^{-1}$. The first possibility is to guess the value of $\mathbf{H}$. In case a correct value for $\mathbf{H}$ is produced, then the attacker is left with computing the solution of a system in the form of Equation (5.2). In order to avoid this attack, we have to choose the field $\mathbb{F}_q$ to be large enough so that making a correct guess for $\mathbf{H}$ is difficult. Notice that by Corollary 4, the number of sequences with linear complexity 2, is equal to $q^3(q-1) \simeq q^4$. Furthermore, guessing a correct value of $\mathbf{M}^{-1}$ is also difficult for large field.

The second possibility is that, the attacker may want to directly solve the equation

$$\mathbf{G}'\mathbf{HM}^{-1} = [\mathbf{0}|\mathbf{G}_1|\mathbf{G}_2]. \tag{5.3}$$

Here $\mathbf{G}'$ is the only known element in this equation. Furthermore, by considering the form of $\mathbf{H}$, which is made of a finite sequence of linear complexity 2, and

also the form of $\mathbf{M}^{-1}$, these two matrices introduce $2n^2 - 3nk + k^2 + n - k + 5$ variables. Even so, we did not yet consider the variables from $\mathbf{G}_1$ and $\mathbf{G}_2$. Here the contribution of $\mathbf{H}$ is just four variables, namely the initial terms and the coefficients of the linear feedback-shift register generating the h_i's.

Notice that the diagonal elements of $\mathbf{H}$ is of the form

$$(h_1, h_2, h_3, \cdots\cdots) =$$
$$(\mu_1, \mu_2, \lambda_1\mu_1 + \lambda_2\mu_2, \lambda_1\mu_2 + \lambda_2(\lambda_1\mu_1 + \lambda_2\mu_2), \cdots\cdots).$$

We can easily see that $\mathbf{G}\mathbf{X}^{-1}$ gives polynomial equations of degree $2n - k$ in the variables from the entries of $\mathbf{M}^{-1}$ and the linear feedback-shift register generating the h_i's. The high degree is indeed from the expression of the entries of $\mathbf{H}$. To conclude, the attacker has the task of solving a system of polynomials of large degree in many variables. Such problem is hard in general, even when the base field is $\mathbb{F}_2$ (See for example [GJ90], Appendix A7). However, we must choose proper values for the parameters n, k, q in order to avoid algorithms for computing some Gröbner basis which help in solving these systems of polynomial equations.

Brute force attack on the ciphertext

The following attack is on the general scheme. It consists of randomly guessing the error sequence (e_i) in the encryption. By Theorem 25, there are $b(2n - k, t) - 1$ possibilities for the errors with linear complexity r such that $1 \leq r \leq t$, where

$$b(2n-k,t) = \begin{cases} \frac{q^{2t+1}+1}{q+1} & \text{if } t \leq n - \frac{k}{2} - \frac{1}{2}, \\ \frac{1-q^{2(n-t)}}{1+q} + q^n & \text{if } t \geq n - \frac{k}{2}. \end{cases}$$

Since we have to choose $\mathfrak{L}(e_i) \leq t$, with $t < \frac{n-k+1}{4}$, then the number of possible error vectors that we can choose is

$$b(2n-k,t) = \frac{q^{2t+1}+1}{q+1} \simeq q^{2t}. \tag{5.4}$$

Therefore , about q^{2t} guesses are needed to decrypt the ciphertext. Of course this can be optimized so that only half guesses are needed. Notice that the linear complexity offers an advantage against the McEliece cryptosystem using Hamming metric. Namely, even if the linear complexity of an error vector is small, it is highly possible that all the entries of the error vector are non-zero i.e. its Hamming weight is very large. This limits the possibility of the use of the information set decoding attack like on the original McEliece cryptosystem. In the McEliece cryptosystem, the attackers only need to guess the position of the

error but do not need the values of the error. There, working with a large field does not really help against the information set decoding attack. In contrast to this, working with large fields increase the difficulty of guessing the error vector when we work with the metric defined by the linear complexity. Namely, the exact value of the error vector is needed when we make a guess.

We have the following example for parameters when taking into account the two previous attacks.

Example 4. We choose $q = 2^{32}, k = 16, n = 24, t = 2$. The encryption rate is $1/2$. The public key size is approximately 16 Kbits. Considering the key recovering attack and by using Corollary 4, there are $q^{2r-1}(q-1) \simeq q^{2r}$ possibilities for the choice of $\mathbf{H}$, where $r = 2$. This is approximately 2^{128}. For the second attack, the error vector which we introduce when encrypting has weight at most 2. Using Equation (5.4), there are approximately 2^{128} possibilities for the error vector. The security of the cryptosystem is about 128-bit. The key recovering attack is already difficult but also a brute force on the ciphertext needs about 2^{128} operations in $\mathbb{F}_{2^{32}}$.

The number of choices for the matrix $\mathbf{H}$ as well as the number of choices for the error vector mainly depends on the size of the field but not on the length of the vector. Therefore, this cryptosystem works better with large field. The cryptosystem has an advantage against the McEliece cryptosystems as we are using smaller key size. Namely, Bernstein *et al.* have the following parameters for the original McEliece cryptosystem using Goppa codes [BLP08]. For an 80-bit security, the public key size is 520 Kbits. For 128-bit security, the public key size is 1357 Kbits. These key sizes are large if we compare to our earlier example of parameters. We would like to point out that for the new cryptosystem, we have only considered the security of the cryptosystem against two attacks, which are mainly brute force attacks. the suggested parameters are just to emphasize the fact that the cryptosystem can help in reducing the public key size in code based cryptosystems. A reason for this reduction of key sizes is explained by the following

- For Hamming metric if we consider and error vector of weight t and length n, then we need to choose t non-zero elements in the field $\mathbb{F}_q$. Then we spread these elements across only t positions noting that $n - t$ positions still have the value 0.

- For the rank metric, if the error vector has weight t, then we also need to choose t non-zero elements of $\mathbb{F}_{q^m}$ which are linearly independent over $\mathbb{F}_q$. The errors will be spread across all the n positions but using linear combinations over the smaller field $\mathbb{F}_q$ only.

- For the linear complexity metric, suppose also that the error vector has weight t. We need to choose t non-zero elements of a field $\mathbb{F}_q$. However, via the recurrence formula of a linear feedback-shift register, we spread the errors to all positions but also the linear combinations are done within the field $\mathbb{F}_q$ itself. In contrast to the rank metric setting we do not move to a smaller field to choose the linear combination and thus we have more choice for the coefficients of the linear combination.

More cryptanalysis should be done in order to find the right parameters for the cryptosystem. For instance, we need to have a look at a Gröbner basis attack for recovering alternative secret keys. Considering this attack we may have to further increase the paramaters and thus we get a larger public key size. Still, I do not expect the public key size to be as large as suggested in [BLP08] for the McEliece cryptosystem.

5.5 Linear complexity coset weight problem

Recall the following problem which we have already stated in a previous section.

Linear complexity coset weight problem: *Given a random subspace S of $\mathbb{F}_q^n$, a finite sequence $(b_i) \in \mathbb{F}_q^n$ and a positive integer w, find the sequence $(a_i) \in S$ such that $\mathbf{d}((a_i),(b_i)) \leq w$.*

To show the difficulty of solving this problem, we will show that the coset weight problem which we state below can be reduced to the linear complexity coset weight problem.

Coset weight problem: *Let w be a positive integer. Let $\mathcal{C}$ be a random linear code over a finite field $\mathbb{F}_q$ together with the Hamming distance $\mathbf{d}_H$. Given a vector $\mathbf{x} \in \mathbb{F}_q^n$, find the codeword $\mathbf{c} \in \mathcal{C}$ such that $\mathbf{d}_H(\mathbf{c},\mathbf{x}) \leq w$.*

To do this let us recall the following notion from complexity theory.

Definition 37. A problem $\mathcal{P}$ is said to be in the class NP if given a solution to $\mathcal{P}$, there is an algorithm which can verify the solution in polynomial time.

Here, we gave an equivalent definition of NP. Usually, NP problems are defined to be the problems which can be solved in polynomial time using a non-deterministic Turing machine.

Definition 38. A problem $\mathcal{P}$ is called NP-hard if any problem in NP can be reduced to $\mathcal{P}$ in polynomial time.

Definition 39. If a problem is both in NP and NP-hard, then it is called NP-complete.

NP-complete problems are considered to be intractable. One example of an NP-complete problem is the coset weight problem as it was proven in [BMvT78].

Theorem 27 ([BMvT78]). *The coset weight problem is NP-complete.*

We will use this result to show that the linear complexity coset weight problem is also NP-complete. First of all, it is easy to see that the problem is in NP. To show the NP-hardness, we will need to translate the notion of Hamming distance into the notion of linear complexity. For that we need the following theorem of König-Rados which we have already seen in Chapter 4.

Theorem 28 (König-Rados, [LN96]). *Let $f(x) = f_0 + f_1 x + \cdots + f_{Q-2}x^{Q-2}$ be a polynomial over a finite field $\mathbb{F}_Q$ of size Q. Then the number of roots of of $f(x)$ in $\mathbb{F}_Q^*$ is given by $Q - 1 - r$, where r is the rank of the matrix*

$$\mathbf{A} = \begin{pmatrix} f_0 & f_1 & \cdots & f_{Q-2} \\ f_1 & \ddots & \ddots & f_0 \\ \vdots & \ddots & \ddots & \vdots \\ f_{Q-2} & f_0 & \cdots & f_{Q-3} \end{pmatrix}.$$

Notice that if r is the rank of $\mathbf{A}$ in the previous theorem, then for $i > r$, the i-th row of $\mathbf{A}$ is a linear combination of the r first rows of $\mathbf{A}$. Therefore, the $(r+1)$-th row of $\mathbf{A}$ is a linear combination of the r first rows. Thus, $(f_0, f_1, \ldots, f_{Q-2})$ can be generated by an LFSR of order r and it is $(Q-1)$-periodic. So we can say that $(f_0, f_1, \ldots, f_{Q-2}, f_0, f_1, \ldots, f_{Q-2})$ has linear complexity $r \leq Q - 1$. We claim that it cannot be generated by a shorter LFSR. Otherwise, by the periodicity, one can show that $\mathbf{A}$ has rank smaller than r. We thus have shown that the linear complexity of $(f_0, f_1, \ldots, f_{Q-2}, f_0, f_1, \ldots, f_{Q-2})$ is equal to r.

Now, let us see how we can convert a linear code into a subspace of sequences. Suppose that we have a finite field $\mathbb{F}_q$ with q elements. Assume that $\mathcal{C} \subset \mathbb{F}_q^n$. Let $l = \left\lceil \log_q(n) \right\rceil$ and suppose that $Q = q^l$. For an element $(c_1, \cdots, c_n) \in \mathbb{F}_q^n$, we extend it to $\mathbf{c} = (c_1, \cdots, c_n, 0 \cdots, 0) \in \mathbb{F}_q^{Q-1}$. If we suppose that $\{a_1, \cdots, a_{Q-1}\} = \mathbb{F}_Q^*$, then any $\mathbf{c}$ can be written as

$$\mathbf{c} = (f(a_1), \cdots, f(a_{Q-1})),$$

for some polynomial $f(x)$ of degree $Q-2$ over $\mathbb{F}_Q$. Now, the Hamming weight of $(c_1, \cdots, c_n)$ is equal to the Hamming weight of $\mathbf{c}$. Using Theorem 28 and the above discussion, we see that the Hamming weight of $\mathbf{c}$ is the same as the linear complexity of $(f_0, f_1, \ldots, f_{Q-2}, f_0, f_1, \ldots, f_{Q-2})$. Therefore we have the following correspondence.

$$\left\{ \begin{array}{c} \text{Linear code in } \mathbb{F}_q^n \text{ of dimension } k \\ \text{metric using the Hamming weight} \end{array} \right\}$$
$$\Updownarrow \tag{5.5}$$
$$\left\{ \begin{array}{c} \text{Vector space of } (Q-1)\text{-periodic sequences in } \mathbb{F}_Q^{2(Q-1)} \text{ of dimension } k \\ \text{metric using the Linear complexity} \\ \text{Linear complexity of the sequence is } Q-1 \text{ at most} \end{array} \right\}$$

In the Correspondence (5.5), if a sequence (a_i) corresponds to a codeword $\mathbf{c}$, the linear complexity of (a_i) is equal to the Hamming weight of $\mathbf{c}$. The minimum distance of these sets are the same.

Using the Correspondence (5.5), we now translate the coset weight problem into an instance of the linear complexity coset weight problem. In other words, a problem of finding the closest codewords in a linear codes will be translated into a problem of find the closest sequence with respect to the linear complexity metric. The setting for the linear code is the following.

Problem 1

- $\mathcal{C}$ is a linear code of dimension k in $\mathbb{F}_q^n$.
- $\mathbf{x}'$ is a vector in $\mathbb{F}_q^n$.
- w is a positive integer.
- Find $\mathbf{c}' \in \mathcal{C}$ such that the Hamming weight of $\mathbf{x}' - \mathbf{c}'$ is at most w.

The problem is transformed into a problem of finding the closest sequence as follows.

Problem 2

- S a subspace of dimension k of sequences in $\mathbb{F}_Q^{2(Q-1)}$ of the form

  $$(f_0, \cdots, f_{Q-2}, f_0, \cdots, f_{Q-2}),$$

  and linear complexity are at most $Q-1$.
- $\mathbf{x}$ a vector of the form $(x_0, \cdots, x_{Q-2}, x_0, \cdots, x_{Q-2}) \in \mathbb{F}_Q^{2(Q-1)}$ with linear complexity at most $Q-1$.
- w is a positive integer.

- Find $\mathbf{c} \in S$ such that the linear complexity of $\mathbf{x} - \mathbf{c}$ is at most w with $w \leq Q - 1$.

Now if it is easy to solve the linear complexity coset weight problem in general with some known algorithm, then we can use that algorithm to solve Problem 2. Since the solution $\mathbf{x}$ is such that $\mathbf{x} - \mathbf{c}$ has linear complexity $Q - 1$ at most and that $\mathbf{x} - \mathbf{c}$ must also have the form $(f_0, \cdots, f_{Q-2}, f_0, \cdots, f_{Q-2}) \in \mathbb{F}_Q^{2(Q-1)}$, then the algorithm must output a solution for Problem 2. But this means that we are able to solve Problem 1. Therefore, it is also easy to solve the coset weight problem.

Remark 15.

(i) The condition for the linear complexity to be $Q - 1$ at most and that the sequences are of the form $(x_0, \cdots, x_{Q-2}, x_0, \cdots, x_{Q-2})$ are important. Namely this ensure that the LFSR generating $\mathbf{x} - \mathbf{c}$ is $Q - 1$ periodic so that a solution to Problem 2 is a solution to Problem 1.

(ii) Switching from the field $\mathbb{F}_q$ of size q to the field $\mathbb{F}_Q$ with $Q = q^{\lceil \log_q(n) \rceil}$ does not increase the difficulty of the problem exponentially.

In other words, solving a coset weight problem over $\mathbb{F}_q^n$, with $\sharp\mathbb{F}_q = q$, can be reduced to solving a linear complexity coset weight problem over $\mathbb{F}_Q^{2(Q-1)}$. Thus, we have the following theorem.

Theorem 29. *Solving the linear complexity coset weight problem is at least as hard as solving the coset weight problem.*

Since solving a general coset weight problem problem is suggested to be a hard problem [BMvT78], we can also conclude that solving the linear complexity coset weight problem is a hard problem.

Theorem 30.
Solving the linear complexity coset weight problem is NP-complete.

5.6 Conclusion

In this work, we have devised a new general cryptosystem based on linear complexity. We have provided a basis of why the general cryptosystem is secure by showing that the problem on which it is based on is a difficult problem. Namely, our new cryptosystem is based on the difficulty of solving the linear complexity coset weight problem. And we have shown that the linear complexity coset

weight problem is at least as hard as the coset weight problem. This later problem is the problem on which the McEliece cryptosystem is based on. We have also provided a construction for the new cryptosystem. We have suggested some parameters based on some attacks which are mainly brute force attack. These attacks consist of recovering the secret key, or randomly guessing the error introduced in the encryption. As a future work, we will look into this construction more closely. We want to investigate algebraic attacks on the cryptosystem.

Chapter 6

Conclusion

Apart from their application in network coding, rank metric codes have found applications in cryptography. Using the fact that it is difficult to decode a random linear code, a cryptosystem was devised by McEliece where messages are encoded via a specific linear code and encryption is done by adding some noise to the codeword. The disadvantage of the original scheme by McEliece is that it requires the use of large public keys. In order to avoid this, the use of classes of optimal codes were suggested. These classes are called maximum rank distance codes in the rank metric setting. Among them, there are the Gabidulin codes. Unfortunately, cryptosystem based on Gabidulin codes were shown to be insecure and therefore we need to find alternative codes.

The first result we have given was that there are indeed many linear codes which have a maximum rank distance over large fields. We have given a theoretical as well as a probabilistic proof for this existence. Furthermore, most of the maximum rank distance codes are not Gabidulin codes. In terms of probability we have the following results:

Let $\mathbb{F}_{q^m}/\mathbb{F}_q$ be a field extension of finite degree. Given a randomly generated matrix $\mathbf{X} \in \mathbb{F}_{q^m}^{k\times(n-k)}$, the probability that the matrix $[\mathbf{I}_k|\mathbf{X}]$ generates a maximum rank distance code of length n and dimension k over the extension $\mathbb{F}_{q^m}/\mathbb{F}_q$ is,

$$\Pr([\mathbf{I}_k|\mathbf{X}] \text{ generates an MRD code }) \geq 1 - \sum_{i=0}^{k} i \begin{bmatrix} k \\ k-i \end{bmatrix}_q \begin{bmatrix} n-k \\ i \end{bmatrix}_q q^{i^2} q^{-m}.$$

When m goes to infinity, then the right-hand side of the inequality goes to 1. We can translate this result to the following statement: When the field $\mathbb{F}_{q^m}$ is large enough, then with high probability, a randomly generated code is a maximum rank distance code.

For the question of generating Gabidulin codes, we got the following result.

Let $\mathbf{X} \in \mathbb{F}_{q^m}^{k\times(n-k)}$ be randomly chosen. Then the probability that $[\mathbf{I}_k|\mathbf{X}]$ generates a generalized Gabidulin code of length n and dimension k over the extension $\mathbb{F}_{q^m}/\mathbb{F}_q$ is,

$$\Pr([\mathbf{I}_k|\mathbf{X}] \text{ generates a gen. Gabidulin code}) \leq \phi(m)q^{-(m-1)(n-k-1)(k-1)}.$$

When m goes to infinity, then the right hand side goes to 0 and therefore, when the field is large enough, it is unlikely that we get a Gabidulin code. A construction of twisted Gabidulin codes by Sheekey gives a new class which contains codes which are not equivalent to any Gabidulin codes. To further confirm this existence theorem, we also have given a construction of a new class of rank metric codes via the use of derivation on polynomial rings. This construction produces codes which are different from (twisted) Gabidulin codes.

For a code to be useful in cryptography, a decoding algorithm is needed. For twisted Gabidulin codes, there were no decoding algorithm until we started this work. In this thesis, we gave two different decoding algorithms for twisted Gabidulin codes. For the second algorithm, it is different from any existing algorithm when we apply it to the classical Gabidulin codes. This algorithm uses the notion of linear complexity for some general linear feedback-shift register and it allows us to use the Berlekamp-Massey algorithm in a different way. From this, we notice that there is a connection between the notion of linear complexity of sequences and the notion of weight of codewords.

We thus investigated this relation between the weight of codewords and the linear complexity of sequences, but we switched to the setting of Hamming metric codes. This equivalence is described as follows.

Let $\mathbb{F}_q = \{0, \alpha_1, \cdots, \alpha_{q-1}\}$ be a finite field with q elements. Let $f(x) = a_0 + a_1x + \cdots + a_{q-2}x^{q-2}$ be a polynomial over $\mathbb{F}_q$. Then the Hamming weight of $\overline{(f(\alpha_1), \cdots, f(\alpha_{q-1}))}$ is equal to the linear complexity of the periodic sequence $\overline{(a_0, \cdots, a_{q-2})}$.

This equivalence holds with the setting that the sequences, we consider, have a fixed period $q-1$. This leads us to think of the setting where we do not restrict the sequences to have a fixed period. It turns out that in this case, the linear complexity defines a metric on the vector space $\mathbb{F}_q^n$. Therefore, we can make a new coding theory by using this new metric. We have developed the theory which are necessary for application in cryptography. We have for instance given a proof that the number $b(n, r)$ of sequences of length n over $\mathbb{F}_q$ with linear complexity at most r is

(i) if $r = 0$, $b(n, 0) = 1$,

(ii) if $r + 1 \leq n - r$,

$$b(n, r) = \frac{q^{2r+1} + 1}{q + 1},$$

(iii) if $n - r \leq r$,

$$b(n, r) = \frac{1 - q^{2(n-r)}}{1 + q} + q^n.$$

Finally, we use these results to construct a new cryptosystem. We have given the general construction and we have proved that decoding random linear codes with respect to the linear complexity is also a difficult problem. This forms the basis for the security of the new cryptosystem. We have given a particular instance of the cryptosystem and we have given some parameters suggestion. One of the primary goal of the thesis was to show that using the linear complexity as a metric helps in reducing the public key sizes for the cryptosystem.

Future work

The cryptosystem based on the linear complexity of sequences is completely new. A lot of work still has to be done before the cryptosystem is ready for practical use. Our goal in the thesis was to provide the foundation of the theory. We have suggested a particular construction. However, we mainly considered the brute force attack against the cryptosystem. We have also seen that some attack can be reduced to solving a system of multivariate polynomial equations. There are many algorithms which can be used to try to solve these problems. In general these algorithms do not always work as each specific system of equations has its own way of solving it. To this regard, we will need to analyze algorithms which can be applied to recover the secret key in our cryptosystem. We need to investigate the speed-up provided by such attack in order to find the right parameters for our cryptosystem. Therefore, as a future work, we want to analyze our cryptosystem against these attacks on solving a system of multivariate polynomial equations.

Bibliography

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 99–108. ACM, 1996.

[BBC+16] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, and D. Schipani. Enhanced Public Key Security for the McEliece Cryptosystem. *Journal of Cryptology*, 29(1):1–27, Jan 2016.

[BBD08] D.J. Bernstein, J. Buchmann, and E. Dahmen. *Post Quantum Cryptography*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[BC07] M. Baldi and F. Chiaraluce. Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC Codes. In *2007 IEEE International Symposium on Information Theory*, pages 2591–2595, June 2007.

[BCGO09] T.P. Berger, P-L. Cayrel, P. Gaborit, and A. Otmani. Reducing Key Length of the McEliece Cryptosystem. In Bart Preneel, editor, *Progress in Cryptology – AFRICACRYPT 2009*, pages 77–97, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[Ber73] E. Berlekamp. Goppa codes. *IEEE Transactions on Information Theory*, 19(5):590–592, September 1973.

[BGLK+17] J. Bolkema, H. Gluesing-Luerssen, C.A. Kelley, K.E. Lauter, B. Malmskog, and J. Rosenthal. Variations of the mceliece cryptosystem. In E.W. Howe, K.E. Lauter, and J.L. Walker, editors, *Algebraic Geometry for Coding Theory and Cryptography*, pages 129–150, Cham, 2017. Springer International Publishing.

[Bla79] R. E. Blahut. Transform techniques for error control codes. *IBM J. Res. Dev.*, 23(3):299–315, May 1979.

[BLP08] D.J. Bernstein, T. Lange, and C. Peters. Attacking and Defending the McEliece Cryptosystem. In Johannes Buchmann and Jintai Ding, editors, *Post-Quantum Cryptography*, pages 31–46, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[Blu04] A.W. Bluher. On $x^{q+1} + ax + b$. *Finite fields and their applications*, 10(3):285 – 305, 2004.

[BMvT78] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, May 1978.

[CFS01] N.T. Courtois, M. Finiasz, and N. Sendrier. How to Achieve a McEliece-Based Digital Signature Scheme. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 157–174, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[CGGU+14] A. Couvreur, P. Gaborit, V. Gauthier-Umaña, A. Otmani, and J-P. Tillich. Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes. *Designs, Codes and Cryptography*, 73(2):641–666, Nov 2014.

[CK12] M.T. Comer and E.L Kaltofen. On the Berlekamp/Massey algorithm and counting singular Hankel matrices over a finite field. *Journal of Symbolic Computation*, 47(4):480 – 491, 2012. Special Issue for Joachim von zur Gathen at 60.

[CMCP14] A. Couvreur, I. Márquez-Corbella, and R. Pellikaan. A polynomial time attack against algebraic geometry code based public key cryptosystems. In *2014 IEEE International Symposium on Information Theory*, pages 1446–1450, June 2014.

[CMP16] A. Cossidente, G. Marino, and F. Pavese. Non-linear maximum rank distance codes. *Designs, Codes and Cryptography*, 79(3):597–609, Jun 2016.

[Day60] D. E. Daykin. Distribution of bordered persymmetric matrices in a finite field. *Journal für die reine und angewandte Mathematik*, 203:47–54, 1960.

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

[DGS06] J. Ding, J.E. Gower, and D. Schmidt. *Multivariate Public Key Cryptosystems (Advances in Information Security)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[dlCKWW16] J. de la Cruz, M. Kiermaier, A. Wassermann, and W Willems. Algebraic structures of MRD codes. *Advances in Mathematics of Communications*, 10:499, 2016.

[Eli18] M. Elia. *An Introduction to Classic Cryptography. With an Exposition of the Mathematics of Private and Public Key Ciphers*. Crittografia book series. Aracne, 2018.

[FGUaO+11] J.C. Faugère, V. Gauthier-Umanã, A. Otmani, L. Perret, and J.P. Tillich. A distinguisher for high rate McEliece cryptosystems. In *2011 IEEE Information Theory Workshop*, pages 282–286, Oct 2011.

[Gab85] E.M. Gabidulin. Theory of codes with maximum rank distance. *Problems of Information Transmission*, 21:1–12, 1985.

[Gab08] Ernst M. Gabidulin. Attacks and counter-attacks on the GPT public key cryptosystem. *Designs, Codes and Cryptography*, 48(2):171–177, Aug 2008.

[Gib95] J. K. Gibson. Severely denting the Gabidulin version of the McEliece Public Key Cryptosystem. *Designs, Codes and Cryptography*, 6(1):37–45, Jul 1995.

[Gie98] M. Giesbrecht. Factoring in skew-polynomial rings over finite fields. *Journal of Symbolic Computation*, 26(4):463 – 486, 1998.

[GJ90] M.R. Garey and D.S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[GJS16] Q. Guo, T. Johansson, and P. Stankovski. A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, pages 789–815, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[Gop70] V.D. Goppa. A new class of linear correcting codes. *Problemy Peredači Informacii*, 6(3):24–30, 1970.

[GPT91] E.M. Gabidulin, A.V. Paramonov, and O.V. Tretjakov. Ideals over a non-commutative ring and their application in cryptology. In Donald W. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, pages 482–489, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.

[GRS16] P. Gaborit, O. Ruatta, and J. Schrek. On the complexity of the rank syndrome decoding problem. *IEEE Transactions on Information Theory*, 62(2):1006–1019, Feb 2016.

[GRSZ14] P. Gaborit, O. Ruatta, J. Schrek, and G. Zémor. New results for rank-based cryptography. In David Pointcheval and Damien Vergnaud, editors, *Progress in Cryptology – AFRICACRYPT 2014*, pages 1–12, Cham, 2014. Springer International Publishing.

[HTM17] A-L. Horlemann-Trautmann and K. Marshall. New criteria for MRD and Gabidulin codes and some Rank-Metric code constructions. *Advances in Mathematics of Communications*, 11:533, 2017.

[HTMR18] A-L. Horlemann-Trautmann, K. Marshall, and J. Rosenthal. Extension of Overbeck's attack for Gabidulin-based cryptosystems. *Designs, Codes and Cryptography*, 86(2):319–340, Feb 2018.

[JM96] H. Janwa and O. Moreno. McEliece public key cryptosystems using algebraic-geometric codes. *Designs, Codes and Cryptography*, 8(3):293–307, Jun 1996.

[KG05] A. Kshevetskiy and E. Gabidulin. The new construction of rank codes. In *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, pages 2105–2108, Sept 2005.

[KK08] R. Koetter and F.R. Kschischang. Coding for errors and erasures in random network coding. *IEEE Transactions on Information Theory*, 54(8):3579–3591, Aug 2008.

[Kle13] Andreas Klein. *Stream Ciphers*. Springer Publishing Company, Inc., 2013.

[Ksh07] A. Kshevetskiy. Security of GPT-like public-key cryptosystems based on linear rank codes. In *2007 3rd International Workshop on Signal Design and Its Applications in Communications*, pages 143–147, Sept 2007.

[LDW94] Y.X. Li, R. H. Deng, and X.M. Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–273, Jan 1994.

[Lef12] S. Lefschetz. *Algebraic Geometry*. Dover Books on Mathematics. Dover Publications, 2012.

[LN96] R. Lidl and H. Niederreiter. *Finite Fields*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2 edition, 1996.

[Loi06] P. Loidreau. A Welch–Berlekamp Like Algorithm for Decoding Gabidulin Codes. In Øyvind Ytrehus, editor, *Coding and Cryptography*, pages 36–45, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[Loi08] P. Loidreau. Properties of codes in rank metric. In *Eleventh International Workshop on Algebraic and Combinatorial Coding Theory ACCT2008*, Pamporovo, Bulgaria, June 2008.

[LQ12] S. Ling and L. Qu. A note on linearized polynomials and the dimension of their kernels. *Finite Fields and Their Applications*, 18(1):56 – 62, 2012.

[LSS14] W. Li, V. Sidorenko, and D. Silva. On transform-domain error and erasure correction by Gabidulin codes. *Designs, Codes and Cryptography*, 73(2):571–586, Nov 2014.

[MB09] R. Misoczki and P.S.L.M. Barreto. Compact McEliece Keys from Goppa Codes. In M.J. Jacobson, V. Rijmen, and R. Safavi-Naini, editors, *Selected Areas in Cryptography*, pages 376–392, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[McE78] R.J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44:114–116, January 1978.

[MM09] M.A.H. MacCallum and A.V. Mikhailov. *Algebraic Theory of Differential Equations*. London Mathematical Society Lecture Note Series. Cambridge University Press, 2009.

[Mor14] K. Morrison. Equivalence for Rank-Metric and Matrix Codes and Automorphism Groups of Gabidulin Codes. *IEEE Transactions on Information Theory*, 60(11):7035–7046, Nov 2014.

[MRS00] C. Monico, J. Rosenthal, and A. Shokrollahi. Using low density parity check codes in the McEliece cryptosystem. In *2000 IEEE International Symposium on Information Theory (Cat. No.00CH37060)*, pages 215–, 2000.

[MS78] F.J. McWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North-holland Publishing Company, 2nd edition, 1978.

[NHTRR18] Alessandro Neri, Anna-Lena Horlemann-Trautmann, Tovohery Randrianarisoa, and Joachim Rosenthal. On the genericity of maximum rank distance and Gabidulin codes. *Designs, Codes and Cryptography*, 86(2):341–363, Feb 2018.

[Nie86] H Niederreiter. Knapsack type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory. Problemy Upravlenija i Teorii Informacii*, 15:19–34, 1986.

[Ore33] Oystein Ore. Theory of non-commutative polynomials. *Annals of Mathematics*, 34(3):480–508, 1933.

[OTD10] A. Otmani, J-P. Tillich, and L. Dallot. Cryptanalysis of Two McEliece Cryptosystems Based on Quasi-Cyclic Codes. *Mathematics in Computer Science*, 3(2):129–140, Apr 2010.

[Ove08] R. Overbeck. Structural Attacks for Public Key Cryptosystems based on Gabidulin Codes. *Journal of Cryptology*, 21(2):280–301, Apr 2008.

[Per12] Edoardo Persichetti. Compact McEliece keys based on quasi-dyadic Srivastava codes. *Journal of Mathematical Cryptology*, 6(2):149–169, October 2012.

[PT91] A.V. Paramonov and O.V. Tretjakov. An Analogue of Berlekamp-Massey algorithm for decoding codes in rank metric. *Proc. of Moscow Institute for Physics and Technology (MIPT)*, 1991. In Russian.

[Ran17] T. Randrianarisoa. A Decoding Algorithm for Rank Metric Codes. *ArXiv e-prints*, December 2017, 1712.07060.

[Ran18] T. Randrianarisoa. Coding Theory using Linear Complexity of Finite Sequences. *ArXiv e-prints*, February 2018, 1802.10034.

[RGH10] H. Rashwan, E.M. Gabidulin, and B. Honary. A Smart approach for GPT cryptosystem based on rank codes. In *2010 IEEE International Symposium on Information Theory*, pages 2463–2467, June 2010.

[RP04] G. Richter and S. Plass. Error and erasure decoding of rank-codes with a modified Berlekamp-Massey algorithm. In *5th International ITG Conference on Source and Channel Coding*, pages 249–256, 2004.

[RR17] J. Rosenthal and T. Randrianarisoa. A decoding algorithm for twisted Gabidulin codes. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2771–2774, June 2017.

[Rue86] R. Rueppel. *Analysis and Design of Stream Ciphers*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1986.

[S+03] Neil J.A. Sloane et al. The on-line encyclopedia of integer sequences, 2003.

[Sch80] J.T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.

[She16] John Sheekey. A new family of linear maximum rank distance codes. *Advances in Mathematics of Communications*, 10(3):475–488, 2016.

[Sho94] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, 1994 Proceedings.*, pages 124–134, 1994.

[SKK08] D. Silva, F.R. Kschischang, and R. Koetter. A rank-metric approach to error control in random network coding. *IEEE Transactions on Information Theory*, 54(9):3951–3967, Sept 2008.

[SS92] M. Sidelnikov and O. Shestakov. On insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 2(4):439, 1992.

[Var97] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, Nov 1997.

[VGM11] Sidorenko V., Richter G., and Bossert M. Linearized shift-register synthesis. *IEEE Transactions on Information Theory*, 57(9):6025–6032, Sept 2011.

[Wie10] C. Wieschebrink. Cryptanalysis of the Niederreiter Public Key Scheme Based on GRS Subcodes. In Nicolas Sendrier, editor, *Post-Quantum Cryptography*, pages 61–72, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.