Topics in Code-Based Cryptography

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Bhagyalekshmy S

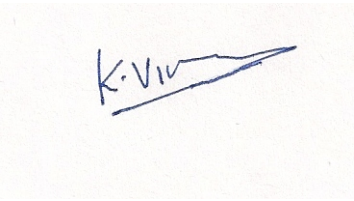
Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

April, 2025

Supervisor: Dr. Krishna Kaipa
Co-supervisor: Dr. Rutuja Kshirsagar

Certificate

This is to certify that this dissertation entitled Topics in Code-Based Cryptography towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Bhagyalekshmy S at Indian Institute of Science Education and Research under the supervision of Dr. Krishna Kaipa, Associate Professor, Department of Mathematics, and Dr. Rutuja Kshirsagar, Fujitsu Research of America, Inc, during the academic year 2024-2025.



Dr. Krishna Kaipa

Dr. Rutuja Kshirsagar

Dr. Vivek Mohan Mallick

Committee:

Dr. Krishna Kaipa

Dr. Rutuja Kshirsagar

Dr. Vivek Mallick

This thesis is dedicated to all my well-wishers.

Declaration

I hereby declare that the matter embodied in the report entitled Topics in Code-Based Cryptography are the results of the work carried out by me at the Department of Mathematics, Indian Institute of Science Education and Research, Pune, under the supervision of Dr. Krishna Kaipa and Dr. Rutuja Kshirsagar , and the same has not been submitted elsewhere for any other degree.

Bhagyalekshmy S

Acknowledgements

I extend my deepest gratitude to my supervisor, Dr. Krishna Kaipa, for allowing me to pursue this project. I am equally thankful to my co-supervisor, Dr. Rutuja Kshirsagar, for her invaluable guidance throughout the project. Her deep commitment and her meticulous attention to detail have remarkably shaped this thesis. I would also like to acknowledge my expert, Dr. Vivek Mallick.

I humbly express my gratitude to God for his countless blessings. I'm really grateful to my parents for encouraging and supporting me to pursue this scientific career. I extend my heartfelt gratitude to Nandagopal for his constant support and encouragement. I also acknowledge the Mathematics department for all the resources and facilities.

Abstract

Given the rapid advancements in the field of quantum computing, the need for a robust post-quantum cryptosystem has become increasingly urgent. The McEliece cryptosystem, or equivalently the Niederreiter cryptosystem, is a promising quantum-resistant alternative that facilitates faster encryption and decryption processes. Both of these cryptosystems rely on a type of code called the Goppa code. However, a significant challenge with these cryptosystems is the large size of the keys − both private and public. These keys are typically represented using large matrices, which complicates the implementation and deployment of these cryptosystems.

To address this issue, numerous efforts have been made to replace Goppa codes with codes that reduce the key size while maintaining security. One such modification, discussed in [15] and [16], involves substituting the Goppa codes in the McEliece and Niederreiter cryptosystems with quasi-cyclic codes. This adaptation results in a variant of the cryptosystem that is not only quantum secure but also offers enhanced performance in terms of transmission and encryption rates, along with significantly smaller key sizes. In this work, we develop one such cryptosystem based on quasi-twisted codes.

To successfully develop a secure and efficient code-based cryptosystem, it is essential to have an effective decoding algorithm for the underlying code. In the context of quasi-twisted (QT) codes, one of the primary challenges is ensuring the existence of such a decoding algorithm. To achieve this, we propose a syndrome-based decoding approach, which efficiently corrects errors up to the Hartmann-Tzeng (HT)-like bound, which defines the maximum number of errors that can be reliably corrected without compromising the structure of the code.

Quantum Fourier Sampling (QFS) plays a central role in many quantum algorithms, including Shor's algorithm, and proving that a cryptosystem can withstand QFS is considered a critical measure of its quantum security. Therefore, a cryptosystem that resists QFS is considered to be secure against quantum attacks. The quasi-cyclic code-based cryptosystem has been demonstrated to be resistant to QFS-based attacks. Furthermore, there are no known attacks on the cryptosystem.

The security of a code-based cryptosystem is fundamentally tied to the indistinguishabil-

ity of the underlying code from a random linear code. We explore the development of a Niederreiter-like cryptosystem based on quasi-twisted codes and thoroughly analyze its security. Quasi-twisted codes [17], are a generalization of cyclic codes, constacyclic codes, and quasi-cyclic codes. By utilizing the general structure of quasi-twisted codes, as opposed to quasi-cyclic codes, we present a broader alternative to cryptosystems based on quasi-cyclic codes. We show that our cryptosystem can withstand QFS.

Contents

Abstract **xi**

1 Introduction **1**

2 Coding Theory **3**

2.1 Finite Fields 3

2.2 Vector space 5

2.3 Linear Codes 7

2.4 Examples of linear codes 9

2.5 Bounds on the minimum distance of codes 12

3 Cryptography **19**

3.1 Public Key Cryptography 19

3.2 Code-based Cryptography 21

3.3 Variants of McEliece and Niederreiter cryptosystems 24

4 Quasi-cyclic codes **29**

4.1 Reduced Gröbner Basis 29

4.2 Spectral Analysis and Lower Bounds 31

4.3 Syndrome-based Decoding Algorithm 33

5 Decoding quasi-twisted codes **37**

5.1 Spectral theory of quasi-twisted codes 37

5.2 HT-like bound on the minimum distance of quasi-twisted codes 39

5.3 Syndrome-based decoding of quasi-twisted codes 41

5.4 Decoding Algorithm up to the HT-like bound 44

6 Niederreiter-like cryptosystem based on quasi-twisted codes **49**

6.1 Introduction 49

6.2 Classical Security 51

6.3 Quantum Security 51

7 Conclusion **61**

Chapter 1

Introduction

In 1994, Peter Shor developed a quantum algorithm for integer factorization which is a potential threat to existing public key cryptosystems once large-scale quantum computers get deployed. Post-quantum cryptography has emerged as an alternative technique to secure information. This field focuses on developing cryptosystems that are resistant to attacks from a quantum computer. Ideal cryptosystems will have sizes and computational complexity comparable to existing classical systems. Such cryptosystems are said to be *quantum-resistant*. Code-based cryptography is one of the candidates for post-quantum cryptography, whose security relies on the use of hard problems from algebraic coding theory, such as the NP-complete problem of decoding a random linear code, to develop the cryptosystem. A detailed explanation and background for broader topics in code-based cryptography can be found in [1] and [2].

The McEliece cryptosystem [3], or equivalently, the Niederreiter cryptosystem [4], are promising code-based quantum-resistant cryptosystems. These facilitate faster encryption and decryption, but the large size of public and private keys renders their implementation almost impossible. This challenge arises due to the use of Goppa codes to develop them. Numerous efforts have been made to replace the Goppa codes with codes that reduce key sizes while maintaining the security. In chapter 6, we propose one such cryptosystem. Our adaptation is based on a Neiderreiter-like cryptosystem developed in [15] and [16], where the underlying code is a quasi-cyclic code. This cryptosystem withstands Quantum Fourier Sampling (QFS) attacks, which is a measure of security. It is not only quantum secure but

also offers enhanced performance in terms of transmission and encryption rates, along with significantly smaller key sizes. We use quasi-twisted codes [17], which are a generalization of quasi-cyclic codes, to develop our cryptosystem and analyze its security.

To successfully develop a secure and efficient code-based cryptosystem, it is essential to have an effective decoding algorithm for the underlying code. To achieve this, we propose a syndrome-based decoding approach in section 5.4, which efficiently corrects errors in accordance with a Hartmann-Tzeng (HT)-like bound on the minimum distance of the code, which defines the maximum number of errors that can be reliably corrected without compromising the structure of the code. The algorithm that we describe has a computational complexity which is quadratic in code length.

This thesis is organised as follows: Chapters 2 and 3 give all the necessary background information to understand the thesis. Chapter 2 talks about the linear algebra basics and introduces the concepts in coding theory. In Chapter 3, we see a detailed explanation of public key cryptography and code-based cryptography, including the pioneers in the latter (namely, McEliece and Niederreiter cryptosystems) and some of their notable variants. This chapter concludes with a quantum-secure modification of the Niederreiter cryptosystem based on quasi-cyclic codes. In the subsequent Chapter 4, we analyze these quasi-cyclic codes, which are a generalization of cyclic codes. A more generalized code, quasi-twisted codes, are studied in Chapter 5 to understand their applicability to a Niederreiter-like cryptosystem. Subsequently a syndrome-based decoding algorithm is presented in this chapter in Section 5.3. Ultimately, Chapter 6 describes a new variant of Niederreiter-like cryptosystem based on quasi-twisted codes.

Chapter 2

Coding Theory

Coding theory is concerned with the reliability of communication by handling distortions due to channel noise. When a message is transmitted through a noisy channel, it is necessary to detect and correct the errors in the received message so as to retrieve the originally sent information. To do this efficiently, messages are encoded by mapping them to a larger space before sending them over. The encoded message is called a 'codeword', and the set of codewords form a code. On the receiving end, a decoder maps the received message back to the nearest codeword. ('nearest' is with respect to a distance metric defined on the larger space.) Hamming distance is a commonly used distance metric (equation 2.1).

In the next two sections, we will review the fundamental concepts of linear algebra essential for understanding coding theory. Later in the chapter, we will revisit key aspects of coding theory that are relevant to the objectives of this thesis.

2.1 Finite Fields

A field $(\mathbb{F}, +, \cdot)$ is a non-empty set of elements along with two operations, $+$ (addition) and $\cdot$ (multiplication), such that it satisfies the following properties:

1. Closure Property: $\forall a, b \in \mathbb{F}, a + b \in \mathbb{F}$ and $a \cdot b \in \mathbb{F}$.
2. Commutative Property: $\forall a, b \in \mathbb{F}$,

- $a + b = b + a$
- $a \cdot b = b \cdot a$

3. Associative Property: $\forall a, b, c \in \mathbb{F}$,
   - $(a + b) + c = a + (b + c)$
   - $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
4. Distributive Property: $\forall a, b, c \in \mathbb{F}$,
   - $a \cdot (b + c) = a \cdot b + a \cdot c$.
5. Existence of identity:
   - Additive identity: There exists an additive identity element $0_{\mathbb{F}}$ such that $a + 0_{\mathbb{F}} = a, \forall a \in \mathbb{F}$.
   - Multiplicative identity: There exists a multiplicative identity element $1_{\mathbb{F}}$ such that $a \cdot 1_{\mathbb{F}} = a, \forall a \in \mathbb{F} \setminus \{0\}$.
6. Existence of inverse:
   - Additive inverse: For all $a \in \mathbb{F}$, there exists an additive inverse $-a \in \mathbb{F}$ such that $a + (-a) = 0$.
   - Multiplicative inverse: For all $a \in \mathbb{F} \setminus \{0\}$, there exists a unique multiplicative inverse $b \in \mathbb{F}$, such that $a \cdot b = 1_{\mathbb{F}}$.

A field with a finite number of elements is a **Finite field**, where the *order* of the field is the total number of elements in the field. We denote a field with q elements as $\mathbb{F}_q$. The non-zero elements of a finite group form a multiplicative group. The multiplicative group of a finite field of order q is denoted as $\mathbb{F}_q^*$. If $q = p^m$, where p is a prime number, then the order of a field is always p^m for some $m \in \mathbb{Z}^+$. A *prime field* is defined as $\mathbb{F}_p := \mathbb{Z}/\mathbb{Z}_p$, for some prime number p.

The *characteristic* of a field is the smallest positive integer p such that $p \cdot 1_{\mathbb{F}} = 0$. If such a p does not exist, then the field is said to have its characteristic as 0. A field of characteristic p has order p^m, for $m \geq 1$. The characteristic of a field is either 0 or a prime number p.

A *primitive* element or *generator* of a finite field $\mathbb{F}_q$ is the element $\alpha \in \mathbb{F}_q$ such that $\mathbb{F}_q = \{0\} \cup \{\alpha^i : 0 < i < q\}$. For every element $a \in \mathbb{F}_q$, we have $a^q = a$.

The order of a non-zero element $a \in \mathbb{F}_q$ ($ord(a)$) is defined as the least positive integer k such that $a^k = 1$. The order of the primitive element is $ord(\alpha) = q - 1$.

A *polynomial ring* over a field $\mathbb{F}_q$ is defined as $\mathbb{F}_q[X] := \left\{ \sum_{i=0}^{n} a_i X^i : a_i \in \mathbb{F}, n \geq 0 \right\}$. When a polynomial $p(X) \in \mathbb{F}_q[X]$ is said to be *irreducible* over $\mathbb{F}_q$, then neither does it have any roots in $\mathbb{F}_q$ nor can it be written as a product of two polynomials (which are not units) of $\mathbb{F}_q[X]$. For an irreducible polynomial $p(X) \in \mathbb{F}_q[X]$, the residue class $\mathbb{F}_q[X]/p(X)$ is a field.

2.2 Vector space

A vector space is defined over a finite field; and the elements of the vector space and the field are called vectors and scalars, respectively. A nonempty set V over the field $\mathbb{F}_q$ along with two operations - vector addition ($+$) and scalar multiplication ($\cdot$) - is a vector space over $\mathbb{F}_q$ if it satisfies the following properties. For all $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w} \in V$ and $\lambda, \mu \in \mathbb{F}_q$,

1. (Closure)
   - (under vector addition) $\mathbf{u} + \mathbf{v} \in V$
   - (under Scalar Multiplication) $\lambda \cdot \mathbf{v} \in V$
2. (Commutative) $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
3. (Associative) $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$
4. (Additive identity) $\exists \mathbf{0} \in V,$ such that $\mathbf{0} + \mathbf{v} = \mathbf{v} = \mathbf{v} + \mathbf{0}, \forall \mathbf{v} \in V$
5. (Additive inverse) $\forall \mathbf{v} \in V$, $\exists (\text{-}\mathbf{v}) \in V$ such that $\mathbf{v} + \text{-}\mathbf{v} = \mathbf{0} = \text{-}\mathbf{v} + \mathbf{v}$
6. (Multiplicative identity) If $1_{\mathbb{F}}$ is the multiplicative identity of $\mathbb{F}_q$, then $1_{\mathbb{F}} \cdot \mathbf{v} = \mathbf{v}$
7. (Distributive)
   - $\lambda \cdot (\mathbf{u} + \mathbf{v}) = \lambda \cdot \mathbf{u} + \lambda \cdot \mathbf{v}$
   - $(\lambda + \mu) \cdot \mathbf{v} = \lambda \cdot \mathbf{v} + \mu \cdot \mathbf{v}$

- $(\lambda\mu)\mathbf{v} = \lambda(\mu \cdot \mathbf{v})$

A set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ is said to be *linearly independent* if, for $\lambda_i \in \mathbb{F}_q, 1 \leq i \leq m$, $\lambda_1\mathbf{v}_1, \lambda_2\mathbf{v}_2, \ldots, \lambda_m\mathbf{v}_m = 0 \implies \lambda_i = 0, \forall i : 1 \leq i \leq m$. Consider a nonempty subset $S = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ of the vector space V over $\mathbb{F}_q$. Then, the (linear) span of S is defined as:

$$\langle S \rangle = \{\lambda_1\mathbf{v}_1 + \lambda_2\mathbf{v}_2 + \ldots + \lambda_k\mathbf{v}_k : \lambda_i \in \mathbb{F}_q\}$$

$\langle S \rangle$ gives a subspace of V and is called the *subspace spanned (or generated) by* S. If for a subset $B = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}, \langle B \rangle = V$ and if B is linearly independent, then B is called a *basis for* V. There can be many bases for a vector space, but each of these bases will have a fixed number of elements, which is called the *dimension* of the vector space. It is denoted as $dim_{\mathbb{F}_q}(V)$ or simply $dim(V)$. If $dim(V) = k$, then V contains q^k elements.

Let $\mathbf{u}, \mathbf{v}$ be two vectors in a vector space V. The *inner product*, $\mathbf{u} \cdot \mathbf{v}$ of the vectors $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is a scalar defined as follows:

$$\mathbf{u} \cdot \mathbf{v} := u_1v_1 + u_2v_2 + \ldots + u_nv_n$$

If $\mathbf{u} \cdot \mathbf{v} = 0$ for some vectors $\mathbf{u}, \mathbf{v} \in V$, then the vectors $\mathbf{u}$ and $\mathbf{v}$ are said to be *orthogonal*.

Matrices: An $m \times n$ *matrix* with entries from the field $\mathbb{F}$ is an array of elements $a_{ij} \in \mathbb{F}$, expressed as,

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}_{(m \times n)}$$

The rows of the matrix A form vectors in $\mathbb{F}^n$ and the columns of A correspond to vectors in $\mathbb{F}^m$. The set of all such $m \times n$ matrices with entries from $\mathbb{F}$, together with the operations of matrix addition and scalar multiplication, forms a *vector space* and is denoted as $\mathbf{M}_{m \times n}(\mathbb{F})$.

For a matrix $A \in \mathbf{M}_{m \times n}(\mathbb{F})$, an *eigenvector of* A is defined as a nonzero vector $v \in \mathbb{F}^n$ such that for some scalar $\lambda \in \mathbb{F}$, $Av = \lambda v$. This scalar λ is defined as the *eigenvalue of* A, with respect to the eigenvector v. The set of all the eigenvectors corresponding to an eigenvalue is called the *eigenspace* $\mathcal{V}$ with respect to that eigenvalue.

2.3 Linear Codes

Consider a field, $\mathbb{F}_q$ where q is a prime power and a vector space over this field, $\mathbb{F}_q^n$ for some positive integer n. A *linear code* over $\mathbb{F}_q$ is a subspace of the vector space $\mathbb{F}_q^n$. In other words, any subset $\mathcal{C}$ of $\mathbb{F}_q^n$ is a linear code if it satisfies the following two properties.

1. $\mathbf{u} + \mathbf{v} \in \mathcal{C}$, $\forall \mathbf{u}, \mathbf{v} \in \mathcal{C}$.

2. $a \cdot \mathbf{v} \in \mathcal{C}$, $\forall \mathbf{v} \in \mathcal{C}$.

If $\mathcal{C}$ is a $k-$dimensional subspace of $\mathbb{F}_q^n$, then the linear code is said to be an $[n, k]$ code. A linear code of length n and dimension k can also be denoted as an $[n, k, d]$ code, where d is the minimum distance of the code.

Distance of a code: It is the minimum of the distances between any two codewords. That is,

$$d_{\mathcal{C}} = min\{d(u, v) : u, v \in \mathcal{C}, u \neq v\}.$$

where, $d_{\mathcal{C}}$ is the distance of the linear code, $\mathcal{C}$ and $d(u, v)$ is the distance between the codewords u and v for a distance metric d.

Remark. *A decoder can efficiently detect up to $d_{\mathcal{C}} - 1$ errors and correct up to $t_{\mathcal{C}} := \lfloor \frac{d_{\mathcal{C}}-1}{2} \rfloor$ errors. This integer $t_{\mathcal{C}}$ is termed the error-correcting capacity of the code $\mathcal{C}$.*

The ***weight*** of a codeword $c \in \mathcal{C}$ is the number of non-zero entries in c, i.e., $wt(c) := |i : c_i \neq 0|$, where $|.|$ denotes the cardinality of the set.

Hamming distance: It is a commonly used distance metric. The Hamming distance on $\mathbb{F}_q^n$, d_H is defined as follows:

$$d_H(u, v) := |\{i : u_i \neq v_i\}| \tag{2.1}$$

where $|.|$ denotes the cardinality of the set.
For linear codes, we can rewrite it as:

$$d_H(u, v) := |\{i : u_i - v_i \neq 0\}| = wt(u - v)$$

Thus, we can define the *minimum distance* of a linear code C as:

$$d_H(C) := min\{wt(x - y) : x, y \in C, x \neq y\} = min\{wt(x) : x \in C \backslash 0\}$$

Consider an $[n, k]$ linear code $\mathcal{C}$. The $k \times n$ matrix G whose rows form a basis of the code $\mathcal{C}$ is defined to be a ***generator matrix*** of the code $\mathcal{C}$. For any vector $\mathbf{v} \in \mathbb{F}_q^n$, $\mathbf{v}G$ is a codeword of the linear code $\mathcal{C}$.
Note that the rows of G are linearly independent; therefore, $rank(G) = k$. A generator matrix G in the form $[I_k|G']$ is said to be in the standard form, where I_k denotes the $k \times k$ identity matrix and G' is a $k \times (n-k)$ matrix.

Recall that two vectors are orthogonal if their inner product is zero.

Dual code: For an $[n, k]$ linear code $\mathcal{C}$, the subspace of $\mathbb{F}_q^n$ containing all those vectors that are orthogonal to every vector (codeword) in $\mathcal{C}$ forms the *dual code* of $\mathcal{C}$. It is denoted as $\mathcal{C}^\perp$.

$$\mathcal{C}^\perp := \left\{\mathbf{u} \in \mathbb{F}_q^n : \mathbf{u} \cdot \mathbf{v} = 0, \forall \mathbf{v} \in \mathcal{C}\right\}$$

The following are immediate from the definition:

- $\mathbf{v} \in \mathcal{C}^\perp \iff \mathbf{v}G^\intercal = 0$.
- $\mathcal{C}^\perp$ is an $[n, n-k]$ linear code.

A generator matrix of the dual code code of $\mathcal{C}$ forms the ***parity-check matrix*** H of $\mathcal{C}$. It is an $n-k \times n$ matrix such that $GH^\intercal = \mathbf{0}$. For a parity-check matrix H, $Hc^\intercal = 0$, $\forall c \in \mathcal{C}$. Observe that, $\mathcal{C}$ is the null space of H. It is a full rank matrix, i.e, $rank(H) = n - k$. A parity-check matrix in the form $[H'|I_{n-k}]$ is said to be in the standard form, where H' is an $(n-k) \times k$ matrix and I_{n-k} is an $(n-k) \times (n-k)$ identity matrix.

2.3.1 Syndrome Decoding Problem

NP-hardness or NP-completeness of a problem is a measure of difficulty of solving the problem (here, NP stands for *Non-deterministic Polynomial-time*). A problem is NP-complete if there doesn't exist a polynomial time algorithm to solve that problem. However, given a solution to an NP-complete problem, it can be verified in polynomial time by a deterministic machine.

Recall that code-based cryptography (which we will see in detail in the next chapter) is based on hard problems from algebraic coding theory. Decoding a random linear code and

the syndrome decoding problem are two examples of NP-complete problems in coding theory. Therefore, these are adopted in code-based cryptography.

Further details on the syndrome decoding problem are given below.

Syndrome: Let H be a parity-check matrix of an $[n, k]$ linear code $\mathcal{C}$ and x be any vector in $\mathbb{F}_q^n$. Then the *syndrome of* x is the vector $S(x)$ of length $n - k$, defined as follows:

$$S(x) := xH^\intercal$$

Notice that $S(x) = 0$ if and only if $x \in \mathcal{C}$ is a codeword.

Syndrome Decoding Problem (SDP): Let $t \in \mathbb{N}$, a parity-check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$ and a syndrome $s \in \mathbb{F}_q^{n-k}$ be given. Then the syndrome decoding problem is to find a vector $e \in \mathbb{F}_q^n$ such that $wt(e) \leq t$ and $eH^T = s$.
SDP is proven to be an NP-complete problem.

2.4 Examples of linear codes

In this section, we recall definitions of various linear codes that are relevant to the results of this thesis. Recall that $\mathbb{F}_q$ is a finite field with $q \in \mathbb{Z}^+$ elements.

Definition 2.4.1. ***Reed-Solomon codes and generalized Reed-Solomon codes:***

Consider two n-tuples α and β $(\in \mathbb{F}_q^n)$ such that $\alpha_i \neq \alpha_j$ $\forall$ $i \neq j \in \{1, 2, ..., n\}$ and $\beta_i \neq 0$ $\forall$ $i \in \{1, 2, ..., n\}$. Let $k \leq n \leq q \in \mathbb{Z}^+$. Then a ***generalized Reed-Solomon code*** *of length n and dimension k is defined as follows:*

$$GRS_{n,k}(\alpha, \beta) := \{(\beta_1 f(\alpha_1), \beta_2 f(\alpha_2), ..., \beta_n f(\alpha_n)) | f \in \mathbb{F}_q[X], deg(f) < k\}.$$

A ***Reed-Solomon code*** *of length n and dimension k refers to an instance of the generalized Reed-Solomon code with $\beta = \{1, 1, ..., 1\}$ and can be defined as follows:*

$$RS_{n,k}(\alpha) := \{(f(\alpha_1), f(\alpha_2), ..., f(\alpha_n)) | f \in \mathbb{F}_q[X], deg(f) < k\}.$$

Definition 2.4.2. ***Subfield Subcodes:***

Suppose $\mathcal{C}$ is a linear code of length n and dimension k defined over the field $\mathbb{F}_{q^m}$, where $m \in \mathbb{Z}^+$. Then the ***subfield subcode of*** *$\mathcal{C}$, $\mathcal{C}_{\mathbb{F}_q}$, over the field $\mathbb{F}_q$ is defined as follows:*

$$\mathcal{C}_{\mathbb{F}_q} := \mathcal{C} \cap \mathbb{F}_q^n$$

The dimension of $\mathcal{C}_{\mathbb{F}_q}$, k' is at least $n - m(n-k)$ and minimum distance of $\mathcal{C}_{\mathbb{F}_q}$ is lower bounded by d, where d is the minimum distance of $\mathcal{C}$.

Definition 2.4.3. ***Alternant codes:***

This is an instance of Subfield Subcodes where $\mathcal{C}$ is a GRS code. Consider two n-tuples $\alpha \in \mathbb{F}_{q^m}^n$ and $\beta \in (\mathbb{F}_{q^m}^*)^n$, such that α_i's are pairwise distinct. Then the ***alternant codes*** *are defined as follows:*

$$\mathcal{A}_{m,n,k}(\alpha, \beta) := GRS_{m,n,k}(\alpha, \beta) \cap \mathbb{F}_q^n$$

The dimension of the alternant code $\mathcal{A}_{m,n,k}$, k' is at least $n - m(n-k)$ and its minimum distance is at least $n - k + 1$.

Definition 2.4.4. ***Goppa codes:***

Classical Goppa codes are an example of Alternant codes. Consider GRS codes with β to be such that each $\beta_i = \prod_{j \neq i}(\alpha_j - \alpha_i)G(\alpha_i)^{-1}$, where $G(x) \in \mathbb{F}_{q^m}[X]$. Then, the ***$q$-ary Goppa code*** *of length n can be defined as follows:*

$$GRS_{m,n,k}(\alpha, \beta) \cap \mathbb{F}_q^n.$$

Definition 2.4.5. ***Expanded codes:***

Let m be a positive integer and γ be a primitive element of $\mathbb{F}_{q^m}$. Consider the following $\mathbb{F}_q$-linear isomorphism:

$$\begin{aligned} \phi_n : \quad & \mathbb{F}_{q^m}^n \quad \longrightarrow \quad \mathbb{F}_q^{mn} \\ & \phi_n(c_0, c_1, ..., c_{n-1}) \quad \longmapsto (\phi(c_0), \phi(c_1), ..., \phi(c_{n-1})) \end{aligned}$$

where, $\phi(c_i) = \phi(a_0^{(i)} + a_1^{(i)}\gamma + ... + a_{m-1}^{(i)}\gamma^{m-1}) = (a_0^{(i)}, a_1^{(i)}, ..., a_{m-1}^{(i)})$. Let $\mathcal{C}$ be $[n, k]$ linear code over $\mathbb{F}_{q^m}$. The image of codewords $c \in \mathcal{C}$ under the map ϕ_n is termed as an ***expanded code*** *of $\mathcal{C}$ with respect to the primitive element γ, $\hat{\mathcal{C}}$. That is, $\hat{\mathcal{C}} := \{\phi_n(c) : c \in \mathcal{C}\}$. Note that, $\hat{\mathcal{C}}$ is an $[mn, mk]$-linear code.*

Definition 2.4.6. ***Cyclic codes:***

Let $\mathcal{C}$ be an $[n,k]$ linear code. Suppose $c = (a_0, a_1, a_2, ..., a_{n-1}) \in \mathcal{C}$ is a codeword. Then $\mathcal{C}$ is a ***cyclic code*** *if any cyclic shift of a codeword c, given by $(a_{n-1}, a_0, a_1, ..., a_{n-2}) \in \mathcal{C}$ is also a codeword in $\mathcal{C}$.*

We can map a codeword $c \in \mathcal{C}$ to the polynomial $a_0 + a_1X + a_2X^2 + ... + a_{n-1}X^{n-1} \in \mathbb{F}_q[X]$. Notice that the cyclic code $\mathcal{C}$ now corresponds to an ideal in $\mathbb{F}_q[x]/(X^n - 1)$.

Definition 2.4.7. ***Constacyclic codes:***

Let $\mathcal{C}$ be an $[n,k]$ linear code and $\lambda \in \mathbb{F}_q^*$. Suppose $c = (a_0, a_1, a_2, ..., a_{n-1}) \in \mathcal{C}$ is a codeword, then $\mathcal{C}$ is a ***λ-constacyclic code*** *if*

$$(\lambda a_{n-1}, a_0, a_1, ..., a_{n-2}) \in \mathcal{C}$$

is also a codeword. Note that, the constacyclic code $\mathcal{C}$ corresponds to an ideal in $\mathbb{F}_q[X]/(X^n - \lambda)$.

Definition 2.4.8. ***Quasi-cyclic codes:***

Let $\mathcal{C}$ be $[n,k]$ linear code and l be a positive integer. Suppose $c = (a_0, a_1, a_2, ..., a_{n-1}) \in \mathcal{C}$ be a codeword. Then $\mathcal{C}$ is an ***l-quasi-cyclic (QC) code*** *(and l is its index) if a cyclic shift of c by l positions, given by*

$$(a_{n-l}, a_{n-l+1}, ..., a_{n-1}, a_0, ..., a_{n-l-1}) \in \mathcal{C}$$

is also a codeword. (here, the subscripts are taken modulo n.)

Notice that l always divides n. Let $\theta = n/l$. $\mathcal{C}$ then corresponds to a submodule of $\mathbb{F}_q[X]/(X^\theta - 1)^l$ in $\mathbb{F}_q[X]/(X^\theta - 1)$.

Definition 2.4.9. ***Quasi-twisted codes:***

Let $\mathcal{C}$ be an $[n,k]$ linear code. Let $\lambda \in \mathbb{F}_q^*$ and l be a positive integer. Suppose $c = (a_0, a_1, a_2, ..., a_{n-1}) \in \mathcal{C}$ be a codeword. Then $\mathcal{C}$ is a (λ, l) ***-quasi-twisted (QT) code*** *if a constacyclic shift of c by l positions given by,*

$$(\lambda a_{n-l}, \lambda a_{n-l+1}, ..., \lambda a_{n-1}, a_0, ..., a_{n-l-1}) \in \mathcal{C}$$

is also a codeword.(here, the subscripts are all taken modulo n.) Quasi-twisted codes were

introduced in [17].

Note that l always divides n. Let $\theta = n/l$. $\mathcal{C}$ then corresponds to a submodule of $\mathbb{F}_q[X]/(X^\theta - \lambda)^l$ in $\mathbb{F}_q[X]/(X^\theta - \lambda)$.

Remark 2.4.1. *Observe that cyclic codes, constacyclic codes and quasi-cyclic codes are all special cases of quasi-twisted codes. Consider a (λ, l)-quasi-twised code. Substituting $\lambda = 1$ or $q = 2$, it becomes an l-quasi-cyclic code and substituting $l = 1$, it becomes a λ-constacyclic code.*

2.5 Bounds on the minimum distance of codes

Determining the minimum distance of a code is a hard problem. Several (upper and lower) bounds have been defined on the minimum distance of a code. Such bounds provide insights into the error-correcting capacity of the code. Recall that for a linear $[n, k, d]$ code $\mathcal{C}$, the decoder can detect up to $d - 1$ errors and correct up to $t = \lfloor \frac{d-1}{2} \rfloor$ errors, where t is the error-correcting capacity of $\mathcal{C}$. That means the greater the minimum distance of $\mathcal{C}$, the better the ability to detect and correct errors during transmission of the code. An example of an upper bound is the Singleton bound. The Singleton bound of an $[n, k, d]$ linear code $\mathcal{C}$ is $d \leq n - k + 1$.

Here, we look at two such lower bounds that are required to understand the results proved in this thesis.

1. <u>BCH Bound:</u>

BCH bound is a lower bound on the minimum distance of a BCH (Bose-Chaudhuri-Hocquenghem) code. Independently, A. Hocquenghem, in 1959 and R. C. Bose and D. K. Ray-Chaudhuri, in 1960, introduced a class of cyclic codes, BCH codes. The construction of these codes involves a design distance parameter δ given by its generator polynomial $g(X)$. BCH codes satisfy the following theorem, thereby are ensured to have a lower bound on its minimum distance. This lower bound is termed the BCH bound.

Theorem 2.5.1. *[23, Thm. 1] Consider a cyclic code, $\mathcal{C}$ of length n and a generator polynomial, $g(X)$. Let $\beta \in \mathbb{F}_q^m$ be a non-zero element of order n. Let $\delta \geq 2$, a, n_1 be positive integers such that $gcd(n, n_1) = 1$. If $g(\beta^{a+in_1}) = 0$ for $i = 0, 1, ..., \delta - 2$, then the minimum distance of $\mathcal{C}$ is $d \geq \delta$.*

Proof.

From the definition of minimum distance of a code, we have that $d = min\{wt(x) : x \in C\backslash\{0\}\}$. Consider the proof by contradiction approach. Let us assume that there exists a codeword $c \in \mathcal{C}$ such that $wt(c) = \omega < \delta$. Let $c(X)$ be the corresponding code polynomial. Since $\mathcal{C}$ is cyclic, we can assume

$$c(X) = 1 + \sum_{i=1}^{\omega-1} \Lambda_i X^{a_i},$$

for some $0 \neq \Lambda_i \in \mathbb{F}_q$ and distinct positive integers $a_i < n$.

Let $\chi_i = \beta^{a_i}$ and $S_j = \sum_{i=1}^{\omega-1} \Lambda_i \chi_i^j$. That is, $S_j = \sum_{i=1}^{\omega-1} \Lambda_i \beta^{a_i j}$. We can rewrite as $S_j = c(\beta^j) - 1$. For all j where $g(\beta^j) = 0$, $S_j = -1$. For positive integers $a, n_1, \delta \geq 2$ and $i = 0, 1, \ldots, \delta - 2$, we have $g(\beta^{a+in_1}) = 0$. Therefore, $S_{a+in_1} = -1$.

Consider the polynomial $\sigma(X)$,

$$\begin{aligned}
\sigma(X) :=& \prod_{i=1}^{\omega-1} (X - \chi_i^{n_1}) \\
=& X^{\omega-1} - \sum_{i=1}^{\omega-1} \chi_i^{n_1} X^{\omega-2} + \sum_{i_1 \neq i_2} \chi_{i_1}^{n_1} \chi_{i_2}^{n_1} X^{\omega-3} + \ldots + (-1)^{\omega-1} \prod_{i=1}^{\omega-1} \chi_i^{n_1} \\
=& X^{\omega-1} + \sigma_1 X^{\omega-2} + \ldots + \sigma_{\omega-2} X + \sigma_{\omega-1}.
\end{aligned}$$

Note that each σ_i is a function of $\chi_i = \beta^{a_i}$ with $a_i > 0$. Therefore, $\sigma(1) = 1 + \sigma_1 + \ldots + \sigma_{\omega-2} + \sigma_{\omega-1} \neq 0$.

Given that β is a non-zero element of order n and $a_i < n$, we have $\chi_i \neq 1$. Since $gcd(n, n_1) = 1$, we have $\chi_i^{n_1} \neq 1$. Let us recall that $\sigma(X) = \prod_{i=1}^{\omega-1} (X - \chi_i^{n_1})$, and thus $\sigma(\chi_i^{n_1}) = 0$. More-

over, since $S_j = \sum_{i=1}^{\omega-1} \Lambda_i \chi_i^j$, we have

$$\begin{aligned} &S_{a+(\omega-1)n_1} + \sigma_1 S_{a+(\omega-2)n_1} + \ldots + \sigma_{\omega-2} S_{a+n_1} + \sigma_{\omega-1} S_a \\ &\quad = \sum_{i=1}^{\omega-1} \Lambda_i \chi_i^a \left(\chi_i^{(\omega-1)n_1} + \sigma_1 \chi_i^{(\omega-2)n_1} + \ldots + \sigma_{(\omega-2)} \chi_i^{n_1} + \sigma_{\omega-1} \right) \\ &\quad = \sum_{i=1}^{\omega-1} \Lambda_i \chi_i^a \Big(\sigma(\chi_i^{n_1}) \Big) = 0 \end{aligned} \tag{2.2}$$

Since $\omega < \delta$ and $S_{a+in_1} = -1$ for $i \in [\delta - 1]$,

$$\begin{aligned} S_{a+(\omega-1)n_1} + \sigma_1 S_{a+(\omega-2)n_1} + \ldots + \sigma_{\omega-2} S_{a+n_1} + \sigma_{\omega-1} S_a \\ = -1(1 + \sigma_1 + \ldots + \sigma_{\omega-2} + \sigma_{\omega-1}) \\ = -1(\sigma(1)). \end{aligned} \tag{2.3}$$

Therefore, from equations 2.2 and 2.3, we can conclude that $\sigma(1) = 0$. However, substituting $X = 1$ in the definition of $\sigma(X)$ had resulted in the conclusion that $\sigma(1) \neq 0$. This is a contradiction. This implies that there cannot exist a codeword with a weight less than δ. Therefore, $d \geq \delta$.

■

2. Hartmann-Tzeng (HT) Bound

HT bound was introduced by C. R. P. Hartmann and K. K. Tzeng in 1972. It is a lower bound on the minimum distance of cyclic codes, that provides a generalization to the BCH bound. Notice that in the BCH bound, the generator polynomial, $g(X)$, with a single set of consecutive roots, is considered. In HT-bound, generator polynomials with multiple sets of consecutive roots are considered, and a stricter lower bound on the minimum distance is achieved.

Theorem 2.5.2. *[23, Thm. 2] Consider a cyclic code, $\mathcal{C}$ of length n and a generator polynomial, $g(X)$. Let $\beta \in \mathbb{F}_q^m$ be a non-zero element of order n. Let $\delta \geq 2, a, n_1, n_2, s$ be positive integers such that $gcd(n, n_1) = 1$ and $gcd(n, n_2) = 1$. If $g(\beta^{a+in_1+jn_2}) = 0$ for $i = 0, 1, ..., \delta - 2$ and $j = 0, 1, ..., s$, then the minimum distance of $\mathcal{C}$ is $d \geq \delta + s$.*

Proof.

In Theorem 2.5.1, we saw that $d \geq \delta$. Let us use the proof-by-contradiction method. Let us assume that there exists a codeword $c \in \mathcal{C}$ of weight ω such that $\delta \leq \omega \leq \delta + s$. Let $c(X)$ be the corresponding code polynomial. Since $\mathcal{C}$ is cyclic, we can assume,

$$c(X) = 1 + \sum_{i=1}^{\omega-1} \Lambda_i X^{a_i}$$

Consider $\chi_i = \beta^{a_i}$ and $S_j = \sum_{i=1}^{\omega-1} \Lambda_i \chi_i^j$. That is, $S_j = \sum_{i=1}^{\omega-1} \Lambda_i \beta^{a_i j}$. We can rewrite as $S_j = c(\beta^j) - 1$. For all j where $g(\beta^j) = 0$, $S_j = -1$. For positive integers $a, n_1, n_2, \delta \geq 2$ and for $i \in [\delta - 1]$, $j \in [s + 1]$, we have $g(\beta^{a+in_1+jn_2}) = 0$. Therefore, $S_{a+in_1+jn_2} = -1$.

Now, consider three polynomials $\sigma(X), \sigma_1(X)$ and $\sigma_2(X)$ as defined below:

$$\begin{aligned}
\sigma_1(X) &= \prod_{i=1}^{\delta-2} (X - \chi_i^{n_1}) \\
&= X^{\delta-2} - \sum_{i=1}^{\delta-2} \chi_i^{n_1} X^{\delta-3} + \sum_{i_1 \neq i_2} \chi_{i_1}^{n_1} \chi_{i_2}^{n_1} X^{\delta-4} + \ldots + (-1)^{\delta-2} \prod_{i=1}^{\delta-2} \chi_i^{n_1} \\
&= X^{\delta-2} + \sigma_1^{(1)} X^{\delta-3} + \ldots + \sigma_{\delta-3}^{(1)} X + \sigma_{\delta-2}^{(1)} \\
\sigma_2(X) &= \prod_{i=\delta-1}^{\omega-1} (X - \chi_i^{n_2}) \\
&= X^{\omega-\delta+1} - \sum_{i=1}^{\omega-\delta+1} \chi_i^{n_1} X^{\omega-\delta} + \sum_{i_1 \neq i_2} \chi_{i_1}^{n_1} \chi_{i_2}^{n_1} X^{\omega-\delta-1} + \ldots + (-1)^{\omega-\delta+1} \prod_{i=1}^{\omega-\delta+1} \chi_i^{n_1} \\
&= X^{\omega-\delta+1} + \sigma_1^{(2)} X^{\omega-\delta} + \ldots + \sigma_{\omega-\delta}^{(2)} X + \sigma_{\omega-\delta+1}^{(2)}
\end{aligned}$$

and

$$\sigma(X) = \sigma_1(X)\sigma_2(X)$$

Notice that each $\sigma_i^{(j)}$, for $i \in [\delta - 1]$ and $j = 1, 2$, is a function of $\chi_i = \beta^{a_i}$ with $a_i \neq 0$. Given that β is a non-zero element of order n and $a_i < n$, we have $\chi_i \neq 1$. Recall that $gcd(n, n_1) = 1$ and $gcd(n, n_2) = 1$. Therefore, we have $\chi_i^j \neq 1$ for $j = n_1, n_2$. As a result, we see that $\sigma_1(X) \neq 0$ and $\sigma_2(X) \neq 0$. Thereby, we have $\sigma(1) \neq 0$.

Recall that $\sigma_1(X) := \prod_{i=1}^{\delta-2} (X - \chi_i^{n_1})$ and $\sigma_2(X) := \prod_{i=\delta-1}^{\omega-1} (X - \chi_i^{n_2})$. Thus, we get $\sigma_1(\chi_i^{n_1}) =$

0 and $\sigma_2(\chi_i^{n_2}) = 0$. Since $S_j = \sum_{i=1}^{\omega-1} \Lambda_i \chi_i^j$, we have

$$
\begin{aligned}
&\Big(S_{a+(\delta-2)n_1+(\omega-\delta+1)n_2} + \sigma_1^{(1)} S_{a+(\delta-3)n_1+(\omega-\delta+1)n_2} + \ldots + \sigma_{\delta-2}^{(1)} S_{a+(\omega-\delta+1)n_2}\Big) \\
&\qquad + \sigma_1^{(2)}\Big(S_{a+(\delta-2)n_1+(\omega-\delta)n_2} + \sigma_1^{(1)} S_{a+(\delta-3)n_1+(\omega-\delta)n_2} + \ldots + \sigma_{\delta-2}^{(1)} S_{a+(\omega-\delta)n_2}\Big) \\
&\qquad + \ldots + \sigma_{\omega-\delta-1}^{(2)}\Big(S_{a+(\delta-2)n_1} + \sigma_1^{(1)} S_{a+(\delta-3)n_1} + \ldots + \sigma_{\delta-2} S_a\Big) \\
&\qquad = \sum_{i=1}^{\omega-1} \Lambda_i \chi_i^a \Bigg(\chi_i^{(\delta-2)n_1} \chi_i^{(\omega-\delta+1)n_2} + \sigma_1^{(1)} \chi_i^{(\delta-3)n_1} \chi_i^{(\omega-\delta+1)n_2} + \ldots + \sigma_{\delta-2}^{(1)} \chi_i^{(\omega-\delta+1)n_2} \\
&\qquad\quad + \sigma_1^{(2)}\Big(\chi_i^{(\delta-2)n_1} \chi_i^{(\omega-\delta)n_2} + \sigma_1^{(1)} \chi_i^{(\delta-3)n_1} \chi_i^{(\omega-\delta)n_2} + \ldots + \sigma_{\delta-2}^{(1)} \chi_i^{(\omega-\delta)n_2}\Big) \\
&\qquad\quad + \ldots + \sigma_{(\omega-\delta+1)}^{(2)}\Big(\chi_i^{(\delta-2)n_1} + \sigma_1^{(1)} \chi_i^{(\delta-3)n_1} + \ldots + \sigma_{\delta-2}^{(1)}\Big)\Bigg) \\
&\qquad = \sum_{i=1}^{\omega-1} \Lambda_i \chi_i^a \Bigg(\sigma(\chi_i^{n_1}) \chi_i^{(\omega-\delta+1)n_2} + \sigma_1^{(2)}\Big(\sigma(\chi_i^{n_1}) \chi_i^{(\omega-\delta)n_2}\Big) + \ldots + \sigma_{\omega-\delta+1}^{(2)}\Big(\sigma(\chi_i^{n_1})\Big)\Bigg) \\
&\qquad = \sum_{i=1}^{\omega-1} \Lambda_i \chi_i^a \sigma_1(\chi_i^{n_1})\Big(\chi_i^{(\omega-\delta+1)n_2} + \sigma_1^{(2)} \chi_i^{(\omega-\delta)n_2} + \ldots + \sigma_{\omega-\delta+1}^{(2)}\Big) \\
&\qquad = \sum_{i=1}^{\omega-1} \Lambda_i \chi_i^a \sigma_1(\chi_i^{n_1}) \sigma_2(\chi_i^{n_2}) = 0
\end{aligned}
\tag{2.4}
$$

Since $\delta \le \omega < \delta + s$ and $S_{a+in_1+jn_2} = -1$ for $i \in [\delta - 1], j \in [s+1]$, we get,

$$
\begin{aligned}
&\Big(S_{a+(\delta-2)n_1+(\omega-\delta+1)n_2} + \sigma_1^{(1)} S_{a+(\delta-3)n_1+(\omega-\delta+1)n_2} + \ldots + \sigma_{\delta-2} S_{a+(\omega-\delta+1)n_2}\Big) \\
&\qquad + \sigma_1^{(2)}\Big(S_{a+(\delta-2)n_1+(\omega-\delta)n_2} + \sigma_1^{(1)} S_{a+(\delta-3)n_1+(\omega-\delta)n_2} + \ldots + \sigma_{\delta-2}^{(1)} S_{a+(\omega-\delta)n_2}\Big) \\
&\qquad + \ldots + \sigma_{\omega-\delta-1}^{(2)}\Big(S_{a+(\delta-2)n_1} + \sigma_1^{(1)} S_{a+(\delta-3)n_1} + \ldots + \sigma_{\delta-2}^{(1)} S_a\Big) \\
&\qquad = -1\Big(\big(1 + \sigma_1^{(1)} + \ldots + \sigma_{\delta-2}^{(1)}\big) + \sigma_1^{(2)}\big(1 + \sigma_1^{(1)} + \ldots + \sigma_{\delta-2}^{(1)}\big) \\
&\qquad\quad + \ldots + \sigma_{\omega-\delta-1}^{(2)}\big(1 + \sigma_1^{(1)} + \ldots + \sigma_{\delta-2}^{(1)}\big)\Big) \\
&\qquad = -1\Big(\sigma_1(1) + \sigma_1^{(2)}\sigma_1(1) + \ldots + \sigma_{\omega-\delta-1}^{(2)}\sigma_1(1)\Big) \\
&\qquad = -1\sigma_1(1)\sigma_2(1) = -1\sigma(1)
\end{aligned}
\tag{2.5}
$$

Thus, we can conclude from equations 2.4 and 2.5 that $\sigma(1) = 0$. However, substituting $X = 1$ in the definitions of $\sigma_1(X)$ and $\sigma_2(X)$ implies that $\sigma(1) \neq 0$, which is a contradiction. Therefore, there cannot exist a codeword with a weight less than $\delta + s$. In other words, $d \geq \delta + s$.

■

Remarks 2.5.1. *Note that the lower bound on the minimum distance of a cyclic code as given in theorem 2.5.2, $d \geq \delta + s$ holds when $gcd(n, n_2) < \delta$.*

Chapter 3

Cryptography

Cryptography (*kryptós* "hidden"; *graphein* "to write") is the art of securing communication in the presence of adversaries by converting (*encrypting*) the messages (*plaintext*) so that they can only be converted back (*decrypted*) by the intended person, that is the person who has the secret key. An encrypted message is termed as a *ciphertext*. The security of a cryptosystem is based on the hardness of the mathematical problem involved in the encryption (or, equivalently the decryption).

Suppose Alice wants to send a message to Bob through a channel that Eve also has access to. Then Alice encrypts the message, using a key, to the corresponding cipher and sends it over to Bob, who can then use the appropriate key to decrypt the cipher and get back the message. So long as the key is inaccessible to Eve, the communication remains secure.

3.1 Public Key Cryptography

This is also known as asymmetric cryptography. In this form of cryptography, encryption and decryption are done using two different keys, namely, the public key and the private key. The security of a cryptosystem relies on the hardness of the underlying mathematical problem. In asymmetric cryptography, integer factorization and discrete log in both finite fields and elliptic curves are involved. For example, RSA is a widely used public key cryptosystem and it exploits the hardness of integer factorization problem. The public key

of the RSA cryptosystem includes a large integer n, and its private key depends on the factors of n. Factorising such a large number is computationally impossible, which makes RSA secure (in the classical setting).

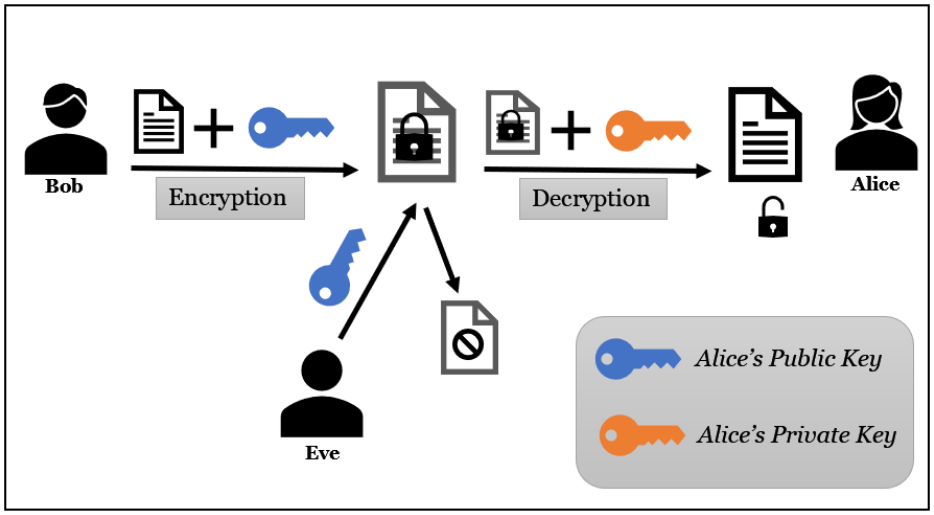


Figure 3.1: Public Key Cryptography

Consider Figure 3.1. It represents a public key cryptosystem. Suppose Bob wants to send a message to Alice. Then, Bob encrypts the message using Alice's public key that she would have already published, and he sends this encrypted message (ciphertext) to her. Any eavesdropper in the network cannot decrypt the ciphertext as they do not have access to Alice's private key. Alice uses her secret key to decrypt Bob's message.

Public key cryptography has its applications in digital signatures, key encapsulation mechanisms and public key encryption frameworks. These are briefly explained below.

1. **Digital Signature:** This application of asymmetric cryptography is used for authorization. Suppose Alice wants to send some data to Bob. Alice can sign the data using her private key and send the data along with the so-created digital signature to Bob. Bob can now use this signature to verify that the data is indeed from Alice by decrypting it using Alice's public key.
2. **Key Encapsulation Mechanism (KEM):** This is used to share a secret key between two parties so as to use it for future encryption/decryption. It has its application in sharing secret keys for symmetric encryption. Suppose Bob wants to establish a secret key between himself and Alice. Then, Bob generates a shared key by using Alice's public key and sends over this key along with an encapsulation of the same

(essentially a ciphertext). Alice can now decapsulate this ciphertext using her private key to get the shared secret key.

3. **Public Key Encryption (PKE):** This is essentially what Fig 3.1 depicts. We will now look at the Public Key Encryption (PKE) framework in detail. Any PKE framework comprises of three main steps, as explained below. Suppose Bob wants to send a message to Alice.

   (a) Key Generation: Alice generates her public and private keys. She reveals her public key and keeps her private key.

   (b) Encryption: Bob encrypts his message using Alice's public key to get a ciphertext, which he sends to Alice.

   (c) Decryption: Alice decrypts the ciphertext using her private key to get back Bob's message.

3.2 Code-based Cryptography

Given the rapid advancements in the field of quantum computing, the need for a robust post-quantum cryptosystem has become increasingly urgent. The National Institute of Standards and Technology (NIST) in the United States is standardizing post-quantum cryptosystems. code-based cryptography is among the few candidates for the NIST's standardization of post-quantum cryptosystems. The first such cryptosystem to use coding theory is the McEliece Cryptosystem [3]. This cryptosystem relies on a type of code called the Goppa code.

3.2.1 McEliece Cryptosystem

This cryptosystem is based on an error-correcting code that has a known and efficient decoding algorithm with an error-correcting capacity t. Here, t is sufficiently large to prevent brute-force attacks. The usage of Goppa codes (also used in the original proposal of the system) has withstood attacks. Suppose Bob wants to send a message to Alice, he can use the McEliece cryptosystem which has three main steps: key generation, encryption, and decryption as defined below. The cryptosystem is summarized in Table 3.1.

1. Key Generation: Alice selects a binary $[n, k]$ linear code $\mathcal{C}$ (say, a Goppa code) and fixes a generator matrix, G, for the code $\mathcal{C}$, which is not revealed. Code $\mathcal{C}$ has an efficient decoding algorithm, A. Alice chooses two other matrices: a non-singular $k \times k$ matrix S, and a random $n \times n$ permutation matrix P. Alice then publishes $G' = SGP$ along with the error-correcting capability(t) of $\mathcal{C}$ as her public key and keeps the parameters (S, P, A) as her private key.

2. Encryption: Bob encrypts his message, m using Alice's public key, (G', t) and a random n-bit vector z of weight t to get the ciphertext, $c := mG' + z$. Bob sends ciphertext c to Alice.

3. Decryption: Alice uses her private key to decrypt the ciphertext c and retrieve the message m. She first finds the inverse of P and S. Then computes $c' = cP^{-1} = mSG + zP^{-1}$. Given that P is a permutation matrix, zP^{-1} has the same weight as P, that is t. Alice now uses the decoding algorithm A to find a unique $m' = mS$. Finally, Alice computes $m = m'S^{-1}$.

3.2.2 Niederreiter Cryptosystem

This cryptosystem was introduced by H. Niederreiter in [4] and is a modified version of the McEliece cryptosystem. Suppose Bob wants to send a message to Alice, he can use the Niederreiter cryptosystem. It has three main steps: key generation, encryption, and decryption, as defined below. This cryptosystem is summarized in Table 3.2.

1. Key Generation: The public and private key generation techniques for the McEliece and Neiderreitter cryptosystems are similar, with 2 differences. First, instead of a generator matrix, Alice chooses a parity-check matrix $H_{(n-k)\times n}$ for the chosen linear code. Second, the chosen non-singular matrix is an $(n-k) \times (n-k)$ matrix instead of the $k \times k$ matrix chosen in the McEliece cryptosystem. Alice then computes $H' = SHP$, publishes (H', t) as her public key, and keeps (P, S, H) as her private key.

2. Encryption: Bob encrypts his message m of length n and weight t using Alice's public key H' to get the ciphertext, $c := H'm^\intercal = SHPm^\intercal$. Bob then sends the ciphertext c to Alice.

3. Decryption: Alice uses her private key to decode the ciphertext c. She first identifies a vector z satisfying, $Hz^\intercal = S^{-1}c$. Since $c = H'm^\intercal$ and $H' = SHP$,

$$\begin{aligned} Hz^\intercal = S^{-1}c &= HPm^\intercal \\ &= H(mP^\intercal)^\intercal \\ H(z - mP^\intercal)^\intercal &= 0. \end{aligned}$$

Therefore, $z - mP^\intercal \in \mathcal{C}$. Observe that $wt(mP^\intercal) = t$, since P is a permutation matrix. Alice can use Patterson's algorithm [5] on z to find $mP^\intercal$, and hence she can retrieve the message m using her knowledge of P.

The aforementioned steps of McEliece and Niederreiter cryptosystems are summarised in the Tables 3.1 and 3.2, respectively.

Alice:	Bob:
1. **Key Generation:** - Select a linear code $\mathcal{C}$, with error-correcting capacity t - Choose a generator matrix G - Compute $G' = SGP$ - Publish the public key: (G', t) - Private key: (S, P, A)	
	2. **Encryption:** - Choose z such that $len(z) = n$ and $wt(z) \leq t$ - Compute $c = mG' + z$ - Sends c to Alice
3. **Decryption:** - Compute P^{-1} & S^{-1} - Compute $c' = cP^{-1}$ - Use A to get $m' = mS$ - Compute $m = m'S^{-1}$	

Table 3.1: McEliece Cryptosystem

Alice:	Bob:
1. **Key Generation:** - Select a linear code $\mathcal{C}$, with error-correcting capacity t - Choose a parity-check matrix H - Compute $H' = SHP$ - Publish the public key: (H', t) - Private key: (S, P, H)	
	2. **Encryption:** - Message m: $len(m) = n$ and $wt_H(m) = t$ - Compute $c = H'm^\intercal$ - Send c to Alice
. 3. **Decryption:** - Compute P^{-1} & $S{-}1$ - Identify z such that $Hz^\intercal = S^{-1}c$ $(Hz^\intercal = S^{-1}c = H(mP^\intercal)^\intercal)$ - Use Patterson's algorithm to get $mP^\intercal$ - Compute m using P^{-1}	

Table 3.2: Niederreiter Cryptosystem

3.3 Variants of McEliece and Niederreiter cryptosystems

Both McEliece and Niederreiter cryptosystems are equivalent from a security point of view [6]. Some of the advantages of using the McEliece cryptosystem (over other public key cryptosystems, such as RSA) are: faster encryption and decryption, and resistance to quantum computing. But the hindrance to their implementation lies in the huge size of their public and private keys, by virtue of the large matrices present in the keys. Many advancements have been made in order to tackle this bottleneck by replacing the traditionally used Goppa codes with a suitable linear code that has a known efficient decoding algorithm. Some of these variants are discussed in this section.

3.3.1 Cryptosystems based on Reed-Solomon codes

A code-based cryptosystem based on generalized Reed Solomon codes in place of Goppa codes was introduced by H. Niederreiter in [4]. However, the cryptosystem is shown to be prone to *distinguisher attack* in [7]. C. Wieschebrink, in [8], used subcodes of GRS codes rather than a GRS code to develop a new cryptosystem. In this cryptosystem, a generator matrix G is disguised by the addition of r random column vectors in $\mathbb{F}^k$ at randomly chosen positions. Thereby obtaining a $k \times (n+r)$ dimensional matrix, G^*, which is published as the public key of this cryptosystem. The column vectors and the positions at which they are inserted in G are kept secret. Encryption of messages is carried out with error vectors of length $n + r$ and is otherwise similar to that in the classical McEliece cryptosystem. Decryption is done by first removing the corresponding positions from the ciphertext (w.r.t. the inserted positions in G^*) and then applying an efficient decoding algorithm. By computing the corresponding $(n + r - k) \times (n + r)$ dimensional parity-check matrix, H^* from G^*, we can develop a variant of the Niederreiter cryptosystem. The Wieschebrink cryptosystem is summarised in table 3.3.
However, it is straightforward to compute the dimension of Schur square of the underlying code and hence, the Wieschebrink cryptosystem is prone to the Sidelnikov-Shestakov attack [9].

3.3.2 Cryptosystems based on expanded Reed-Solomon codes

Another variant of the Neiderreiter cryptosystem based on expanded Reed-Solomon codes was presented by K. Khathuria et. al. in [10]. Variants of McEliece and Neiderreiter cryptosystems that aimed at reducing key size by replacing Goppa codes are prone to attacks as they fall short in hiding the algebraic structure of the underlying code. The cryptosystem in [10] hides the structure of the underlying code by first shortening the expanded Reed-Solomon code by puncturing it and then hiding the parity-check matrix of the shortened code, $\hat{H}_s$. That is, given a block diagonal matrix, T and a permutation matrix, P_σ, a matrix H' is computed as, $H' = \hat{H}_s Q$, where $Q := TP_\sigma$. H' is then published as the public key.
While this cryptosystem was shown to be secure against various structural attacks on cryptosystems based on GRS codes, A. Couvreur and M. Lequesne proved a way to attack the expanded Reed-Solomon code-based cryptosystem in [11].

Alice:	Bob:
1. **Key Generation:** - Select a Reed-Solomon code $\mathcal{C}$, with a generator matrix G - Randomly choose r column vectors $c_1, c_2, ..., c_r \in \mathbb{F}^k$ - Compute G^* by inserting $c_i's$ at random positions $p_i's$ of G - Publish the public key: G^* - Private key: $c_i's$ and $p_i's$	
	2. **Encryption:** - Choose e such that $len(e) = n + r$ and $wt(e) \leq t$ - Compute $c = mG^* + e$ - Send c to Alice
3. **Decryption:** - Delete corresponding $p_1, p_2, ..., p_r$ from c to get $c' = mG + e'$ - $wt(e') \leq t$ - Use decoding algorithm to get mG and hence, the message m	

Table 3.3: Wieschebrink cryptosystem

3.3.3 Cryptosystems based on twisted codes

Twisted codes have large Schur square dimensions. These codes can be used to develop cryptosystems that can withstand distinguisher attacks. Code-based cryptosystems based on twisted Reed-Solomon codes and twisted Hermitian codes were introduced, respectively, in [12] and [13]. The use of twisted codes in place of Goppa codes makes these cryptosystems differ from the original versions of McEliece/Niederreiter cryptosystems.

However, J. Lavauzelle and J. Renner in [14] presented cryptanalysis of twisted Reed-Solomon code-based cryptosystems. This attack is a possible threat to the twisted Hermitian code-based cryptosystem as well.

3.3.4 Cryptosystems based on quasi-cyclic codes

Another variant of the McEliece and Niederreiter cryptosystems based on quasi-cyclic codes was presented in [15] and [16], respectively. This variant is not only quantum secure but also offers enhanced performance in terms of transmission and encryption rates, along with significantly smaller key sizes. The smaller key size is obtained by the use of *circulant matrices*[1]since a circulant matrix can be described by its first row.

The proposed Niederreiter variant in [16] uses an $(m-1)/m$ quasi-cyclic code (ie, an $[mp, (m-1)p]$ code where p is a prime number and m is bounded above by a polynomial in p). This variant differs from the original Niederreiter cryptosystem in how the parity-check matrix is expressed. The $p \times mp$ parity-check matrix is expressed as follows:

$$H = [C_0 = I_p | C_1 | C_2 | ... | C_{m-1}],$$

where C_i is a circulant matrix of size p. Thus, we can denote H as $H = [I|C]$, where C is a concatenation of C_i's.

Remark. *Currently, this variant, based on the quasi-cyclic codes in place of the traditionally used Goppa codes, stands unbroken. Therefore, we study the quasi-cyclic codes in detail in the next chapter.*

[1]Circulant matrix is a square matrix such that every row (other than the first row) is a circular shift of the previous row.

Chapter 4

Quasi-cyclic codes

In this chapter, we discuss a Gröbner basis representation, spectral theory and an efficient decoding algorithm for quasi-cyclic codes as presented by K. Lally and P. Fitzpatrick in [20], by P. Semenov and P. Trifonov in [18] and by A. Zeh and S. Ling in [19]. A. Zeh and S. Ling in [19] propose a syndrome-based decoding algorithm. The error-correction capacity of this decoding algorithm is related to a new lower bound on the mimimum disatnce of quasi-cyclic codes.

4.1 Reduced Gröbner Basis

Consider the finite field $\mathbb{F}_q$ and a polynomial ring over this field, $\mathbb{F}_q[X]$. Recall from definition 2.4.8, that an $[n = ml, k, d]_q$ linear code $\mathcal{C}$ is said to be $l-$quasi-cyclic if it is invariant under cyclic shifts by l positions. We can consider the codewords in $\mathcal{C}$ as an $m \times l$ matrix in $\mathbb{F}_q^n$ as follows:

$$c = \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,l-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m-1,0} & c_{m-1,1} & \cdots & c_{m-1,l-1} \end{bmatrix}$$

Invariance of codewords under cyclic shifts now corresponds to the above matrix being closed under the shift of rows such that the last row replaces the first, and all the other $m-1$ rows are shifted downward by one row. Each column of this matrix can be mapped to a polynomial $c_i(X) = \sum_{j=0}^{m-1} c_{j,i} X^j, \ \ \forall i \in [l]$ in $\mathbb{F}_q[X]/\langle X^m - 1 \rangle$.

Let R denote $\mathbb{F}_q[X]/\langle X^m - 1\rangle$. Consider the following $R-$module isomorphism, where the matrix corresponding to a codeword is mapped to a polynomial in R^l:

$$\begin{array}{lccc} \phi: & \mathbb{F}_q^n & \longrightarrow & R^l \\ c = \begin{bmatrix} c_{0,0} & c_{0,1} & \dots & c_{0,l-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m-1,0} & c_{m-1,1} & \dots & c_{m-1,l-1} \end{bmatrix} & \longmapsto & \begin{array}{c} c(X) \\ \| \\ (c_0(X), c_1(X), \dots, c_{l-1}(X)) \end{array} \end{array} \tag{4.1}$$

The polynomial $c(X) \in R^l$ corresponds to $(c_0(X), c_1(X), \dots, c_{l-1}(X))$, where $c_i(X) = \sum_{j=0}^{m-1} c_{j,i} X^j$, $\forall i \in [l]$ is closed under multiplication by X and reduction modulo $X^m - 1$.

Therefore, a quasi-cyclic code can be seen as an $R-$submodule of the algebra R^l.

K. Lally and P. Fitzpatrick have proved in [20] that the generating set of any quasi-cyclic code can be expressed in the form of a reduced Gröbner basis with respect to the position-over-term order in $\mathbb{F}_q[X]^l$.

$$\tilde{G}(X) = \begin{pmatrix} g_{0,0}(X) & g_{0,1}(X) & \dots & g_{0,l-1}(X) \\ & g_{1,1}(X) & \dots & g_{1,l-1}(X) \\ & & \ddots & \vdots \\ & \mathbf{0} & & g_{l-1,l-1}(X) \end{pmatrix}$$

This $l \times l$ upper-triangular matrix $\tilde{G}(X)$ satisfying the properties listed below forms a reduced Gröbner basis of $\mathcal{C}$. Every codeword $c \in \mathcal{C}$ can be written as $c(X) = a(X)\tilde{G}(X)$. The dimension of $\mathcal{C}$ can be computed as $k = ml - \sum_{i=0}^{l-1} deg\big(g_{i,i}(X)\big) = \sum_{i=0}^{l-1} \Big(m - deg\big(g_{i,i}(X)\big)\Big)$.

1. $g_{i,j}(X) = 0, \ \forall 0 \leq j < i < l.$

2. $deg(g_{i,j}(X)) < deg(g_{j,j}(X)), \ \forall i < j.$

3. $g_{i,i}(X) | (X^m - 1), \ \forall i \in [l].$

4. If $g_{i,i} = X^m - 1,$ then $g_{i,j} = 0 \ \forall j \neq i.$

4.2 Spectral Analysis and Lower Bounds

4.2.1 Spectral theory of quasi-cyclic codes

Let $\tilde{G}(X)$ be the reduced Gröbner basis of an $[ml, k, d]_q$ $l-$quasi-cyclic code $\mathcal{C}$. Let $\alpha \in \mathbb{F}_{qr}$ be a primitive m^{th} root of unity, and r be the smallest positive integer such that $m|(q^r-1)$. The determinant of $\tilde{G}(X)$ is $det\big(\tilde{G}(X)\big) = \prod_{i=0}^{l-1} g_{i,i}(X)$. That means, the eigenvalues (λ_i) of $\tilde{G}(X)$ are the roots of the polynomials $g_{i,i}(X)$. Recall that $g_{i,i}(X)|(X^m - 1)$. Therefore, each λ_i can be written as a power of α $(i.e., \lambda_i = \alpha^{j_i})$. These eigenvalues of the Gröbner basis $\tilde{G}(X)$ are conventionally referred to as the eigenvalues of the quasi-cyclic code $\mathcal{C}$.

The greatest integer u_i such that $(X - \lambda_i)^{u_i}|det\big(\tilde{G}(X)\big)$ is called the *algebraic multiplicity* of λ_i. The *geometric multiplicity* of an eigenvalue λ_i is defined as the dimension of the right eigenspace $(\mathcal{V}_i)$ of the Gröbner basis, which is the solution space of $\tilde{G}(\lambda_i)\mathbf{v} = 0$. P. Semenov and P. Trifonov has shown in [18] that the algebraic multiplicity u_i for an eigenvalue λ_i is equal to its geometric multiplicity μ_i for the reduced Gröbner basis representation $\tilde{G}(X) \in \mathbb{F}_q[X]^{l\times l}$ of $\mathcal{C}$.

Consider the $u_i \times l$ matrix V_i constructed by stacking the vectors $\mathbf{v}_{i,j}^{\intercal}$, $0 \leq j < u_i$, for some basis $\mathbf{v}_{i,j}$ of the eigenspace $\mathcal{V}_i$. Let H_i denote the matrix corresponding to the eigenvalue λ_i, defined as the following Kronecker product: $H_i := (1, \lambda_i, \lambda_i^2, \ldots, \lambda_i^{m-1}) \bigotimes V_i$.
P. Semenov and P. Trifonov have proved in [18], that the matrix H obtained by stacking these matrices H_i corresponding to all the eigenvalues λ_i, $1 \leq i \leq t$ is a parity-check matrix of the quasi-cyclic code $\mathcal{C}$. They have used such a parity-check matrix over the extended field $\mathbb{F}_{q^r}$ in order to generalize the BCH bound to the quasi-cyclic case.

Since in the above construction of a parity-check matrix, the choice of different bases of eigenspaces $\mathcal{V}_i$ gives different equivalent parity-check matrices, we can consider an intersection of eigenspaces $\mathcal{V}$ corresponding to a number of eigenvalues. The minimum distance of the quasi-cyclic code is then given by the minimum distance of the cyclic code which has a generator polynomial with these eigenvalues as its roots.This forms the motivation behind the BCH-like and HT-like lower bounds for quasi-cyclic codes. These are discussed in detail in the next section.

4.2.2 Some lower bounds on minimum distance of quasi-cyclic codes

In this section, a BCH-like ([18, Thm. 2]) bound and an HT-like bound ([19, Thm. 1]) on quasi-cyclic codes are discussed. We begin by defining an eigencode. Let $[n] := \{0, \ldots, n-1\}$ for any integer n.

Definition 4.2.1. Eigencode: *Consider an eigenspace $\mathcal{V} \subseteq \mathbb{F}_q^l$. Then an $[l, k_{\mathbb{C}}, d_{\mathbb{C}}]_q$ eigencode $\mathbb{C}$ is defined as:*

$$\mathbb{C} := \left\{ c \in \mathbb{F}_q^l : \forall \mathbf{v} \in \mathcal{V}, \sum_{i=0}^{l-1} v_i c_i = 0 \right\}$$

If for some vector $\mathbf{v} \in \mathcal{V}$, its elements $v_1, v_2, \ldots, v_{l-1}$ are linearly independent over $\mathbb{F}_q$, then the corresponding eigencode $\mathbb{C} = (0, 0, \ldots, 0)$ and its minimum distance $d_{\mathbb{C}}$ is assumed to be infinity.

Theorem 4.2.1. *[18, Thm. 2] Let $\alpha \in \mathbb{F}_{q^r}$ be a primitive m^{th} root of unity. Consider the following eigenvalues of an $l-$quasi-cyclic code $\mathcal{C}$: for some $b \geq 0$, $\lambda_0 = \alpha^b, \lambda_1 = \alpha^{b+1}, \ldots, \lambda_{\delta-1} = \alpha^{b+\delta-1}$. Let $\mathcal{V}$ denote the intersection of the corresponding eigenspaces $\mathcal{V}_i$, i.e., $\mathcal{V} = \bigcap_{i=0}^{\delta-1} \mathcal{V}_i$ and let $\mathbb{C}$ be the eigencode given by $\mathcal{V}$.($\mathbb{C}$ is the direct sum of eigencodes $\mathbb{C}_i$ corresponding to $\mathcal{V}_i$.) Then the minimum distance of the quasi-cyclic code $\mathcal{C}$ is given by $d_{\mathcal{C}} \geq min(\delta, d_{\mathbb{C}})$.*

Theorem 4.2.2. *[19, Thm. 1] Consider an $[ml, k, d]$ quasi-cyclic code $\mathcal{C}$ and a primitive m^{th} root of unity $\alpha \in \mathbb{F}_{q^r}$. For some integers $f, \delta > 2, z > 0$ such that $gcd(m, z) = 1$, consider the following set:*

$$D := \Big\{ \begin{array}{cccc} f, & f+z, & \ldots & f+(\delta-2)z, \\ f+1, & f+1+z, \ldots & & f+1+(\delta-2)z, \\ \vdots & \vdots & \ldots & \vdots \\ f+v, & f+v+z, \ldots & & f+v+(\delta-2)z \end{array} \Big\}$$

Let the eigenvalues be $\lambda_i = \alpha^i$, $\forall i \in D$ and the corresponding eigenspaces be $\mathcal{V}_i$, $\forall i \in D$. Consider the intersection of these eigenspaces to be $\mathcal{V} := \bigcap_{i \in D} \mathcal{V}_i$ and the eigencode corresponding to this intersection as $\mathbb{C}$. Let $\mathbf{v} \in \mathcal{V}$ be an eigenvector such that the following

condition holds for all codeword polynomials $c(X) = (c_0(X), c_1(X), ..., c_{l-1}(X)) \in \mathcal{C}$:

$$\sum_{i=0}^{\infty} c(\alpha^{f+zi+j}) \cdot \mathbf{v} X^i \equiv 0 \mod X^{\delta-1}, \ \forall j \in [v+1] \tag{4.2}$$

Then the minimum distance of the quasi-cyclic code $\mathcal{C}$ *is given by* $d \geq min(\delta + v, d_{\mathbb{C}})$.

4.3 Syndrome-based Decoding Algorithm

In this section, we discuss the decoding algorithm described in [19], which can correct up to $\frac{d^*-1}{2}$ errors (where $d^* = min(\delta + v, d_{\mathbb{C}})$). Consider an $[ml, k, d]_q$ $l-$quasi-cyclic code. Let $r(X)$ be the polynomial corresponding to the received word.

$$\begin{aligned} r(X) =& \big(r_0(X), r_1(X), ..., r_{l-1}(X)\big) \\ =& \big(c_0(X) + e_0(X), c_1(X) + e_1(X), ..., c_{l-1}(X) + e_{l-1}(X)\big) \end{aligned}$$

where, $e_i(X)$ $\forall i \in [l]$ are the error polynomials, defined as,

$$e_i(X) := \sum_{j \in \mathcal{E}_i} e_{i,j} X^j, \ \forall i \in [l].$$

Let the set of error locations be given by $\mathcal{E} := \bigcup_{i=0}^{l-1} \mathcal{E}_i$ such that $|\mathcal{E}| = \varepsilon$. Note that $\varepsilon \leq \frac{d^*-1}{2}$.

Consider an intersection of eigenspaces, $\mathcal{V} := \bigcap_{i \in [\delta-1], j \in [v+1]} \mathcal{V}_{f+iz+j}$. Let $\mathbf{v} = (v_0, v_1, \ldots, v_{l-1}) \in \mathcal{V}$ be an eigenvector. Then the $v+1$ syndrome polynomials in $\mathbb{F}_{q^r}[X]$ are defined as:

$$\begin{aligned} S_t(X) &:= \sum_{i=0}^{\infty} \left(\sum_{j=0}^{l-1} r_j(\alpha^{f+iz+t}) v_j \right) X^i \mod X^{\delta-1} \\ &= \sum_{i=0}^{\delta-2} \left(\sum_{j=0}^{l-1} r_j(\alpha^{f+iz+t}) v_j \right) X^i, \ \forall t \in [v+1] \end{aligned} \tag{4.3}$$

Using the equation 4.2 from theorem 4.2.2, we can simplify $S_t(X)$ to get,

$$S_t(X) = \sum_{i=0}^{\delta-2} \left(\sum_{j=0}^{l-1} e_j(\alpha^{f+iz+t}) v_j \right) X^i, \ \ \forall t \in [v+1] \tag{4.4}$$

Using the Generalised Extended Euclidean Algorithm (GEEA) in the decoding algorithm, we get the error locator polynomial to be a polynomial which has $\alpha^{-iz}, \forall i \in \mathcal{E}$ as its roots. Therefore, we can define the *error locator polynomial* as follows:

$$\Lambda(X) := \prod_{i \in \mathcal{E}} (1 - X\xi^{in_1}) \tag{4.5}$$

Let $\Omega_t(X), \forall t \in [v+1]$ denote the *error evaluator polynomials*. It is defined as follows:

$$\Lambda(X) \cdot S_t(X) \equiv \Omega_t(X) \mod X^{\delta-1}, \ \ \forall t \in [v+1] \tag{4.6}$$

These $v+1$ equations form the set of *Key Equations*. In order to solve these, GEEA is being used in the decoding algorithm.

Table 4.1 ([19]) summarises the syndrome-based decoding algorithm for quasi-cyclic codes.

Decoding Algorithm for an $[ml, k, d]_q$-quasi-cyclic code

Input:
$m, l, k, q, r \longleftarrow$ Parameters of the quasi-cyclic code $\mathcal{C}$
$r(X) = \big(r_0(X), r_1(X), ..., r_{l-1}(X)\big) \in \mathbb{F}_q[X]^l \longleftarrow$ Received word
$f, v \geq 0, \delta > 2, z > 0$ with $gcd(z, m) = 1$
$\lambda_i = \alpha^{f+iz+j}$, $\forall i \in [\delta - 1], j \in [v + 1] \longleftarrow$ Eigenvalues
$\mathbf{v} = (v_0, v_1, ..., v_{l-1}) \in \mathbb{F}_{q^r}^l \longleftarrow$ Eigenvector

Output:
$c(X) = (c_0(X), c_1(X), ..., c_{l-1}(X)) \longleftarrow$ Estimated Codeword
or
DECODING FAILURE

Algo:
- Compute $S_t(X), \forall t \in [v + 1]$ as in equation 4.3
- Solve the Key Equations jointly by applying the Generalised Extended Euclidean Algorithm on $(X^{\delta-1}, S_0(X), S_1(X), ..., S_v(X))$
- Find all i_k such that $\Lambda(\alpha^{-i_k z}) = 0 \implies \mathcal{E} = \{i_1, i_2, ..., i_\varepsilon\}$
- If $\varepsilon < deg(\Lambda(X))$:
  DECODING FAILURE $\longleftarrow$ **Output**
  else:
  Compute error values $E_{i_1}, E_{i_2}, ..., E_{i_\varepsilon} \in \mathbb{F}_{q^r}$ from one of the error evaluator polynomials, $\Omega_t(X), t \in [v + 1]$
  Compute $e_{i_k,0}, e_{i_k,1}, ..., e_{i_k,l-1} \in \mathbb{F}_q$ s.t. $E_{i_k} := \sum_{j=0}^{l-1} e_{i_k j} v_j, \ \forall i_k \in \mathcal{E}$
  Compute $e_j(X) = \sum_{i \in \mathcal{E}_j} e_{i,j} X^i, \ \forall j \in [l]$
  Compute $c_j(X) = r_j(X) - e_j(X), \ \forall j \in [l]$
  $c(X) = (c_0(X), c_1(X), ..., c_{l-1}(X)) \longleftarrow$ **Output**

Table 4.1: Decoding Algorithm (taken from [19])

Chapter 5

Decoding quasi-twisted codes

In this chapter, more details on quasi-twisted codes are discussed, including a Hartmann-Tzeng(HT)-like bound on the minimum distance of quasi-twisted codes in section 5.2. In the section 5.3, we describe a syndrome-based decoding algorithm of quasi-twisted codes that can decode up to $\varepsilon \leq \frac{d^*-1}{2}$ errors, where $d^* = min\big(\delta + s, d_{\mathbb{C}}\big)$.

5.1 Spectral theory of quasi-twisted codes

We analyze the spectral theory of quasi-twisted codes in this section. A detailed explanation of the same can be found in [21].

Recall from definition 2.4.9, that an $[n = ml, k, d]_q$ linear code $\mathcal{C}$ is said to be $(\lambda, l)-$quasi-twisted if it is invariant under $\lambda-$constashift by l positions. We can consider a similar isomorphism as in eqn. 4.1, with $R := \mathbb{F}_q[X]/\langle X^m - \lambda \rangle$. Thereby, the invariance of the codewords in $\mathbb{F}_q^{ml}$ under $\lambda-$constashift by l positions now corresponds to the matrix in $\mathbb{F}_q^{m \times l}$ being closed under the row $\lambda-$constashift. That means the polynomials $c(X) \in R^l$ are closed under component-wise multiplication by X and reduction modulo $X^m - \lambda$. So, a $(\lambda, l)-$quasi-twisted code can be seen as an $R-$submodule of R^l.

M. F. Ezerman et. al. has shown in [21] that a generating set of quasi-twisted codes can also be expressed in the form of a *reduced Gröbner basis* with respect to the position-over-

term order. Thus, for every quasi-twisted code, its reduced Gröbner basis $\tilde{G}(X)$ is of the following form and satisfies the properties listed below.

$$\tilde{G}(X) = \begin{pmatrix} g_{0,0}(X) & g_{0,1}(X) & \dots & g_{0,l-1}(X) \\ & g_{1,1}(X) & \dots & g_{1,l-1}(X) \\ & & \ddots & \vdots \\ & \mathbf{0} & & g_{l-1,l-1}(X) \end{pmatrix}$$

1. $g_{i,j}(X) = 0, \ \forall 0 \leq j < i < l.$
2. $deg(g_{i,j}(X)) < deg(g_{j,j}(X)), \ \forall i < j.$
3. $g_{i,i}(X) | (X^m - \lambda), \ \forall i \in [l].$
4. If $g_{i,i}(X) = X^m - \lambda, \ \text{then } g_{i,j}(X) = 0 \ \forall j \neq i.$

Every codeword polynomial $c(X) \in \mathcal{C}$ can then be written as $c(X) = a(X)\tilde{G}(X)$. The $\mathbb{F}_q-$dimension of $\mathcal{C}$ is given by,

$$k = ml - \sum_{i=0}^{l-1} deg\big(g_{i,i}(X)\big) = \sum_{i=0}^{l-1} \Big(m - deg\big(g_{i,i}(X)\big)\Big).$$

The determinant of $\tilde{G}(X)$ is $det\big(\tilde{G}(X)\big) = \prod_{i=0}^{l-1} g_{i,i}(X)$, which means that every eigenvalue of the quasi-twisted code $\mathcal{C}$, β_i is a root of the polynomial $g_{i,i}(X)$, for some $i \in [l]$. From the aforementioned property 3, each eigenvalue $\beta_i, \forall i \in [m]$ can be written as $\beta_i = \alpha\xi^i$, where α is a fixed m^{th} root of λ and ξ is a primitive m^{th} root of unity. The *algebraic multiplicity* of an eigenvalue β_i is defined as the largest integer, a such that $(X - \beta_i)^a | det\big(\tilde{G}(X)\big)$. Its *geometric multiplicity* is the dimension of the null space of $\tilde{G}(\beta_i)$, which is the eigenspace of β_i. i.e., $\mathcal{V}_i := \{\mathbf{v} \in \mathbb{F}^l : \tilde{G}(\beta_i)\mathbf{v}^\intercal = \mathbf{0}\}$. Here $\mathbb{F}$ is the splitting field of $X^m - \lambda$ (i.e., the smallest extension of $\mathbb{F}_q$ containing all the roots of $X^m - \lambda$). It is shown in [21] that the algebraic multiplicity of an eigenvalue of a $(\lambda, l)-$quasi-twisted code is equal to its geometric multiplicity.

5.2 HT-like bound on the minimum distance of quasi-twisted codes

Consider an $[m \cdot l, k, d]_q$ (λ, l)-quasi-twisted code, say $\mathcal{C}$. Let ξ be a primitive m^{th} root of unity and α be an m^{th} root of λ so that $\Omega := \{\alpha\xi^i : 0 \leq i \leq m-1\}$ forms the set of all m^{th} roots of λ or equivalently the set of roots of $X^m - \lambda$. Recall that for $l = 1$, we get a $\lambda-$constacyclic code. Given the generator polynomial $g(X)$ of a $\lambda-$constacyclic code, the set $L := \{\alpha\xi^i : g(\alpha\xi^i) = 0\}$ is called its *defining set*.

Let $\mathbb{C}$ be an eigencode corresponding to an eigenspace $\mathcal{V}$ as stated in definition 4.2.1. We now re-state a spectral bound on the minimum distance of quasi-twisted codes, which is presented in [21, Theorem 11].

Theorem 5.2.1. *[21, Theorem 11] Consider a $(\lambda, l)-$quasi-twisted code $\mathcal{C}$. Let $\overline{\Omega} \subset \Omega$ denote the non-empty set of eigenvalues of $\mathcal{C}$. Consider a non-empty subset of eigenvalues, $P \subseteq \overline{\Omega}$ and a $\lambda-$constacyclic code $\mathcal{C}_P$ with its defining set $L \supseteq P$. Let d_P denote a lower bound on the minimum distance of $\mathcal{C}_P$. Consider the intersection of eigenspaces ($\mathcal{V}_\beta$) of the eigenvalues (β) in P, $\mathcal{V}_P := \bigcap_{\beta \in P} \mathcal{V}_\beta$ and the corresponding eigencode $\mathbb{C}_P$. Then, the minimum distance of the $(\lambda, l)-$quasi-twisted code $\mathcal{C}$ is given by, $d_\mathcal{C} \geq min\{d_P, d_{\mathbb{C}_P}\}$.*

In theorem 5.2.2, we formulate the HT-like bound for quasi-twisted codes. This bound can be deduced from [21, Remark 5], following the motivation behind the HT-like bound for quasi-cyclic codes (theorem 4.2.2). However, to the best of our knowledge, this is the first instance of a formal proof for this bound.

Theorem 5.2.2. ***(HT-like Bound)*** *Consider an $[m \cdot l, k, d]_q$ (λ, l)-quasi-twisted code $\mathcal{C}$. For positive integers a, n_1, n_2, s and $\delta \geq 2$ with $gcd(m, n_1) = 1$ and $gcd(m, n_2) < \delta$, define the set,*

$$
\begin{aligned}
D := \{&a, a+n_1, ..., a+(\delta-2)n_1,\\
&a+n_2, a+n_2+n_1, ..., a+n_2+(\delta-2)n_1,\\
&\vdots \qquad\quad \vdots \qquad \ldots \qquad \vdots\\
&a+sn_2, a+sn_2+n_1, ..., a+sn_2+(\delta-2)n_1\}
\end{aligned}
$$

For some $a \geq 0$, the eigenvalues $\beta_i = \alpha\xi^i, \forall i \in D$. Let the corresponding eigenspaces be denoted by $\mathcal{V}_i, \forall i \in D$. Consider the intersection of these eigenspaces, $\mathcal{V} := \bigcap_{i \in D} \mathcal{V}_i$ and an eigenvector $\mathbf{v} = (v_0, v_1, ..., v_{l-1}) \in \mathcal{V}$. Let the eigencode given by $\mathcal{V}$ (that is, the direct sum

of eigencodes defined by $\mathcal{V}_i$) be $\mathbb{C}$ with minimum distance $d_{\mathbb{C}}$. If

$$c(\alpha\xi^{a+i_1n_1+i_2n_2})\mathbf{v}^\intercal = 0, \ \ \forall i_1 \in [\delta-1], i_2 \in [s+1] \tag{5.1}$$

holds true for all $c(X) = \big(c_0(X), c_1(X), ..., c_{l-1}(X)\big) \in \mathcal{C}$, then the HT-like bound for the quasi-twisted code $\mathcal{C}$ is $d \geq d^* = min\big\{\delta + s, d_{\mathbb{C}}\big\}$.

Proof.
Recall that an eigenspace $\mathcal{V}_i$ of an eigenvalue β_i is defined as,

$$\mathcal{V}_i := \Big\{\mathbf{v} \in \mathbb{F}^l : \tilde{G}(\beta_i)\mathbf{v}^\intercal = \mathbf{0}\Big\}.$$

Since $\mathcal{V}$ is the intersection of the eigenspaces given by $\mathcal{V} := \bigcap_{i\in D} \mathcal{V}_i$, we can conclude that

$$\tilde{G}(\beta_i)\mathbf{v}^\intercal = \mathbf{0}, \ \ \forall i \in D \text{ and } \forall \mathbf{v} \in \mathcal{V} \tag{5.2}$$

Recall that any codeword $C(X) \in \mathcal{C}$ can be written as $c(X) = a(X)\tilde{G}(X)$. Now, the LHS of the given condition 5.1 can be simplified as,

$$\begin{aligned} c(\alpha\xi^{a+i_1n_1+i_2n_2})\mathbf{v}^\intercal =& c(\alpha\xi^i)\mathbf{v}^\intercal, \ \ \forall i \in D \\ =& a(\alpha\xi^i))\tilde{G}(\alpha\xi^i)\mathbf{v}^\intercal, \ \ \forall i \in D \\ =& 0 \text{ (using equation 5.2)} \end{aligned}$$

Let $\overline{\Omega}$ denote the set of all eigenvalues of the $(\lambda, l)-$quasi-twisted code $\mathcal{C}$. Note that this set can be written as $\overline{\Omega} = \{\beta_i : i \in D\}$. In other words, the premise of this theorem can be considered as a special case of theorem 5.2.1, where $P = \overline{\Omega}$. Therefore, the minimum distance of the $(\lambda, l)-$quasi-twisted code $\mathcal{C}$ is $d_{\mathcal{C}} \geq min\{d_{\overline{\Omega}}, d_{\mathbb{C}_{\overline{\Omega}}}\}$, where we have the following observations on $d_{\overline{\Omega}}$ and $d_{\mathbb{C}_{\overline{\Omega}}}$.

[22, Corollary 2.ii] gives a bound on the minimum distance of the $\lambda-$constacyclic code, therefore we have $d_{\overline{\Omega}} \geq \delta + s$.

Since $\overline{\Omega} = \{\beta_i : i \in D\}, \mathbb{C}_{\overline{\Omega}}$ is the same as $\mathbb{C}$, which is the eigencode corresponding to the common eigenspace of the eigenvalues in $\overline{\Omega}$.

Therefore, we can conclude that the minimum distance of a $(\lambda, l)-$quasi-twisted code $\mathcal{C}$ is given by, $d \geq d^* = min\big\{\delta + s, d_{\mathbb{C}}\big\}$. ∎

5.3 Syndrome-based decoding of quasi-twisted codes

In this section, we describe some polynomials and matrices that are necessary for the decoding procedure detailed in the section 5.4.

Consider an $[m \cdot l, k, d]_q$ (λ, l)-quasi-twisted code $\mathcal{C}$. Let $r(X)$ be the received word and $\mathcal{E} = \{i_1, i_2, ..., i_\varepsilon\}$ be the set of error locations such that,

$$\begin{aligned} r(X) &= \big(r_0(X), r_1(X), ..., r_{l-1}(X)\big) \\ &= \big(c_0(X) + e_0(X), c_1(X) + e_1(X), ..., c_{l-1}(X) + e_{l-1}(X)\big) \end{aligned} \tag{5.3}$$

where

$$c_j(X) = \sum_{i \in [m]} c_{i,j} X^i \text{ and } e_j(X) = \sum_{i \in \mathcal{E}_i} e_{i,j} X^i, \ \ \forall j \in [l].$$

$$\mathcal{E} := \bigcup_{i=0}^{l-1} \mathcal{E}_i \tag{5.4}$$

$$\implies |\mathcal{E}| = \varepsilon \leq \sum_{i=0}^{l-1} |\mathcal{E}_i|$$

Syndrome Polynomials: Let $\mathcal{V} = \bigcap_{i_1 \in [\delta-1], i_2 \in [s+1]} \mathcal{V}_{a+i_1 n_1 + i_2 n_2}$. Consider an eigenvector $\mathbf{v} = (v_0, v_1, \ldots, v_{l-1}) \in \mathcal{V}$. We consider $(s+1)$ syndrome polynomials defined as follows:

$$\begin{aligned} S_t(X) &:= \sum_{i=0}^{\infty} \left(\sum_{j=0}^{l-1} r_j (\alpha \xi^{a+in_1+tn_2}) v_j \right) X^i \mod X^{\delta-1} \\ &= \sum_{i=0}^{\delta-2} \left(\sum_{j=0}^{l-1} r_j (\alpha \xi^{a+in_1+tn_2}) v_j \right) X^i, \ \ \forall t \in [s+1] \end{aligned} \tag{5.5}$$

Using the equation 5.1 from theorem 5.2.2, we can simplify $S_t(X)$ to get,

$$S_t(X) = \sum_{i=0}^{\delta-2} \left(\sum_{j=0}^{l-1} e_j (\alpha \xi^{a+in_1+tn_2}) v_j \right) X^i, \ \ \forall t \in [s+1] \tag{5.6}$$

Error locator polynomial: We use the generalized Berlekamp-Massey algorithm [24] in

the decoding procedure. Using this algorithm, we get the error locator polynomial to be a polynomial with $\xi^{-in_1}, \forall i \in \mathcal{E}$ as its roots. Therefore, we define the error locator polynomial as follows:

$$\Lambda(X) := \prod_{i \in \mathcal{E}} (1 - X\xi^{in_1}) \tag{5.7}$$

Error evaluator polynomials: Let $\Omega_t(X), \forall t \in [s+1]$ denote the error evaluator polynomials. It is defined as follows:

$$\Lambda(X) \cdot S_t(X) \equiv \Omega_t(X) \mod X^{\delta-1}, \ \ \forall t \in [s+1] \tag{5.8}$$

Remark 5.3.1. *These $(s+1)$ equations in 5.8 form the 'Key Equations'. Solving these equations jointly is equivalent to applying the generalized Berlekamp-Massey algorithm (described in [24]) or the generalized expanded Euclidean algorithm (described in [25]) on $(X^{\delta-1}, S_0(X), S_1(X), ..., S_s(X))$. We use the former one in our decoding algorithm.*

We now consider the matrix representation of the syndrome polynomial and a decomposition of this matrix. More detailed background can be found in [24, Section VI.A]. The decomposition of the syndrome matrix is done in accordance with a matrix operation $*$, which was introduced in [26]. These will be used in the decoding procedure discussed in the next section.

$$\begin{aligned}
S :=& \big(S^{\langle 0 \rangle} \ \ S^{\langle 1 \rangle} \ \ ... \ \ S^{\langle s \rangle}\big)^\intercal, \\
&\text{where } \ S^{\langle t \rangle} := \big(S^{\langle t \rangle}_{i+j}\big)^{j \in [\varepsilon+1]}_{i \in [\delta-1-\varepsilon]}, \ \ \forall t \in [s+1] \\
&\text{and } \ S^{\langle t \rangle}_k = \sum_{j=0}^{l-1} r_j (\alpha \xi^{a+kn_1+tn_2}) v_j, \ \ \forall k \in [\delta-1], t \in [s+1]; \\
X :=& \big(X^{\langle 0 \rangle} \ \ X^{\langle 1 \rangle} \ \ ... \ \ X^{\langle s \rangle}\big)^\intercal, \\
&\text{where } \ X\langle t \rangle := \big(\xi^{(n_1 i + n_2 t) j}\big)^{j \in \{1,2,...,\varepsilon\}}_{i \in [\delta-1-\varepsilon]} \ \ \forall t \in [s+1]; \\
Y :=& diag\big((\alpha\xi^a)^{i_1} E_{i_1}, (\alpha\xi^a)^{i_2} E_{i_2}, ..., (\alpha\xi^a)^{i_\varepsilon} E_{i_\varepsilon}\big), \\
&\text{where } \ E_{i_k} := \sum_{j=0}^{l-1} e_{i_k j} v_j \ \ \forall i_k \in \mathcal{E};
\end{aligned}$$

$$\tilde{X} := \big(\xi^{in_1 j}\big)^{j \in [\varepsilon+1]}_{i \in \{1,2,...,\varepsilon\}}; \qquad A := \big(\xi^{ij}\big)^{j \in 1,2,...,\varepsilon}_{i \in [s+1]}; \qquad B := X^{\langle 0 \rangle}$$

These are explicitly written below:

$$S = \begin{bmatrix} S_0^{\langle 0\rangle} & S_1^{\langle 0\rangle} & \dots & S_{\varepsilon}^{\langle 0\rangle} \\ S_1^{\langle 0\rangle} & S_2^{\langle 0\rangle} & \dots & S_{\varepsilon+1}^{\langle 0\rangle} \\ \vdots & \vdots & \dots & \vdots \\ S_{\delta-2-\varepsilon}^{\langle 0\rangle} & S_{\delta-1-\varepsilon}^{\langle 0\rangle} & \dots & S_{\delta-2}^{\langle 0\rangle} \\ \hdashline S_0^{\langle 1\rangle} & S_1^{\langle 1\rangle} & \dots & S_{\varepsilon}^{\langle 1\rangle} \\ S_1^{\langle 1\rangle} & S_2^{\langle 1\rangle} & \dots & S_{\varepsilon+1}^{\langle 1\rangle} \\ \vdots & \vdots & \dots & \vdots \\ S_{\delta-2-\varepsilon}^{\langle 1\rangle} & S_{\delta-1-\varepsilon}^{\langle 1\rangle} & \dots & S_{\delta-2}^{\langle 1\rangle} \\ \hdashline \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \hdashline S_0^{\langle s\rangle} & S_1^{\langle s\rangle} & \dots & S_{\varepsilon}^{\langle s\rangle} \\ S_1^{\langle s\rangle} & S_2^{\langle s\rangle} & \dots & S_{\varepsilon+1}^{\langle s\rangle} \\ \vdots & \vdots & \dots & \vdots \\ S_{\delta-2-\varepsilon}^{\langle s\rangle} & S_{\delta-1-\varepsilon}^{\langle s\rangle} & \dots & S_{\delta-2}^{\langle s\rangle} \end{bmatrix}_{(\delta-1-\varepsilon)(s+1)\times(\varepsilon+1)} \tag{5.9}$$

$$X = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \xi^{n_1} & \xi^{2n_1} & \dots & \xi^{\varepsilon n_1} \\ \vdots & \vdots & \dots & \vdots \\ \xi^{(\delta-2-\varepsilon)n_1} & \xi^{2(\delta-2-\varepsilon)n_1} & \dots & \xi^{\varepsilon(\delta-2-\varepsilon)n_1} \\ \hdashline \xi^{n_2} & \xi^{2n_2} & \dots & \xi^{\varepsilon n_2} \\ \xi^{(n_1+n_2)} & \xi^{2(n_1+n_2)} & \dots & \xi^{\varepsilon(n_1+n_2)} \\ \vdots & \vdots & \dots & \vdots \\ \xi^{\left((\delta-2-\varepsilon)n_1+n_2\right)} & \xi^{2\left((\delta-2-\varepsilon)n_1+n_2\right)} & \dots & \xi^{\varepsilon\left((\delta-2-\varepsilon)n_1+n_2\right)} \\ \hdashline \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \hdashline \xi^{sn_2} & \xi^{2sn_2} & \dots & \xi^{\varepsilon sn_2} \\ \xi^{(n_1+sn_2)} & \xi^{2(n_1+sn_2)} & \dots & \xi^{\varepsilon(n_1+sn_2)} \\ \vdots & \vdots & \dots & \vdots \\ \xi^{\left((\delta-2-\varepsilon)n_1+sn_2\right)} & \xi^{2\left((\delta-2-\varepsilon)n_1+sn_2\right)} & \dots & \xi^{\varepsilon\left((\delta-2-\varepsilon)n_1+sn_2\right)} \end{bmatrix}_{(\delta-1-\varepsilon)(s+1)\times\varepsilon} \tag{5.10}$$

$$Y = \begin{bmatrix} (\alpha\xi^a)^{i_1}E_{i_1} & & & \\ & (\alpha\xi^a)^{i_2}E_{i_2} & & \\ & & \ddots & \\ & & & (\alpha\xi^a)^{i_\varepsilon}E_{i_\varepsilon} \end{bmatrix}_{\varepsilon\times\varepsilon} \tag{5.11}$$

$$\tilde{X} = \begin{bmatrix} 1 & \xi^{n_1} & \xi^{2n_1} & \dots & \xi^{\varepsilon n_1} \\ 1 & \xi^{2n_1} & \xi^{4n_1} & \dots & \xi^{2\varepsilon n_1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \xi^{\varepsilon n_1} & \xi^{2\varepsilon n_1} & \dots & \xi^{\varepsilon^2 n_1} \end{bmatrix}_{\varepsilon\times(\varepsilon+1)} \qquad A = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \xi^{n_2} & \xi^{2n_2} & \dots & \xi^{\varepsilon n_2} \\ \vdots & \vdots & \dots & \vdots \\ \xi^{sn_2} & \xi^{2sn_2} & \dots & \xi^{\varepsilon sn_2} \end{bmatrix}_{(s+1)\times\varepsilon} \tag{5.12, 5.13}$$

$$B = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \xi^{n_1} & \xi^{2n_1} & \dots & \xi^{\varepsilon n_1} \\ \vdots & \vdots & \dots & \vdots \\ \xi^{(\delta-2-\varepsilon)n_1} & \xi^{2(\delta-2-\varepsilon)n_1} & \dots & \xi^{\varepsilon(\delta-2-\varepsilon)n_1} \end{bmatrix}_{(\delta-1-\varepsilon)\times\varepsilon} \tag{5.14}$$

5.4 Decoding Algorithm up to the HT-like bound

In this section, we develop a syndrome-based decoding procedure for quasi-twisted codes. This algorithm (Table 5.1) can be used to correct up to $\varepsilon = \frac{d^*-1}{2}$ errors, where $d^* = min(\delta + s, d_{\mathbb{C}})$ is the HT-like bound for quasi-twisted codes. We begin by proving a condition which ensures that the decoding algorithm can correct up to ε errors. We then analyze the time complexity of the algorithm to gauge its efficiency.

Theorem 5.4.1. *Consider an $[m\cdot l, k, d]_q$ (λ, l)-quasi-twisted code $\mathcal{C}$ such that the condition 5.1 along with the assumptions in theorem 5.2.2 hold. Let the number of errors be at most ε such that $\mathcal{E} = \{i_1, i_2, ..., i_\varepsilon\}$ is the set of error locations. Consider the (s+1) syndrome polynomials $S_0(X), S_1(X), ..., S_s(X)$ as defined in 5.5. Then, the syndrome matrix (refer 5.9) will have $rank(S) = \varepsilon$.*

Proof.
In the previous section, we have defined the syndrome matrix S and the matrices $X, Y, \tilde{X}$ (refer eqns. 5.9 to 5.10) such that we get the decomposition $S = XY\tilde{X}$. $\tilde{X}$ being a Vandermonde matrix and $gcd(m, n_1) = 1$ ensures that the $rank(\tilde{X}) = min(\varepsilon, \varepsilon + 1) = \varepsilon$. The diagonal matrix, Y is non-singular. Therefore, we see that $rank(S) = rank(X)$.

According to the $*$ operation defined in [26], the matrix X can be decomposed as $X = A * B$ where A and B are defined in 5.13 and 5.14. Then, using [24, Section VI.A], we see that if $rank(A) + rank(B) > \varepsilon$, then $rank(X) = \varepsilon$. Notice that both A and B are Vandermonde matrices and $gcd(m, n_1) = 1$. Therefore, $rank(A) = min((s + 1), \varepsilon)$ and $rank(B) = min((\delta - 1 - \varepsilon), \varepsilon)$. It is shown in [24] that we can assume $\delta - 1 > s$, for the case of HT-bound. Recall that $\varepsilon \leq \frac{d^* - 1}{2}$, where $d^* = min(\delta + s, d_{\mathbb{C}})$. Then, we get

$$\varepsilon \leq \frac{d^* - 1}{2} = \frac{min(\delta + s, d_{\mathbb{C}}) - 1}{2} \leq \frac{\delta + s - 1}{2} < \delta - 1.$$

Therefore, we can simplify the four cases of $rank(A) + rank(B)$ as follows:

$$\begin{aligned}
&(s + 1) + (\delta - 1 - \varepsilon) = (\delta + s - 1) + 1 - \varepsilon > 2\varepsilon + 1 - \varepsilon = \varepsilon + 1 > \varepsilon,\\
&(s + 1) + \varepsilon > \varepsilon,\\
&\varepsilon + (\delta - 1 - \varepsilon) = \delta - 1 > \varepsilon,\\
&\varepsilon + \varepsilon > \varepsilon
\end{aligned}$$

Note that $rank(A) + rank(B) > \varepsilon$ in all the four cases. Thus we get $rank(X) = \varepsilon$. Therefore, $rank(S) = rank(X) = \varepsilon$.

■

Remark 5.4.1. *Note that the condition $rank(S) = \varepsilon$ guarantees that the error locator polynomial generated is unique with roots $\xi^{-in_1}, \forall i \in \mathcal{E}$. Thus, we can perform the decoding algorithm as described in the following table 5.1, provided the number of errors is upper-bounded by ε.*

Table 5.1 describes the decoding algorithm for an $[m \cdot l, k, d]_q$ (λ, l)-quasi-twisted code $\mathcal{C}$.

Decoding Algorithm for a (λ, l)-QT code

Input:
$\lambda, m, l, k, q, r \longleftarrow$ parameters of the quasi-twisted code $\mathcal{C}$
$r(X) = \big(r_0(X), r_1(X), ..., r_{l-1}(X)\big) \in \mathbb{F}_q[X]^l \longleftarrow$ received word
$a \geq 0, \delta \geq 2, \{s, n_1, n_2\} \in \mathbb{Z}^+$ with $gcd(m, n_1) = 1, gcd(m, n_2) < \delta$
$\beta = \alpha\xi^{a+in_1+jn_2}$, $\forall i \in [\delta - 1], j \in [s + 1] \longleftarrow$ eigenvalues
$\mathbf{v} = (v_0, v_1, ..., v_{l-1}) \in \mathbb{F}_{q^r}^l \longleftarrow$ eigenvector

Output:
$c(X) = (c_0(X), c_1(X), ..., c_{l-1}(X)) \longleftarrow$ estimated codeword
or
DECODING FAILURE

Algo:
- Compute $S_t(X), \forall t \in [s + 1]$ as in eqn. 5.5
- Solve the Key Equations jointly by applying the generalized Berlekamp-Massey algorithm on $(X^{\delta-1}, S_0(X), S_1(X), ..., S_s(X))$ (refer Remark. 5.3.1)
- Find all i_k such that $\Lambda(\xi^{-i_k n_1}) = 0 \implies \mathcal{E} = \{i_1, i_2, ..., i_\varepsilon\}$
- If $\varepsilon < deg(\Lambda(X))$:
  DECODING FAILURE $\longleftarrow$ **Output**
  else:
  Compute $E_{i_1}, E_{i_2}, ..., E_{i_\varepsilon} \in \mathbb{F}_{q^r}$ from one of the error evaluator polynomials, $\Omega_t(X), t \in [s + 1]$
  Compute $e_{i_k,0}, e_{i_k,1}, ..., e_{i_k,l-1} \in \mathbb{F}_q$ s.t. $E_{i_k} := \sum_{j=0}^{l-1} e_{i_k j} v_j, \ \forall i_k \in \mathcal{E}$
  Compute $e_j(X) = \sum_{i \in \mathcal{E}_j} e_{i,j} X^i, \ \forall j \in [l]$
  Compute $c_j(X) = r_j(X) - e_j(X), \ \forall j \in [l]$
  $c(X) = (c_0(X), c_1(X), ..., c_{l-1}(X)) \longleftarrow$ **Output**

Table 5.1: Decoding Algorithm

Complexity Analysis:

The first line of the algorithm involves computing the syndrome polynomials, which includes computing l dot products and evaluating for $\delta - 1$ coefficients. That means this step has a complexity that is linear in l. The second line, where the generalised Berlekamp-Massey algorithm is employed, has a complexity which is quadratic in l. The next step involves finding the roots of the error locator polynomial, which can be done by using Chien search and thus has a complexity which is linear in the number of errors, ε.

Therefore, we can clearly see that the highest complexity term of the overall complexity will be from the second line of the algorithm, where the generalised Berlekamp-Massey algorithm is used. Thus, the overall complexity can be approximated as $\mathcal{O}(l^2)$.

Chapter 6

Niederreiter-like cryptosystem based on quasi-twisted codes

6.1 Introduction

Let l, m, λ be positive integers such that m is a prime power and l is bounded above by a polynomial in m. Consider an $[n = ml, k = (l-1)m]_q$ $(\lambda, l)-$quasi-twisted code $\mathcal{C}$ with rate $\frac{l-1}{l}$. (Recall that the rate of a code is defined as $\frac{k}{n}$).

Suppose $c = \{a_0, a_1, a_2, ..., a_{n-1}\} \in \mathcal{C}$ is a codeword. Then by definition 2.4.9, we know that $\{\lambda a_{n-l}, \lambda a_{n-l+1}, ..., \lambda a_{n-1}, a_0, ..., a_{n-l-1}\} \in \mathcal{C}$ is a codeword.

Definition 6.1.1. ***Twistulant matrix:*** *Consider $c = (c_0, c_1, \ldots, c_{m-1}) \in \mathbb{F}_q^m$. An $m \times m$ matrix is called a (right) twistulant matrix if its rows are composed of the (right) $\lambda-$constashifts of c. Precisely, G is a twistulant matrix if c forms the first row of G and every other row of G can be obtained by a $\lambda-$constashift of the row above it.*

$$G = \begin{bmatrix} c_0 & c_1 & c_2 & \ldots & c_{m-1} \\ \lambda c_{m-1} & c_0 & c_1 & \ldots & c_{m-2} \\ \lambda c_{m-2} & \lambda c_{m-1} & c_0 & \ldots & c_{m-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda c_1 & \lambda c_2 & \lambda c_3 & \ldots & c_0 \end{bmatrix}$$

A twistulant matrix can be represented by its first row. Notice that this row can be mapped to the polynomial $c(X) = c_0 + c_1X + c_2X^2 + \ldots + c_{m-1}X^{m-1} \in \mathbb{F}_q[X]/\langle X^m - \lambda\rangle$. This polynomial $c(X)$ is called the *defining polynomial* of the twistulant matrix G. When $\lambda = 1$, a twistulant matrix is indeed a circulant matrix.

It is proved in [27] that the generator matrix of a $(\lambda, l)-$QT code can be expressed in the form of blocks of twistulant matrices. Each row of such a matrix is conventionally termed a *generator*. Therefore, the generator matrix of a 1-generator QT-code can be expressed as,

$$G = \Big[G_0|G_1|\ldots|G_{l-1}\Big], \tag{6.1}$$

where each G_i is an $m \times m$ twistulant matrix.

Note that the dual code $\mathcal{C}^\perp$ of the $[ml, (l-1)m]$ $(\lambda, l)-$QT code, which is an $[ml, m]$ $(\lambda^{-1}, l)-$QT code [17], is a 1-generator code. Thus, the generator matrix of $\mathcal{C}^\perp$ will be of the form given in the above eqn. 6.1 with each G_i as:

$$G_i = \begin{bmatrix} c_0 & c_1 & c_2 & \ldots & c_{m-1} \\ \lambda^{-1}c_{m-1} & c_0 & c_1 & \ldots & c_{m-2} \\ \lambda^{-1}c_{m-2} & \lambda^{-1}c_{m-1} & c_0 & \ldots & c_{m-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda^{-1}c_1 & \lambda^{-1}c_2 & \lambda^{-1}c_3 & \ldots & c_0 \end{bmatrix}$$

V. Bhargava et. al. in [28] describes how a generator matrix of a rate $1/m$ code can be expressed in the standard form. Since $\mathcal{C}^\perp$ is a rate $1/m$ code, if one of the $G_i's$ is invertible (say, G_0), we can rewrite G in the standard form as,

$$G^* = \Big[\mathcal{I}|G_1^*|G_2^*|\ldots|G_{l-1}^*\Big], \tag{6.2}$$

where $\mathcal{I}$ is the $m \times m$ identity matrix and $G_i^* = G_i/G_0$, $\forall\ 1 \leq i \leq l-1$. It is proved in [29] that the set of all right twistulant matrices is closed under multiplication. Moreover, in [30], it is proved that for an invertible $\lambda-$twistulant matrix, its inverse is also a $\lambda-$twistulant matrix. Using these two facts, we can clearly see that G_i^* $(\forall\ 1 \leq i \leq l-1)$ is also a $\lambda-$twistulant matrix.

Since G^* forms a generator matrix of $\mathcal{C}^\perp$, we can consider $H = G^*$ as a parity-check matrix of $\mathcal{C}$, which is the quasi-twisted code of our interest.

6.2 Classical Security

Here, we discuss the classical security of our cryptosystem. The equivalence of McEliece and Niederreiter cryptosystems from a security perspective is proved in [6]. Two important classical attacks against McEliece or Niederreiter cryptosystems are considered in this section.

1. Information Set Decoding (ISD) attacks: These can be induced by two strategies, namely, Lee and Brickell's method and Stern's algorithm. From [31], we know that these attacks have the minimum work factor for the classical cryptanalysis purpose, and the so attained minimum work factor can, in turn, be considered as the security level of these cryptosystems.

   Now we analyse the minimum work factor for our proposed variant using $[ml, (l-1)m]$ $(\lambda, l)-$QT code. Let the error-correcting capacity of this code be ε.

   Then (using [32]) the minimum work factor is given by,

   $$W_{min} = T_2(\alpha k^3 + N_2 \beta k) = T_2(k^3 + N_2 k)$$

   setting $\alpha = \beta = 1$ and plugging in $n = ml$ and $k = (l-1)m$, it simplifies to:

   $$W_{min} = T_2\big((l-1)^3 m^3 + (l-1)mN_2\big)$$

   $$\text{where, } T_2 = \frac{1}{\sum_{i=0}^{2} Q_i}, Q_i = \frac{\binom{\varepsilon}{i}\binom{n-2}{k-i}}{\binom{n}{k}} \text{ and } N_2 = \sum_{i=0}^{2} \binom{k}{i}$$

2. Attack on dual code: This attack is prone to happen when the parity-check matrix is sparse, resulting in a dual code with low-weight codewords. This can be resisted by avoiding sparse parity-check matrices.

6.3 Quantum Security

In this section, we show that our cryptosystem, using the quasi-twisted codes, can withstand Quantum Fourier Sampling (QFS), depicting its resistance to all those quantum at-

tacks based on QFS. Quantum Fourier Sampling (QFS) plays a central role in many quantum algorithms, including Shor's algorithm. A detailed background can be found in [33].

6.3.1 Hidden Subgroup Problem

One way of attacking the Niederreiter cryptosystem is the Scrambler-Pemutation attack, which further can be reduced into an instance of the Hidden Subgroup problem. QFS can be used to solve the Hidden Subgroup Problem. We define the following, keeping in mind the following setting. The public key H' of a Niederreiter cryptosystem is generated as $H' = SHP$, where H is the $(n-k) \times n$ parity-check matrix of the underlying code, S is a non-singular matrix of size $(n-k)$ and P is an $n \times n$ permutation matrix.

Definition 6.3.1. Scrambler Permutation Attack: *This attack involves finding the scrambler-permutation pair i.e., the matrices S and P, assuming that H and H' are known. (Any S', P' satisfying $H' = S'HP'$ is enough to make the attack successful.)*

Definition 6.3.2. Hidden Shift Problem: *Consider a finite group G and a finite set Σ. Define two functions $f_0, f_1 : G \longrightarrow \Sigma$. The problem is concerned with finding a constant $g \in G$ such that $f_1(x) = f_0(gx), \forall x \in G$, provided that such a constant (termed as 'shift' from f_0 to f_1) exists. (There might exist more than one such shift; finding any one from those is enough.)*

We can see that the Scrambler-Permutation problem reduces to the Hidden Shift problem when the group is taken to be $G = GL_{n-k}(\mathbb{F}_q) \times S_n$ and the functions are defined on this group such that $\forall (S, P) \in GL_{n-k}(\mathbb{F}_q) \times S_n$,

$$f_0(S, P) = S^{-1}HP, \quad f_1(S, P) = S^{-1}H'P$$

(here, the permutation matrix P is mapped to the corresponding permutation in S_n.) So, we see that $H' = SHP \iff (S^{-1}, P)$ is a shift from f_0 to f_1. That is,

$$f_0((S^{-1}, P)(S, P)) = S^{-1}(SHP)P = S^{-1}H'P = f_1(S, P)$$

$\forall (S, P) \in GL_{n-k}(\mathbb{F}_q) \times S_n$.

Definition 6.3.3. Hidden Subgroup Problem: *Given a function f on a group G such*

that $f(x_1) = f(x_2)$ if and only if $x_1H = x_2H$ for some unknown subgroup $H < G$, the problem is to find a set of generators for the subgroup H.

The Hidden Shift problem on a group G reduces to the Hidden Subgroup problem on $G \wr \mathbb{Z}_2 = G^2 \rtimes \mathbb{Z}_2$. Consider two functions f_0 and f_1 defined on G and define a function $f : G \wr \mathbb{Z}_2 \longrightarrow \Sigma \times \Sigma$. For $(x_1, x_2) \in G^2$ and $a \in \mathbb{Z}_2$,

$$f\big((x_1, x_2), a\big) := \begin{cases} \big(f_0(x_1), f_1(x_2)\big) & \text{if } a = 0 \\ \big(f_1(x_2), f_0(x_1)\big) & \text{if } a = 1 \end{cases}$$

When the group in Hidden Shift problem is taken to be $G = GL_{n-k}(\mathbb{F}_q) \times S_n$ and the shift from f_0 to f_1 to be s, it reduces to the Hidden Subgroup problem on $G^2 \rtimes \mathbb{Z}_2 = (GL_{n-k}(\mathbb{F}_q) \times S_n)^2 \rtimes \mathbb{Z}_2$. If $H_0 := G|_{f_0}$, then the hidden subgroup is

$$K := G \wr \mathbb{Z}_2|_f = \left(\left(\Big(H_0, s^{-1}H_0 s\Big), 0\right) \bigcup \left(\Big(H_0 s, s^{-1}H_0\Big), 1\right)\right)$$

Finding this hidden subgroup $K = G \wr \mathbb{Z}_2|_f$ enables us to find a shift from f_0 to f_1. That is, if we have $\big((g_1, g_2), 1\big) \in K$, then it means that $g_1 \in H_0 s$. Thus, g_1 is a shift from f_0 to f_1. We can verify this as below, by taking $s = (S^{-1}, P)$. We have,

$$H_0 := G|_{f_0} \implies H_0 = \{(S, P) \in GL_{n-k}(\mathbb{F}_q) \times S_n : S^{-1}HP = H\}$$

Let $(\mathcal{S}, \mathcal{P}) \in H_0$. Then, by definition, we have $\mathcal{S}^{-1}H\mathcal{P} = H$.
So, $(\mathcal{S}, \mathcal{P})(S^{-1}P) = (\mathcal{S}S^{-1}, \mathcal{P}P) \in H_0 s$.

$$\implies f_0\Big(\big(\mathcal{S}S^{-1}, \mathcal{P}P\big)\big(S, P\big)\Big) = S^{-1}S\mathcal{S}^{-1}H\mathcal{P}PP = S^{-1}SHPP = S^{-1}H'P = f_1(S, P)$$

Therefore, $(\mathcal{S}S^{-1}, \mathcal{P}P) \in H_0 S$ is indeed a shift. ∎

Summing up: In short, we see that solving the hidden subgroup problem on $(GL_{n-k}(\mathbb{F}_q) \times S_n)^2 \rtimes \mathbb{Z}_2$ for the hidden subgroup taken as K leads to a solution to the hidden shift problem over $GL_{n-k}(\mathbb{F}_q) \times S_n$. This, in turn corresponds to the scrambler-permutation problem, which as stated before is an attack on the Niederreiter cryptosystem. So, we now focus on the Hidden Subgroup problem on $(GL_{n-k}(\mathbb{F}_q) \times S_n)^2 \rtimes \mathbb{Z}_2$ to prove the resistance of our cryptosystem to the Scrambler-Permutation attack.

6.3.2 Indistinguishability by QFS

In this section, we describe a sufficient condition for the indistinguishability of the subgroup K using some of the results in [33]. Let G be a finite group. Consider QFS over G in the given basis, $\{B_\rho\}$. Two subgroups, H_1 and H_2, are said to be indistinguishable if their probability distributions, say P_{H_1} and P_{H_2}, have their total variations really close to one another. We check the indistinguishability of the hidden subgroup $H < G$ from its conjugate subgroups gHg^{-1} or the trivial subgroup $\langle e \rangle$. Weak Fourier sampling cannot distinguish the conjugate subgroups since $P_{gHg^{-1}}$ does not depend on g, giving the same distribution for all the conjugate subgroups.

For some non-trivial subgroup H, we aim to show that strong Fourier sampling cannot distinguish its conjugates from one another or from the trivial subgroup, efficiently. It is to be noted that the probability distribution of the trivial hidden subgroup using strong Fourier sampling, $P_{\langle e \rangle}(\cdot|\rho)$, is the same as the uniform distribution U_{B_ρ} on the basis B_ρ, where ρ is an irreducible representation $\rho \in \hat{G}$ given by weak Fourier sampling. (where $\hat{G}$ is the set of irreducible unitary representations of G). Thus, it is enough to show that, for a random $g \in G$, $P_{gHg^{-1}}(\cdot|\rho)$ is close to U_{B_ρ} in total variation.

We begin by restating the following definition from [33, Defn. 5].

Definition 6.3.4. Distinguishability of a subgroup using strong QFS: *Let $\mathcal{D}_H$ denote the distinguishability of a subgroup $H < G$ using strong Fourier sampling over G. Then, $\mathcal{D}_H$ is defined to be the expectation of the squared L_1-distance between $P_{gHg^{-1}}(\cdot|\rho)$ and U_{B_ρ}. That is,*

$$\mathcal{D}_H := \mathbb{E}_{\rho,g}\Big[\left\| P_{gHg^{-1}}(\cdot|\rho) - U_{B_\rho} \right\|_1^2 \Big]$$

for $\rho \in \hat{G}$ and a random $g \in G$. The subgroup H is said to be indistinguishable if,

$$\mathcal{D}_H \leq \log^{-\omega(1)} |G| \tag{6.3}$$

Using Markov's inequality, if strong Fourier sampling cannot distinguish the subgroup H, then for all constant $c > 0$,

$$\left\| P_{gHg^{-1}}(\cdot|\rho) - U_{B_\rho} \right\|_{t.v.} < \log^{-c} |G|$$

with a minimum probability of $1 - \log^{-c} |G|$ in both g and ρ.

6.3.3 Application to the Niederreiter-like cryptosystem

Recollect that we are considering the Niederreiter-like cryptosystem using $[n = ml, k = (l-1)m]$ $(\lambda, l)-$quasi-twisted code $\mathcal{C}$ with parity-check matrix of the form 6.2.

$$i.e., H = \Big[\mathcal{I}|G_1^*|G_2^*|\ldots|G_{l-1}^*\Big]_{m\times ml} \tag{6.4}$$

where, $\mathcal{I}$ is the $m\times m$ identity matrix and G_i^* is a $\lambda^{-1}-$twistulant matrix with entries from $\mathbb{F}_q$. That means, for all $1 \leq i \leq l-1$,

$$G_i^* = \begin{bmatrix} c_0 & c_1 & c_2 & \ldots & c_{m-1} \\ \lambda^{-1}c_{m-1} & c_0 & c_1 & \ldots & c_{m-2} \\ \lambda^{-1}c_{m-2} & \lambda^{-1}c_{m-1} & c_0 & \ldots & c_{m-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda^{-1}c_1 & \lambda^{-1}c_2 & \lambda^{-1}c_3 & \ldots & c_0 \end{bmatrix}_{m\times m}$$

Note that $n-k=m$ for the $[ml,(l-1)m]_q$ code under consideration.

Remark 6.3.1. *We impose certain conditions on the parity-check matrix* $H = \Big[\mathcal{I}|G_1^*|G_2^*|\ldots|G_{l-1}^*\Big]$ *as follows:*

1. *No two twistulant matrices* G_i^* *and* G_j^**, for* $i \neq j, 1 \leq i,j \leq l-1$*, can have the same defining polynomial,* $c(X) = c_0 + c_1X + c_2X^2 + \ldots + c_{m-1}X^{m-1} \in \mathbb{F}_q[X]/\langle X^m - \lambda\rangle$.

2. *If* $\lambda = 1$ *(or equivalently, if* $q = 2$*), at least one of the coefficients of the defining polynomial,* $c(X)$*, should be distinct from the rest of the coefficients of* $c(X)$*. In other words,* $\exists\ i,\ 1 \leq i \leq m-1$ *such that* $c_i \neq c_j$ *for all* $j \neq i,\ 1 \leq j \leq m-1$.

Recall that, the scrambler-permutation attack on the Niederreiter cryptosystem boils down to the Hidden Subgroup problem on $(GL_m(\mathbb{F}_q) \times S_n)^2 \rtimes \mathbb{Z}_2$ with the hidden subgroup as

$$K = G \wr \mathbb{Z}_2|_f = \Bigg(\Big(\Big(H_0, s^{-1}H_0s\Big), 0\Big) \bigcup \Big(\Big(H_0s, s^{-1}H_0\Big), 1\Big)\Bigg)$$

for a hidden element $s \in GL_m(\mathbb{F}_q) \times S_n$ and here,

$$H_0 := G|_{f_0} \implies H_0 = \{(S,P) \in GL_m(\mathbb{F}_q) \times S_n : S^{-1}HP = H\}.$$

We can define the **automorphism group** of the linear code generated by H ($Aut(H)$) as the projection of H_0 onto S_n. i.e.,

$$Aut(H) = \{P \in S_n : S^{-1}HP = H \text{ for some } S \in GL_m(\mathbb{F}_p)\}$$

$\implies \forall (S, P) \in H_0, \exists P \in Aut(H)$

Definition 6.3.5. Minimal degree: *The minimal degree of a permutation group is defined to be the number of points that are not fixed by a non-trivial element (i.e., not the identity element) of the group.*

Recall the definition of $\mathcal{D}_K$ from definition 6.3.4. We can rewrite the [33, Thm. 4] for Niederreiter cryptosystems as follows:

Theorem 6.3.1. *Assume $q^{(n-k)^2} \leq n^{an}$ for some constant $0 < a < 1/4$. Let d be the minimal degree of the automorphism group $Aut(H)$. Then, for sufficiently large n, the subgroup K has $\mathcal{D}_K \leq O(|K|^2 e^{-\delta d})$, where $\delta > 0$ is a constant.*

Remark 6.3.2. *Recall from eqn. 6.3 that the subgroup K is indistinguishable if $\mathcal{D}_K \leq \log^{-\omega(1)}|G|$. Using the given assumption $q^{(n-k)^2} \leq n^{an}$, we can simplify $\log|G|$ as $\log|(GL_{n-k}(\mathbb{F}_q) \times S_n)^2 \rtimes \mathbb{Z}_2| = O(\log n! + \log q^{(n-k)^2}) = O(n \log n)$. This implies that the subgroup K is indistinguishable if $|K|^2 e^{-\delta d} \leq (n \log n)^{-\omega(1)}$.*

Note that the size of the subgroup, K, is $|K| = 2|H_0|^2$ and $|H_0| = |Aut(H)| \times |Fix(H)|$, where $Fix(H) := \{S \in GL_{n-k}(\mathbb{F}_q) : SH = H\}$. Clearly, $|Fix(H)| = 1$ or $|H_0| = |Aut(H)|$.

In short, in order to check for indistinguishability of the hidden subgroup K, we need to find the size and the minimal degree of the Automorphism group, $Aut(H)$.

Recall that our parity-check matrix H is an $m \times ml$ matrix, where m is a prime power. Using eqn. 6.4, we can rewrite H as $\Big[\mathcal{I}|G_1^*|G_2^*|\dots|G_{l-1}^*\Big]$, where, $\mathcal{I}$ is the $m \times m$ identity matrix and G_i^* is a $\lambda^{-1}-$twistulant matrix.

Consider $P \in Aut(H)$. By definition, there exists an $S \in GL_m(\mathbb{F}_q)$ such that,

$$S^{-1}HP = H$$

Let $H = \Big[\mathcal{I}|C\Big]$ such that C is a concatenation of G_i^*, which means $C := \Big[G_1^*|G_2^*|\dots|G_{l-1}^*\Big]$. Therefore,

$$S^{-1}\Big[\mathcal{I}|C\Big]P = \Big[S^{-1}|S^{-1}C\Big]P = \Big[\mathcal{I}|C\Big].$$

Since right multiplication by a permutation matrix essentially permutes the columns, the P matrix will be permuting the columns of $S^{-1}H$ to get the H matrix. Notice that, the entries in S are from $\mathbb{F}_p$ whereas C has its entries from $\mathbb{F}_q$, where $q = p^n$ for some $n > 0$. Therefore, $S^{-1}C$ cannot have any columns of C. In other words, P permutes the columns of S^{-1} to get I and that of $S^{-1}C$ to get C. This implies that every permutation matrix $P \in Aut(H)$ is a block diagonal matrix, whose top block, P_0, is of size m. Moreover, the twistulant structure of C forces the block diagonal matrix P to have $l-1$ more blocks of size m each and P can be represented as,

$$P = \begin{pmatrix} P_0 & & & \\ & P_1 & 0 & \\ & & \ddots & \\ & & & P_{l-2} \\ & 0 & & & P_{l-1} \end{pmatrix}_{ml \times ml}$$

where each block P_i acts on $S^{-1}G_i^*$, for all $1 \leq i \leq l-1$. That means,

$$\Big[S^{-1}|S^{-1}C\Big]P = \Big[S^{-1}P_0|S^{-1}G_1^*P_1|S^{-1}G_2^*P_2|\ldots|S^{-1}G_{l-1}^*P_{l-1}\Big] = \Big[I|G_1^*|G_2^*|\ldots|G_{l-1}^*\Big]$$

Observe that $S^{-1}P_0 = I$. Therefore, $S^{-1} = P_0^{-1}$ and thus, $P_0^{-1}G_i^*P_i = G_i^*$ for all $1 \leq i \leq l-1$. Recall that, $Aut(H) = \{P \in S_n : S^{-1}HP = H \text{ for some } S \in GL_m(\mathbb{F}_p)\}$. Note that the size of $Aut(H)$ can now be rewritten as the cardinality of the following set:

$$A_1 := \Big\{(P_0, P_i)|P_0^{-1}G_i^*P_i = G_i^* \text{ for some } 1 \leq i \leq l-1\Big\}$$

Recall the conditions specified in Remark 6.3.1. The second condition implies that no two columns within any of the G_i^* can be identical, along with the first condition, we can infer that no two columns of H are identical. Moreover, $S^{-1}H = P_0^{-1}H$ doesn't have any identical columns. In the premise of the set A_1, we can see that, for all $1 \leq i \leq l-1$, $P_0^{-1}G_i^*$ has distinct columns, and thereby there exists at most one such P_i that can permute the columns of $P_0^{-1}G_i^*$ to get G_i^*. Therefore, the cardinality of the set A_1 is the number of such P_0's. That is, if

$$A_2 := \Big\{P_0 \in S_m|P_0^{-1}G_i^*P_i = G_i^* \text{ for all } 1 \leq i \leq l-1\Big\}$$

then, $|Aut(H)| = |A_1| = |A_2|$.

Observe that the group $Aut(H)$ and thereby, A_2 are subgroups of the symmetric group S_n (by definition), that is $Aut(H) < S_n$ and $A_2 < S_m$.

Now we state the Burnside-Schur theorem, which will prove useful in finding the size and minimal degree of $Aut(H)$.

Theorem 6.3.2. (Burnside-Schur) *Every primitive finite permutation group containing a regular cyclic subgroup is either 2-transitive or permutationally isomorphic to a subgroup of the affine group $AGL_1(p)$ where p is a prime.*

In the theorem statement, $AFL_1(p)$ denotes the affine group of degree one over the field of p elements ($\mathbb{F}_p$), where p is a prime number. It is proved in [34, pp. 339-343] that the theorem holds true for the permutation groups of prime power degree as well.

Recall that A_2 is a subgroup of S_m, where m is a prime power. Now if it contains an $m-$cycle, then we can apply the Burnside-Schur theorem on A_2. Consider the following matrix:

$$B = \begin{bmatrix} 0 & 0 & \dots & 0 & \lambda \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}_{m \times m}$$

Notice that B corresponds to an $m-$cycle. Now we check if B is an element of A_2. Recall that $A_2 = \left\{ P_0 \in S_m | P_0^{-1} G_i^* P_i = G_i^* \text{ for all } 1 \leq i \leq l-1 \right\}$. Therefore, if $B \in A_2$, then $B^{-1} G_i^* P_i$ should give us G_i^*. We evaluate it as follows:

$$B^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ \lambda^{-1} & 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{m \times m}$$

$$B^{-1}G_i^* = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ \lambda^{-1} & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} c_0 & c_1 & c_2 & \dots & c_{m-1} \\ \lambda^{-1}c_{m-1} & c_0 & c_1 & \dots & c_{m-2} \\ \lambda^{-1}c_{m-2} & \lambda^{-1}c_{m-1} & c_0 & \dots & c_{m-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda^{-1}c_1 & \lambda^{-1}c_2 & \lambda^{-1}c_3 & \dots & c_0 \end{bmatrix}$$

$$= \begin{bmatrix} \lambda^{-1}c_{m-1} & c_0 & c_1 & \dots & c_{m-2} \\ \lambda^{-1}c_{m-2} & \lambda^{-1}c_{m-1} & c_0 & \dots & c_{m-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda^{-1}c_1 & \lambda^{-1}c_2 & \lambda^{-1}c_3 & \dots & c_0 \\ \lambda^{-1}c_0 & \lambda^{-1}c_1 & \lambda^{-1}c_2 & \dots & \lambda^{-1}c_{m-1} \end{bmatrix}$$

We see that for $P_i = B$, $B^{-1}G_i^*P_i = G_i^*$, which is demonstrated below:

$$B^{-1}G_i^*B = \begin{bmatrix} \lambda^{-1}c_{m-1} & c_0 & c_1 & \dots & c_{m-2} \\ \lambda^{-1}c_{m-2} & \lambda^{-1}c_{m-1} & c_0 & \dots & c_{m-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda^{-1}c_1 & \lambda^{-1}c_2 & \lambda^{-1}c_3 & \dots & c_0 \\ \lambda^{-1}c_0 & \lambda^{-1}c_1 & \lambda^{-1}c_2 & \dots & \lambda^{-1}c_{m-1} \end{bmatrix} \begin{bmatrix} 0 & 0 & \dots & 0 & \lambda \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} c_0 & c_1 & c_2 & \dots & c_{m-1} \\ \lambda^{-1}c_{m-1} & c_0 & c_1 & \dots & c_{m-2} \\ \lambda^{-1}c_{m-2} & \lambda^{-1}c_{m-1} & c_0 & \dots & c_{m-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda^{-1}c_1 & \lambda^{-1}c_2 & \lambda^{-1}c_3 & \dots & c_0 \end{bmatrix} = G_i^*$$

Note that for all $1 \leq i \leq l-1$, $B^{-1}G_i^*B = G_i^*$. Therefore, we can say that B belongs to the set A_2. Now, on applying the Burnside-Schur theorem on A_2, we can conclude that A_2 is either 2-transitive or is permutationally isomorphic to $AGL_1(m)$. We restrict ourselves to the latter case of $A_2 \leq AGL_1(m)$. Thereby, we get the size of A_2 to be bounded above by the size of $AGL_1(m)$, which means $|A_2| \leq m(m-1)$. Using our previous observations and arguments, we can conclude that the size of the automorphism group, $Aut(H)$, is given by, $|Aut(H)| \leq m(m-1)$.

Observe that $B \in A_2$ and $A_2 \leq AGL_1(m)$. Therefore, we see that $B \in AGL_1(m)$. Then, for some $A \in M_{m\times m}(\mathbb{F})$ and $b \in \mathbb{F}_m$, $B(x) = Ax + b \mod m$, $\forall x \in \mathbb{F}_m$. Recall from the definition (definition 6.3.5) that the minimal degree is defined as the number of points

that are not fixed by a non-identity element of the group. If B fixes more than one point, then A must be the identity matrix $\mathcal{I}_m$ and b must be the zero vector, $b = 0$. However, that means B is an identity element. Therefore, any non-trivial element fixes at most one element. Hence, the minimal degree of $Aut(H)$ is at least $m - 1$.

Now, the following theorem provides the necessary conditions under which the hidden subgroup K, corresponding to the proposed cryptosystem, is indistinguishable.

Theorem 6.3.3. *Consider an $[n = ml, k = (l-1)m]_q$ $(\lambda, l)-$quasi-twisted code $\mathcal{C}$, where m is a prime power. Suppose $m < 1/4l(\log_q m + \log_q l)$. Then the subgroup K defined as follows:*

$$K = G \wr \mathbb{Z}_2|_f = \left(\left(\left(H_0, s^{-1} H_0 s \right), 0 \right) \bigcup \left(\left(H_0 s, s^{-1} H_0 \right), 1 \right) \right),$$

is indistinguishable.

Proof. Recall from Theorem 6.3.1 and Remark 6.3.2 that under the assumption of $q^{n-k^2} \leq n^{an}$, the subgroup K is said to be indistinguishable if $|K|^2 e^{-\delta d} \leq (n \log n)^{-\omega(1)}$. Here, $\delta > 0$ is a constant and d is the minimal degree of the automorphism group $Aut(H)$. We can see that the assumption of Theorem 6.3.1, $q^{n-k^2} \leq n^{an}$, for some $0 < a < 1/4$, holds for our case from the following: (From here on, log refers to log to the base q.)

$$\begin{aligned} q^{m^2} &\leq (ml)^{aml} \\ m^2 &\leq aml \log(ml) \\ m^2 &\leq aml(\log m + \log l) \\ m &\leq al(\log m + \log l) \end{aligned}$$

Since $0 < a < 1/4$, this is indeed the assumption given in this theorem, $m < 1/4l(\log_q m + \log_q l)$. Now, we can use the theorem 6.3.1 to check for the indistinguishability of K. Recall that $|K| = 2|H_0|^2 = 2|Aut(H)|^2$, where $|Aut(H)| \leq m(m-1)$ and $d \geq m - 1$ On substituting the values for $|K|$ and d, we get

$$|K|^2 e^{-\delta d} \leq (2m^4)^2 e^{-\delta m}$$

Therefore, $|K|^2 e^{-\delta d} \leq 4m^8 e^{-\delta m}$, where $\delta > 0$ is a constant. Using the bound on l, we can see that $4m^8 e^{-\delta m} \leq (ml \log(ml))^{-\omega(1)}$. Hence, we get $|K|^2 e^{-\delta d} \leq (ml \log(ml))^{-\omega(1)}$ and therefore, the hidden subgroup K is indistinguishable. Thereby, we can conclude that the cryptosystem can withstand QFS attacks. ∎

Chapter 7

Conclusion

This thesis addresses the rising need for a quantum-secure cryptosystem in response to the expanding capabilities of quantum computing. The McEliece cryptosystem (equivalently, the Niederreiter cryptosystem) is a potential solution, given that it is quantum-secure and facilitates faster encryption and decryption. However, the large key sizes make it not feasible for implementation. We analyzed the different modifications of the McEliece/Niederreiter cryptosystems and found a variant of the Niederreiter cryptosystem based on quasi-cyclic codes (presented in [16]). This work inspired us to further study the quasi-cyclic codes and a more generalized code family - quasi-twisted codes. We explored the possibility of replacing Goppa codes in the Niederreiter cryptosystem with quasi-twisted codes to develop a more secure code-based cryptosystem.

It is necessary for a code to have an efficient decoding algorithm in order to adopt it in a code-based cryptosystem. One main challenge in developing a Niederreiter-like cryptosystem based on quasi-twisted codes was to establish a decoding algorithm. In order to provide a decoding procedure, we show a new lower bound on the minimum distance of quasi-twisted codes - the Hartmann-Tzeng(HT)-like bound (discussed in theorem 5.2.2). We then describe a syndrome-based decoding algorithm in Section 5.3 that can correct up to $\varepsilon = \frac{d^*-1}{2}$ errors, where d^* is the minimum distance determined by the HT-like bound for quasi-twisted codes. Our algorithm operates with a time complexity that is quadratic in relation to the length of the code.

Having described a decoding procedure for quasi-twisted codes, we present a Niederreiter-

like cryptosystem based on quasi-twisted codes in Chapter 6 of this thesis. We draw motivation from the quasi-cyclic variant ([16]) and develop a more secure cryptosystem that uses $[ml, (l-1)m]_q$ $(\lambda, l)-$quasi-twisted codes, where m is a prime power and l is bounded above by a polynomial in m. We show that our cryptosystem is both classical as well as quantum secure. To prove the quantum security, we consider the Scrambler-Permutation attack and prove that our cryptosystem can resist such an attack. This is done by proving the indistinguishability of the hidden subgroup in the Hidden Subgroup Problem, by quantum Fourier sampling (QFS). Resistance to QFS attacks is considered as a measure of quantum-security and we show that our cryptosystem can withstand QFS attacks. As we currently understand, the quasi-cyclic code-based cryptosystem withstands any known attacks. Therefore, we expect that this would be the case for our cryptosystem as well.

We leave the analysis of the security of our cryptosystem against attacks that are not based on QFS, such as variants of the Sidelnikov-Shestakov attack [7] that uses the dimension of the Schur square of the underlying code for future work. Furthermore, the parameter bounds presented in this work may not represent the most efficient or optimal values. There is potential for these bounds to be further refined and improved through more in-depth analysis in future research, which could lead to enhanced performance and more accurate results.

Bibliography

[1] A.L. Horlemann. "An Introduction to Code-Based Cryptography." *In Summer School Finite Geometry and Friends,* 2023.

[2] V. Weger, N. Gassner, and J. Rosenthal. "A survey on code-based cryptography," *arXiv.org,* 2022. https://arxiv.org/abs/2201.07119

[3] R. J. McEliece. "A public-key cryptosystem based on algebraic coding theory." *Technical report, DSN Progress report,* pages 42-44, Jet Propulsion Laboratory, Pasadena, 1978.

[4] H. Niederreiter. "Knapsack-type cryptosystems and algebraic coding theory." *Problems of Control and Information Theory* 15, 1(6):159–166, 1986.

[5] S. V. Bezzateev and I. K. Noskov, "Patterson Algorithm for Decoding Separable Binary Goppa Codes," *2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF),* St. Petersburg, Russia, pp. 1-5, 2019. https://doi.org/10.1109/WECONF.2019.8840650

[6] Yuan Xing Li, R. H. Deng and Xin Mei Wang, "On the equivalence of McEliece's and Niederreiter's public-key cryptosystems," *in IEEE Transactions on Information Theory,* vol. 40, no. 1, pp. 271-273, Jan. 1994. https://doi.org/10.1109/18.272496

[7] V. M. Sidelnikov and S. O. Shestakov, "On insecurity of cryptosystems based on generalized Reed-Solomon codes," *Discrete Mathematics and Applications,* vol. 2, no. 4, 1992. https://doi.org/10.1515/dma.1992.2.4.439

[8] C. Wieschebrink. "Two NP-complete problems in coding theory with an application in code based cryptography." *In 2006 IEEE International Symposium on Information Theory,* pages 1733–1737, 2006.

[9] E. M. Gabidulin and O. Kjelsen, "How to avoid the Sidel'nikov-Shestakov attack," *Lecture Notes in Computer Science,* pp. 25–32, 1994. https://doi.org/10.1007/3-540-58265-7_4

[10] K. Khathuria, J. Rosenthal and V. Weger, "Encryption scheme based on expanded Reed-Solomon codes," *Advances in Mathematics of Communications,* 15(2), 207–218, 2021. https://doi.org/10.3934/amc.2020053

[11] A. Couvreur and M. Lequesne, "On the security of subspace subcodes of Reed-Solomon codes for public key encryption," *arXiv.org,* 2020. http://arxiv.org/abs/2009.05826

[12] P. Beelen, S. Puchinger, and J. Rosenkilde, "Twisted Reed-Solomon Codes," *arXiv.org,* 2021. http://arxiv.org/abs/2107.06945

[13] A. Allen, K. Blackwell, O. Fiol, R. Kshirsagar, B. Matsick, G. L. Matthews, and Z. Nelson, "Twisted Hermitian Codes," *Mathematics*, 9(1), 40, 2020. https://doi.org/10.3390/math9010040

[14] J. Lavauzelle and J. Renner, "Cryptanalysis of a system based on twisted Reed–Solomon codes," *Designs, Codes and Cryptography*, vol. 88, no. 7, pp. 1285–1300, Mar. 2020. https://doi.org/10.1007/s10623-020-00747-6.

[15] U. Kapshikar, "McEliece-type Cryptosystems over Quasi-cyclic Codes," *arXiv.org*, May 2018. http://arxiv.org/abs/1805.09972

[16] U. Kapshikar and A. Mahalanobis, "A Quantum-Secure Niederreiter Cryptosystem using Quasi-Cyclic Codes," *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications*, pp. 506–513, Jan. 2018. https://doi.org/10.5220/0006843005060513

[17] Y. Jia, "On quasi-twisted codes over finite fields," *Finite Fields and Their Applications*, vol. 18, no. 2, pp. 237–257, Sep. 2011. https://doi.org/10.1016/j.ffa.2011.08.001

[18] P. Semenov and P. Trifonov, "Spectral Method for Quasi-Cyclic Code Analysis," *in IEEE Communications Letters*, vol. 16, no. 11, pp. 1840-1843, November 2012. https://doi.org/10.1109/lcomm.2012.091712.120834

[19] A. Zeh and S. Ling, "Decoding of quasi-cyclic codes up to a new lower bound on the minimum distance," *2014 IEEE International Symposium on Information Theory,* Honolulu, HI, USA, 2014, pp. 2584-2588, https://doi.org/10.1109/ISIT.2014.6875301

[20] K. Lally and P. Fitzpatrick, "Algebraic structure of quasicyclic codes," *2000 IEEE International Symposium on Information Theory* (Cat. No.00CH37060), Sorrento, Italy, 2000, pp. 196-, https://doi.org/10.1109/ISIT.2000.866494

[21] M. F. Ezerman, S. Ling, B. Özkaya and J. Tharnnukhroh, "Spectral Bounds for Quasi-Twisted Codes," *2019 IEEE International Symposium on Information Theory*

(ISIT), Paris, France, 2019, pp. 1922-1926, https://doi.org/10.1109/ISIT.2019.8849734

[22] M. F. Ezerman, J. M. Lampos, S. Ling, B. Özkaya and J. Tharnnukhroh, "A Comparison of Distance Bounds for Quasi-Twisted Codes," *IEEE Transactions on Information Theory*, vol. 67, no. 10, pp. 6476-6490, Oct. 2021, https://doi.org/10.1109/TIT.2021.3084146

[23] C.R.P. Hartmann, K.K. Tzeng, "Generalizations of the BCH bound," *Information and Control*, Volume 20, Issue 5, 1972, Pages 489-498, ISSN 0019-9958, https://doi.org/10.1016/S0019-9958(72)90887-X

[24] G.-L. Feng and K.K. Tzeng, "A generalization of the Berlekamp-Massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes," *in IEEE Transactions on Information Theory*, vol. 37, no. 5, pp. 1274-1287, Sept. 1991, https://doi.org/10.1109/18.133246

[25] G.-L. Feng and K. K. Tzeng, "A generalised Euclidean algorithm for multisequence shift-register synthesis," *in IEEE Transactions on Information Theory*, vol. 35, no. 3, pp. 584-594, May 1989, https://doi.org/10.1109/18.30981

[26] J. van Lint and R. Wilson, "On the minimum distance of cyclic codes," *in IEEE Transactions on Information Theory*, vol. 32, no. 1, pp. 23-40, January 1986, https://doi.org/10.1109/TIT.1986.1057134

[27] N. Aydin, I. Siap, and D. K. Ray-Chaudhuri, "The structure of 1-generator quasi-twisted codes and new linear codes," *Designs Codes and Cryptography*, vol. 24, no. 3, pp. 313–326, Jan. 2001, https://link.springer.com/content/pdf/10.1023/A:1011283523000.pdf

[28] V. Bhargava, G. Seguin, and J. Stein, "Some (ink, k) cyclic codes in quasi-cyclic form (Corresp.)," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 630–632, Sep. 1978, https://doi.org/10.1109/tit.1978.1055930

[29] S. Jitman, S. Ruangpum, and T. Ruangtrakul, "Group structures of complex twistulant matrices," *AIP Conference Proceedings*, vol. 1775, p. 030016, Jan. 2016, https://doi.org/10.1063/1.4965136

[30] H. T. Recillas and J. A. Velazco-Velazco, "Group structures of twistulant matrices over rings," *International Electronic Journal of Algebra*, Sep. 2024, https://doi.org/10.24330/ieja.1596075

[31] M. Baldi, M. Bodrato, and Franco Chiaraluce, "A New Analysis of the McEliece Cryptosystem Based on QC-LDPC Codes," *Lecture notes in computer science,* pp. 246–262, Jan. 2008, https://doi.org/10.1007/978-3-540-85855-3_17

[32] P. J. Lee and E. F. Brickell, "An Observation on the Security of McEliece's Public-Key Cryptosystem," *Lecture notes in computer science,* pp. 275–280, Jan. 1988, https://doi.org/10.1007/3-540-45961-8_25

[33] H. Dinh, C. Moore, and A. Russell, "McEliece and Niederreiter cryptosystems that resist quantum Fourier sampling attacks," *Lecture Notes in Computer Science,* pp. 761–779, 2011. https://www.iacr.org/archive/crypto2011/68410758/68410758.pdf

[34] W. Burnside, "Theory of Groups of Finite Order," *Cambridge Univ. Press,* 2nd Edition, London, 1911.