McEliece-type Cryptosystems Over Quasi-Cyclic Codes

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

Upendra Kapshikar

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

May, 2018

Supervisor: Dr. Ayan Mahalanobis

Certificate

This is to certify that this dissertation entitled **McEliece-type Cryptosystems Over Quasi-Cyclic Codes** 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 **Upendra Kapshikar** at Indian Institute of Science Education and Research under the supervision of **Dr. Ayan Mahalanobis**, Assistant Professor, Department of Mathematics , during the academic year 2017-2018.

Ayan Mahalanobis

Dr. Ayan Mahalanobis

Committee:

Dr. Ayan Mahalanobis

Dr. Krishna Kaipa

मुक्ताईस, तिच्या ज्ञानादादास अन्
तिच्या ताटीच्या त्या अभंगास समर्पित.

Declaration

I hereby declare that the matter embodied in the report entitled **McEliece-type Cryptosystems Over Quasi-Cyclic Codes** 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. Ayan Mahalanobis** and the same has not been submitted elsewhere for any other degree.

Upendra Kapshikar

Acknowledgments

I would like to express my sincere gratitude towards Dr. Ayan Mahalanobis for his constant help and guidance. I would like to mention how grateful I am for having been given the freedom to try out new things. I thank Dr. Krishna Kaipa for his frequent feedback and monitoring of the project. I appreciate his interest and the time that he put in for this project. I would like to thank IISER Pune for giving me this opportunity. I would specially like to thank Mathematics Department at IISER Pune for providing computing facilities through Bhaskara Lab and other computing facilities. Finally, I would like to thank my family and friends for their support.

Abstract

In 1994, Peter Shor came up with a quantum algorithm that can factor integers efficiently. Along with the world of computations, this revolutionised the field of cryptography as one of the most popular protocols, RSA, assumes hardness of this problem. Though no practical quantum computers existed by then, it was evident that we are in dire need of a cryptosystem which will remain secure even in the era of quantum computers.

Interestingly, Shor's algorithm can be generalised, in a loose sense, to a higher class of problems commonly refereed to as hidden subgroup problems. The hidden subgroup problem can be stated as: given a function f over a group G that separates cosets [1] of an unknown subgroup H, the task is to find a generating set for subgroup H in $O(poly(log(\mathrm{G})))$ many measurements and post-processing steps. The class consists of a wide range of problems from order finding ($\mathbb{Z}/n\mathbb{Z}$) to graph isomorphism (S_n) and shotest vector problem in lattices (D_{2n}). One of the key ingradients of Shor's algorithm is quantum Fourier sampling (QFS). The succuess of the algorithm depends on how effective quantum Fourier sampling is in the corresponding case. The algorithm uses a quantum gate known as QFT (quantum Fourier transform) which performs the natural analogue of discrete Fourier transform over a group. Using this QFT, the quantum Fourier sampling produces a probabilty distribution over irreducible representations of group G (or on positions inside a matrix of represntaions if one does strong Fourier sampling) based on cosets of f. Basic idea to solve a hidden subgroup problem is reconstructing subgroup from obtained probability distribution. This process is feasible in every finite abelian group but the problem is open to a large extent for non-commutative groups. Naturally, this motivates us to look into cryptosystems where the underlying structure is non-commutative.

McEliece cryptosystem, developed by Robert McEliece in 1978, is one such cryptosystem where the underlying structure is non-commutative. McEliece cryptosystem is based on linear algebraic codes, one of the two most promissing enviromnents for post-quantum cryptography (other one being lattice based cryptography). The main advantages for McEliece system other than possibly being quantum secure is faster speed than RSA and ElGamal. Hence, there is a lot of interest in McEliece cryptosystem. However, most of the claim of security does not have a proper theoretical backing. The idea to come up with a proof of

[1] A fuction that is constant on a coset and different on different cosets

security for a McEliece cryptosystem was started by Dinh, Moore and Russel. The idea used by them can be traced back to the work by Julia Kempe and Aner Shalev where they charcterized hidden subgroups of S_n that are distinguishable from the identity subgroup. When two subgroups give probabilty distribution from QFS that are *very close* to each other then we can not differentiate between those two and hence, hidden subgroup problem can not be solved. Dinh, Moore and Russel extended this idea for identity subgroup and conjugate subgroups in the corresponding group for McEliece cryptosystem with goppa codes.

Though Niederreiter cryptosytem with goppa codes is shown to be quantum secure by Dinh, Moore and Russel, the system developed by goppa codes has two major drawbacks. One, the key sizes are too large and two, transmission rate is small, approximately 0.52. We extend this result to a new class of Quasi-cyclic codes of a certain kind. We show that it is impossible that the known quantum attack, using hidden subgroup problem and quantum Fourier sampling (or QFS) will break Niederreiter cryptosystem using a particular type of parity check matrices. We use a result provided by Dinh, Moore and Russel for this. To our knowledge, this is only the second variant of McEliece, after classical goppa codes, that is quantum secure.

Our proposed system has a better transmission rate than the previous variant. In most cases, key sizes are significantly smaller than the original variant. We also provide an algorithm that generates a parity check matrices satisfying the required set of conditions.

Chapter 1 gives brief summary of definitons and concepts from coding theory and cryptosystems based on algebraic coding theory. Quantum Fourier sampling and problems of hidden subgroup, hidden shift are reviewed in chapter 2. In chapter 3, we look into a closely related problem of combinatorial optimizaion where we are trying to find a class of QCCs that have small automorrphism group and large enough minimal degree. For this probelm also, we give a class of such codes and provide an algorithm that gives generator matrices for such codes. Chapter 4 provides a Niederreiter variant that is both classically and quantum secure.

Contents

Chapter 1

Coding Theory and Public Key Cryptosystems

In 1978, Robert McEliece [1] came up with a cryptosystem based on algebraic coding theory. The system never gained much popularity due to its large key sizes. Despite of this major drawback, McEliece and similar cryptosystems have started to gain mathematicians' attention as it is believed to be quantum secure. Unlike RSA or ElGamal, McEliece cryptosystem is based on non-commutative structure of algebraic codes. Compared to traditional cryptosystems, McEliece cryptosystem has following advantages

a) It is fast. Faster than RSA or ElGamal.

b) It is believed to be quantum secure.

The major problem with the McEliece cryptosystem is its large key size which makes implementation difficult.

a) Key sizes huge.

b) Cipher-text becomes much larger than the plain-text because of the redundancy added by the encoding process.

In this chapter, we start with basic definition from coding theory. Then we describe encryption and decryption processes for both McEliece and Niederreiter cryptosystems. Some important classical attacks are reviewed and current specifications of parameters are given.

1.1 Coding Theory

Error correcting codes are broadly separated in two sections; linear codes and non linear codes. Due to their elegant mathematical structure, linear codes received major interest of the coding community. Traditionally, linear codes are categorized into two types; block linear codes and convolutional codes. As every McEliece cryptosystem is based on block linear codes, we look specifically into block linear codes. Our standard reference for coding theory will be Blahut [2, Chapter 3]. Most of our discussions are on binary linear codes, that is, codes over $\mathbb{F}_{2^l}$ for some integer $l \geq 1$ but the same theory can be applied over any finite field.

A q-ary linear code $\mathcal{C}$ of length n and rank k is a k dimensional linear subspace of $\mathbb{F}_q^n$

Definition 1.1.1. *Hamming weight: Hamming weight or simply weight w of a vector v, denoted by $w(v)$, is defined as the number of non-zero entries in v.*

The hamming weight sets up a canonical distance function over $\mathbb{F}_{q^n}$. This distance is called the hamming distance.

Definition 1.1.2. *Hamming distance $d_H(x, y) := w(x - y)$.*

The distance of code $\mathcal{C}$ is defined as the minimum of the distance between any two distinct codewords in $\mathcal{C}$. We denote this by $d(\mathcal{C})$ or simply d if code $\mathcal{C}$ is clear from the context.

$$d(\mathcal{C}) = \min_{x,y\in\mathcal{C},x\neq y} d_H(x, y)$$

.

Traditionally, such a code is denoted as $[n, k, d]$ Following is an easy lemma, proof of which can be found in any standard text such as [3, 4].

Lemma 1.1.1. *Let $\mathcal{C}$ be a linear code then*

$$d(\mathcal{C}) = \min_{x\in\mathcal{C},x\neq 0} w(x)$$

One of the important parameters of a code is its error correction capacity.

Definition 1.1.3. *Error correction capacity t : Let $\mathcal{C}$ be $[n,k,d]$ code. Then the error correcting capacity is the largest integer t such that for any $y \in \mathbb{F}_q^n$ there exists a unique $c \in \mathcal{C}$ such that $d_H(y,c) \leq t$. The ratio $\frac{t}{n}$ is known as the error correction rate of the code $\mathcal{C}$.*

Error correction capacity and the minimum distance of a code are related to each other by an obvious relation $t = \left\lfloor \frac{d(\mathcal{C})-1}{2} \right\rfloor$.

A natural way to construct this k dimensional space $\mathcal{C}$ is by viewing it as a range space of some $k \times n$ matrix G known as the generator matrix of the code $\mathcal{C}$ and we say that G generates code $\mathcal{C}$. Alternatively, $\mathcal{C}$ can be constructed by kernel space of some matrix H of size $(n-k) \times n$. In such cases H is known as the parity check matrix. Of course, for a given code $\mathcal{C}$ there are many generator matrices and there are many parity check matrices. Code generated by H is known as the dual code of $\mathcal{C}$ and is denoted as $\mathcal{C}^{\perp}$. Note that both G and H are full rank matrices.

It is easy to check that $GH^T = 0$. Also dual of the dual is the same code,that is, $\left(\mathcal{C}^{\perp}\right)^{\perp} = \mathcal{C}$. So another way to define dual code is as follows:

Definition 1.1.4. *Dual code $\mathcal{C}^{\perp}$: Let $\mathcal{C}$ be a linear $[n,k]$ code over $\mathbb{F}_q$ then its $[n,n-k]$ dual $\mathcal{C}^{\perp}$ is defined as*

$\mathcal{C}^{\perp} = \{y \in \mathbb{F}_q^n$ *such that for all* $x \in \mathcal{C}$ $\langle x,y \rangle = 0\}$ *where* $\langle x,y \rangle$ *denotes inner product of* x *and* y *over* $\mathbb{F}_q$.

Now we give a few examples of codes.

(I) Hadamard Code $[n = 2^r, k = r, d = 2^{r-1}]$:

Hadamard Code is a linear code generated from a generator matrix whose i^{th} column is the number i written as a binary expression in r bits. Thus, there are 2^r columns corresponding to all binary numbers with r bits. The code has minimum distance $d = 2^{r-1}$ and hence can correct roughly about $t = 2^{r-2}$ errors.

$$\text{eg. } G = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

(II) Binary Goppa Codes $[n, k, 2t+1]$:

Construction of a binary Goppa code is done with a polynomial $g(x)$ of degree t over a finite field $\mathbb{F}_{2^m}$ of characteristic 2 without multiple zeros. Then construction of binary Goppa codes is done in the following way

Pick a set of n-points $\{p_1, p_2, \ldots, p_n : p_i \in \mathbb{F}_{2^m}, g(p_i) \neq 0\}$ i.e. none of the $p_i's$ are a root of the polynomial $g(x)$. Then parity check matrix for corresponding code is a product of two matrices; first one is a vandermonde matrix and the second one is a diagonal matrix. Construct following matrices V and D

$$V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ p_1 & p_2 & \cdots & p_n \\ p_1^2 & p_2^2 & \cdots & p_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ p_1^{t-1} & p_2^{t-1} & \cdots & p_n^{t-1} \end{bmatrix} \quad D = \begin{bmatrix} \frac{1}{g(p_1)} & 0 & \cdots & 0 \\ 0 & \frac{1}{g(p_2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{g(p_n)} \end{bmatrix}$$

$H = VD$.

Binary Goppa codes are the codes used in the original McEliece cryptosystems.

(III) Cyclic Codes:

Let $a = (a_1, a_2, \ldots, a_n)$ be an element of $\mathbb{F}_q^n$ then we define the right shift of a as $(a_n, a_1, \ldots, a_{n-1})$. A linear code is called a cyclic code if it is closed under right shifts.

Cyclic codes have a nice algebraic structure. The code can be considered as an ideal in the polynomial ring $R = \mathbb{F}_q[x]/(x^n - 1)$ which is a PID. In this ring multiplication by x to a codeword c results into a right shift of c. Being an ideal, the code space can be generated by a single polynomial $g(x)$, known as generator polynomial of the code.

(IV) Quasi Cyclic Codes (QCCs) $[n = m_1 p, k = m_2 p, d]$: Quasi-cyclic code is a simple generalization of a cyclic code. A linear code is considered from a class of quasi-cyclic codes if there exists an integer m such that code is closed under m right shifts. That is, if we denote R as a right shift operator then for all $c \in QCC$ we have $R^n(c) \in QCC$.

Such lowest m for which the code is closed under m right shifts is known as the index of the code. For cyclic codes, we have $m = 1$. QCCs can be constructed by generator matrices whose every block is a circulant matrix of size p and the number of such blocks in a particular row indicates its index. Similar to cyclic code , QCCs also have a nice algebraic structure. We will come back to QCCs later to talk in detail as our both variants are built over QCCs.

Consider a communication where we want a k-bit message $m = (b_1, b_2, \ldots, b_k)$. Now when we send this over a channel, there is a possibility that some b_i gets corrupted to b_i'. Notice that such corrupted vector is also a possible message that one could intend to send. Hence the receiver receives a wrong message. To overcome this difficulty, we add redundancy to the message eg. repeating the message bits thrice, i.e, for message $m = (b_1, b_2, \ldots, b_k)$ of k-bits we instead send a message of $3k$-bits $m' = (b_1, b_1, b_1, b_2, b_2, b_2, \ldots, b_k, b_k, b_k)$. Such codes are known as repetition codes. So even if some bit b_i gets corrupted to b_i' at one particular position, we have two other copies to recover the original message m. [1] In practice, much larger numbers than 3 are used. It is highly unlikely that the exact same corruption happens at multiple places.

This process of adding redundancies before sending a message is known as the encoding of an error correcting code. The process for a general linear error correcting code can be implemented as follows: Suppose we want to send messages of k-bits. So we have a message space of dimension k over $\mathbb{F}_q$. We embed this space into much higher dimensional space, say of dimension n, over $\mathbb{F}_q$. One of the ways this can be done is by using a compatible linear map G of full rank.

$$\begin{aligned} \varphi_G : \quad \mathbb{F}_q^k &\longrightarrow \mathbb{F}_q^n \\ x &\mapsto xG \end{aligned}$$

The map has a k-dimensional range with trivial kernel. As one might guess, our generator matrix G can be used as a linear map. The domain of the map φ_G is known as the message space while the range is known as the code space. This process is known as encoding and the map φ_G is known as the encoding function.

[1] Note that corrupted message is not in the possible sample set for receiver since theoretically receiver expects vectors with some particular pattern, namely each partition of 3-components has same element throughout the partition.

Once received, the vector from $\mathbb{F}_q^n$ has to be converted back to the original message vector in $\mathbb{F}_q^k$. This is done by the process known as decoding. A decoding process is a two step process. Usually, the first step consists of finding the nearest codeword to the received vector. And the second step is obtaining the message from this codeword. The second step can be done using common routines from linear algebra such as Gaussian elimination and generally is not a difficult task.

$$\psi : \ \mathbb{F}_q^n \ \xrightarrow{\psi_1} \ \mathcal{C} \xrightarrow{\psi_2} \ \mathbb{F}_q^k$$

$$\text{where } \psi_1(x) \mapsto c \text{ such that for all } c' \in \mathcal{C},\ d_H(x, c') \geqslant d_H(x, c)$$

$$\text{and } \psi_2 \text{ is an inverse of } \varphi_G \text{ when restricted on } \mathcal{C}.$$

Here, $e = x - c$ is known as the error and process of computing c from x, denoted by ψ_1 above is called the error correction. As this is the hardest computation in ψ, sometimes, ψ_1 is also referred as decoding of a linear code. This process of decoding is not easy for a random linear code. The problem of decoding a random linear code is known as the general decoding problem and is NP-complete [5]. But there are some codes for which this can be done easily eg. RS codes, Binary Goppa codes, cyclic codes, some subclasses of QCCs. Such codes are called decodable codes and it is said that they have a decoder.

1.2 Public Key cryptosystems based on coding theory

Using this hardness of the general decoding problem, Robert McEliece [1] proposed a cryptosystem based on algebraic coding theory. Later Niederreiter proposed his knapsack like variant based on the same principle. The rough idea behind both the systems can be explained as below :

Take a code $\mathcal{C}$ which has a good decoding algorithm. Transform this code into a new code $\mathcal{C}'$ with no apparent structure. A sender sends message using code $\mathcal{C}'$. Now as the code $\mathcal{C}'$ has no visible structure it is as good as a random linear code and no one can decode it. As you are the one who generated the system you have the transformation from $\mathcal{C}$ to $\mathcal{C}'$ you can revert the transformation and work in the frame of $\mathcal{C}$ and then decode successfully as $\mathcal{C}$

has a good decoder. Again, an eavesdropper can not decode this as he does not have a way to transform the system from $\mathcal{C}'$ to $\mathcal{C}$.

Now we move into McEliece and Niederreiter cryptosystems. We first explain there encryption and decryption algorithms and classical security. We address the question of quantum security in the chapter 2.

1.2.1 Description of the McEliece cryptosystem

Let M be a generator matrix for a $[n, k]$ linear code $\mathcal{C}$ for which a fast decoding algorithm exists. Let $\mathcal{E}$ be the number of errors that $\mathcal{C}$ can correct.

Private Key: (S,M,P) where $S \in GL_k(\mathbb{F}_2)$ and P is a $n \times n$ permutation matrix.

Public Key: $M' = SMP$.

Encryption:

Let $p \in \mathbb{F}_q^k$ be a k-bit plain-text. Corresponding cipher-text $c \in \mathbb{F}_q^n$ is obtained by calculating $c = pM' + e$ where e is a random error vector such that $wt(e) \leq \mathcal{E}$.

Decryption:

Received cipher-text c is decrypted in the following way:

Multiplying c by P^{-1} we obtain $cP^{-1} = (pM' + e)P^{-1} = (pSMP + e)P^{-1} = pSM + e_2$. Note that e_2 has same weight as e.

Now use the decoding algorithm for M on vector $SAM + e_2$ to obtain pS.

Multiply by S^{-1} to recover plain-text p.

A brief explanation for security:

Consider a communication where Alice wants to send a message to Bob. Similar to any public key cryptosystem, Bob generates his private key and computes its public counterpart.

Private key consists of three matrices S, M, P with M having a good decoder. The private key has a corresponding public key M'. Since our S *and* P are chosen randomly, the resulting matrix M' is also a random matrix with no structure and hence no efficient decoding with M' is possible. Clearly, every process in decryption is trivial for Bob and he can decrypt easily. An eavesdropper, however, can not do this because he has no good decoding algorithm for publicly known matrix M' and he does not have knowledge of S and P which is essential.

Another interesting question regrading security of the system is, *'what happens if M is known ?'* The problem is same as finding transformation (S, P) between two codes. It is believed that this problem is not easy to solve. Decision version of this, that is, finding if two codes are equivalent is NP-hard and graph isomorphism problem can be reduced to it; so, if one solves the code equivalence problem, one of the problems that received the most attention in last few decades by theoretical computer science community can be solved efficiently [6]. This problem of finding transformation between codes or generator matrices of equivalent codes remains intractable. But this presents a possible window of attack from a quantum computer as this transformation problem, known as scrambler-permutation problem, can be modelled as a hidden shift problem and further can be reduced to a hidden subgroup problem where quantum Fourier sampling can come into play. Thus, it becomes essential to make sure that McEliece or its variants resist this quantum Fourier sampling attack.

1.2.2 Niederreiter Cryptosystem

Let H be a $(n-k) \times n$ parity matrix for a [n,k] linear code $\mathcal{C}$ for which a fast decoding algorithm exists. Let $\mathcal{E}$ be the number of errors that $\mathcal{C}$ can correct.

Private Key: (S,H,P) where $S \in GL_k(\mathbb{F}_2)$ and P is a permutation matrix of size n.

Public Key: $H' = SHP$.

Encryption:

Let p be a n-bit plain-text with weight at most $\mathcal{E}$. Corresponding cipher-text c of $n-k$ bits is obtained by calculating $c = H'p^T$.

Decryption:

Compute $y = S^{-1}c$. Thus $y = HPp^T$.

By linear algebra find a z such that $Hz^T = y$. As $y = HPp^T$ we have $z - pP^T \in \mathcal{C}$.

Now use fast decoding on z with H to get pP^T and thus recover p.

1.2.3 Signature scheme

Initially it was thought that McEliece cryptosystem could not accommodate for signature scheme as it is not a commutative cryptosystem in the sense that order or role of encryption and decryption algorithms can not be altered. In other cryptosystems such as RSA, where encryption and decryption algorithms do commute, ready signature schemes are available. Later, in 2001, Courtois, Finiasz and Sendrier [7] came up with a signature scheme for Niederreiter cryptosystem.

1.3 Classical attacks

In this section we briefly go over the generic classical attacks. Most of the attacks are local attacks in the sense that they try to decrypt a given ciphertext. Complete breaks that completely recover private keys are extremely demanding and not possible.

The attacks trying to recover plaintext from the knowledge of ciphertext and public key are really hard due to the general decoding problem and are out of the discussion. Most of the classical attacks that stand a chance come under category called Information Set Decoding (ISD). There are two popular ISD attacks; one by Stern [8] and other by Lee and Brickell [9]. As mentioned in [10], ISD attacks are the best known classical attacks and hence considered as security level of the system.

One of the basic attacks was suggested by McEliece [1]. Lee and Brickell improved his attack and added an important verification step where attacker can confirm whether recovered message is the correct one. The strategy is based on repeatedly selecting k bits at random from an n-bit cipher-text in hope that none of the selected bits are part of the error. Similar attacks can also be implemented over Niederreiter cryptosystems. Lee and Brickell also provided a closed-form equation for complexity of the attack. The work factor for this attack can be given as

$W_j = T_j\left(\alpha k^3 + N_j \beta k\right)$ where,

$$T_j = \frac{1}{\sum\limits_{i=0}^{i=j} \frac{\binom{t}{i}\binom{n-t}{k-i}}{\binom{n}{k}}} \qquad \text{and} \qquad N_j = \sum_{i=o}^{i=j} \binom{k}{i}$$

Other attack is by Stern [8]. The basic idea behind this attack is to recover intentional vector by embedding public code space into some higher dimensional codespace. Let C' be the code generated by public key G' then Stern constructs a code given by generator matrix $G'' = \begin{bmatrix} G' \\ x \end{bmatrix}$. Bernstein, Lange and Peters [11] improved the attack using Markov Chain Modelling. This modified version of attack made the attack faster by a factor of 2^{12} and parameters for the system had to be readjusted.

For a given code, both McEliece and Niederreiter cryptosystem have same security [12] on the equation level. That means, if one tries to recover plain-text from cipher-text for a McEliece cryptosystem it is hard as doing the same for Niederreiter cryptosystem and hence ISD attacks have same strength when applied on McEliece or Niederreiter over a code with same parameters. Li, Deng and Wang [12] also analyzed both the systems under the attack by Lee and Brickell.

1.3.1 Parameters

Robert McEliece in his original work suggested parameters $n = 1024,\ k = 524,\ t = 50$. As mentioned before, after [11] these parameters were not secure. Bernstein suggested some new sets of parameters. We mention a few of them here.

(a) For Goppa codes and 80 bit security $n = 1632$, $k = 1269$, $t = 33$. The public key size in this case is 460647 $bits$.

(b) Without list decoding suggested set is $n = 2048,\ k = 1751,\ t = 27$. The public key size in this case is 520047 *bits*

(c) For 128-bit security $n = 2960,\ k = 2288,\ t = 56$. This parameter selection leads to public keys of size 1537536 *bits*.

From these parameter choices it is very clear that McEliece cryptosystem or its Niederreiter variant suffers two major drawbacks:

(I) Large public key sizes.

(II) Low transmission rate or encryption rate.

Various attempts have been made to overcome above problems but most of them turned out to be insecure. McEliece variants over Quasi-cyclic circulant codes [10] is one of the notable attempts in that direction. In this particular version, authors looked at QC-LDPC codes and put forth a cryptosystem which is similar to the McEliece cryptosystem with matrices S and P replaced by block circulant matrices. Though in this variant shorter key structure is possible and higher encryption rates can be achieved, one of the most important piece of the puzzle, quantum security of this variant is still missing. It is absolutely essential that if McEliece or its any variant were to replace any of the current popular systems such as RSA, ElGamal first priority should be its quantum security than key sizes and encryption rate.

In next few chapters we will look into quantum security of McEliece-type cryptosystems and then provide a Niederreiter variant that is quantum secure having high transmission rate. We also present a brief comparative analysis of key sizes and encryption rate to original McEliece cryptosystems.

Chapter 2

Quantum Fourier Sampling

In this chapter, we look at basics of quantum Fourier sampling. We begin by defining a problem in abelian group theory.

Problem 2.0.1. *Let f be a function from $\mathbb{Z}/N\mathbb{Z}$ to $\mathbb{Z}/N\mathbb{Z}$ defined as*

$$\begin{array}{rccc} f(x): & \mathbb{Z}/N\mathbb{Z} & \longmapsto & \mathbb{Z}/N\mathbb{Z} \\ & x & \mapsto & a^x \end{array} \quad \textit{for some constant } a \in (\mathbb{Z}/N\mathbb{Z})^\times \ .$$

We want to find period the of $f(x)$.

Classically, it is well known that this problem is very hard. Integer factorization problem can be reduced to above order finding problem and hence one can solve integer factorization problem by solving order finding problem. However, with quantum computer this problem can be efficiently solved. The algorithm that solves this problem was given by Shor [13] [14] in 1995. This algorithm is the first quantum algorithm for some practical problem. Before Shor's algorithm, there were a few interesting algorithms that showed glimpses of the power of a quantum computer [15] [16] but these problems have artificial flavor in them and have hardly little practical importance.

Shor's algorithm was a great boost to algorithmic computer science in itself. Along with solving an important question for cyclic groups this algorithm opened doors for much powerful mathematics. Naturally, this situation presented two questions in front of computer science community. One, can we solve this order finding problem in a more general setting, say an abelian group or more generally in any group and two, if we were to extend this order finding function to some class of functions what would be right choice for the problem and will we be able to solve this problem?

Interestingly, both the questions have a common answer, quantum Fourier Sampling (QFS). In some sense, QFS can be viewed as a general form of Shor's algorithm. Quantum Fourier Sampling is an important tool that acts behind the scene for almost all known quantum algorithms that offer exponential speed up compared to classical algorithms. It is the reason why Shor's and Simon's algorithms work. The order finding problem is a special case of what is known as a hidden subgroup problem.

2.1 The Hidden Subgroup Problem (HSP)

Definition 2.1.1. *Hidden Subgroup Problem: Let* G *be a group and f a function from* G *to a set* X. *We know that $f(g_0) = f(g_1)$ if and only if $g_0\mathrm{H} = g_1\mathrm{H}$ for some subgroup* H. *The problem is, given f find a generating set for the unknown subgroup*[1] H.

In particular, Shor's algorithm is a hidden subgroup problem over $\mathbb{Z}/N\mathbb{Z}$ with function $f(x) = a^x$. In this case hidden subgroup is $\mathrm{H} = \langle r \rangle$ where r is the order of a in $(\mathbb{Z}/N\mathbb{Z})^{\times}$.

Quantum Fourier Sampling is an algorithm which uses Quantum Fourier Transform as a building block. Before going into QFS we recall some of the facts form representation theory. Given a group G a matrix representation is a homomorphism $\rho : \mathrm{G} \longrightarrow GL_{d_\rho}(\mathbb{C})$ where $GL_d(\mathbb{C})$ is a space of $d \times d$ matrices over complex numbers. We denote set of all the irreducible representations of the group G by $\widehat{\mathrm{G}}$. So for every $\rho \in \widehat{\mathrm{G}}$ and every $g \in \mathrm{G}$; $\rho(g)$ gives us a $d_\rho \times d_\rho$ matrix and $\rho_{(i,j)}(g)$ will denote the the entry in i^{th} row and j^{th} column of $\rho(g)$. We stick only to finite groups and their complex irreducible representations. One of the very well known result from representation theory of finite groups states that $\sum_\rho {d_\rho}^2 = |\mathrm{G}|$.

Let $|\mathrm{G}| = N$. Fix an ordering $(g_1, g_2, \ldots, g_N)$ on elements of G. For a vector in $\mathbb{C}^N$, a

[1]The function f in the hidden subgroup problem is said to be separating cosets of H as f is constant on a each coset and different on different cosets.

general normalized state in basis $\mathcal{B}_1 = \{g_i;\ 1 \leqslant i \leqslant N\}$ can be described as

$$|\psi\rangle = \sum_{i=1}^{N} \alpha_i |g_i\rangle \text{ where } \alpha_i s \text{ are complex numbers such that} \sum_{i=1}^{N} |\alpha_i|^2 = 1.$$

Now consider another basis for $\mathbb{C}^N$ given by $\mathcal{B}_2 = \{(\rho, i, j)\ \rho \in \widehat{\mathrm{G}},\ 1 \leqslant i, j \leqslant d_\rho\}$. Clearly, $|\mathcal{B}_2| = N$ as $\sum_\rho d_\rho^2 = N$. A general normalized state in basis $\mathcal{B}_2$ is denoted as

$$|\psi\rangle = \sum_{\rho,i,j} \beta_{\rho,i,j} |\rho, i, j\rangle \text{ where } \beta'_{\rho,i,j} s \text{ are complex numbers such that} \sum_{\rho,i,j} |\beta_{\rho,i,j}|^2 = 1.$$

A quantum Fourier transform is a map that takes a normalized state in basis $\mathcal{B}_1$ to a normalized state in $\mathcal{B}_2$. Under this map the vector associated with g is $\sum_{\rho,i,j} \frac{\sqrt{d_\rho}}{\sqrt{|\mathrm{G}|}} \rho_{(i,j)}(g)|\rho, i, j\rangle$. Quantum Fourier transform can be viewed as $\mathcal{C}$ linear extension of the above association. Precisely, quantum Fourier transform on a general normalized state $\psi = \sum_l \alpha_l |g_l\rangle$ in basis $\mathcal{B}_1$ would give

$$QFT|\psi\rangle = \sum_l \alpha_l\ QFT|g_l\rangle = \sum_l \alpha_l \sum_{\rho,i,j} \frac{\sqrt{d_\rho}}{\sqrt{|\mathrm{G}|}} \rho_{(i,j)}(g)|\rho, i, j\rangle.$$

Measurement is a nonlinear operator used in almost every quantum algorithm. Measurement operator is defined with respect to a basis. Here we describe measurement with respect to an orthonormal basis only as both $\mathcal{B}_1$ and $\mathcal{B}_2$ described above form an orthonormal basis. For measurement in a general basis or more general measurement operator see [17], [18]. Consider an orthonormal basis for n dimensional space $\mathcal{B} = \{b_1, b_2, \ldots b_n\}$. Then a normalized general state in this basis can be represented by $|\psi\rangle = \sum_i \beta_i |b_i\rangle$. Measurement on state $|\psi\rangle$ gives b_i as its output with probability $|\beta_i|^2$. In other words, after measurement the state collapses to one of the basis element states; the probability that it falls on a particular state depends on coefficient of that basis element in the state before measurement.

After this short background on quantum computing, we are ready to describe quantum Fourier sampling. For further reading a reader can refer to [18, 20]

Algorithm 1 Quantum Fourier Sampling (QFS)

procedure QFS(G, f) ▷ QFS over group G with function f
  $|\psi\rangle = \sum_g |g, 0\rangle$
  apply U_f on $|\psi\rangle$ Let $|\psi_2\rangle = U_f|\psi\rangle$
    ▷ U_f is a two state unitary operator such that $U_f|x, y\rangle = |x, y \oplus f(x)\rangle$
  Measure in the second component to get $|\psi_3\rangle$
    ▷ Measurement of vector $|\phi\rangle = \sum_i \alpha_i|e_i\rangle$ gives output e_i with probability $|\alpha_i|^2$
  Apply QFT on first component of $|\psi_3\rangle$
  Measure $|\rho, i, j\rangle$ for strong Fourier sampling *OR*
   Measure $|\rho\rangle$ component for weak Fourier sampling
end procedure

Strong quantum Fourier sampling gives you output $|\rho, i, j\rangle$ for hidden subgroup H with probability given by

$$\mathsf{P}_{\mathrm{H}}\left(|\rho, i, j\rangle\right) = \frac{1}{|\mathrm{G}|}\sum_{g\in\mathrm{G}} \mathsf{P}_{g\mathrm{H}}\left(|\rho, i, j\rangle\right)$$

where

$$\mathsf{P}_{g\mathrm{H}} = \frac{d_\rho}{|\mathrm{G}||\mathrm{H}|}\left|\sum_{h\in\mathrm{H}} \rho_{i,j}\left(g\mathrm{H}\right)\right|^2$$

Details about quantum Fourier sampling and in general status about non-abelian hidden subgroup problem can be found in [19]. For basics of quantum computation and quantum Fourier sampling specifically from hidden subgroup point of view we refer [20]. For more broader view of quantum computation reader can use standard texts such as [17], [18]. The rough idea behind using this QFS to solve a hidden subgroup problem is to try and reconstruct H from P_{H}. We make this idea precise in next few sections. But before going there, let us briefly look at how hidden subgroup problem can be used to break McEliece cryptosystem or its variants. We define scrambler - permutation problem.

2.2 McEliece-type Cryptosystems and HSP

Problem 2.2.1. *Scrambler - Permutation Problem: Consider two $k \times n$ matrices M and M' over $\mathbb{F}_{q^l}$. It is known that they are related by equation $M' = SMP$ for some unknown $S \in GL_k(\mathbb{F}_q)$ and some unknown permutation matrix P of size n. The problem is to find S and P.*

Clearly, one of the ways to attack McEliece or Niederreiter cryptosystem is by solving the scrambler-permutation problem. For this attack, we assume attacker knows both M and M' and he is trying to recover remaining part of the private key. This attack is known as the scrambler - permutation attack. As far as the quantum attacks go, this is the only known way of attacking a McEliece cryptosystem. This structural attack is exactly same for a McEliece cryptosystem or a Niederreiter cryptosystem except that instead of finding a scrambler-permutation pair from generator matrix G to G' one has to find scrambler-permutation pair from parity check matrix H to H'. The algebraic structure of the problem remains the same. So, we present it in general form, to find a scrambler-permutation pair from a $k \times n$ matrix M to other $k \times n$ matrix M' keeping in mind $M = G$ and $M' = G'$ in case of McEliece while $M = H$ and $M' = H'$ in case of Niederreiter. To mean either a McEliece or a Niederreiter cryptosystem we use a broad term McEliece-type cryptosystems. In this attack, we assume M and M' known, the attack is to find A and B such that $AMB = M'$ with A and B coming form groups as defined before. Notice finding any A' and B' such that $A'MB' = M'$ will also make the attack successful.

Problem 2.2.2 (Hidden Shift Problem)**.** *Let* G *be a group. Let* f_0 *and* f_1 *be two functions from group* G *to a set* X*. Given* $f_0(g) = f_1(g_0 g)$ *for some unknown constant* g_0 *the task is to find a constant* $g_0 \in \mathrm{G}$*. Note that there can be many* g_0 *that satisfy the above condition. Hidden shift problem asks us to find any one of those constants.*

Let $M' = AMP$. A McEliece-type cryptosystem will be broken if we find one possible pair (A, P) from M and M'. Consider two functions from group $\mathrm{G} = GL_k(\mathbb{F}_2) \times S_n$ given by

$$f_0(A, P) = A^{-1}MP \tag{2.1}$$

$$f_1(A, P) = A^{-1}M'P \tag{2.2}$$

Then one can check that $f_1(A, P) = f_0((A_0^{-1}, P_0).(A, P))$, that is (A_0^{-1}, P_0) is the shift between f_0 and f_1. Hence, if one can solve the hidden shift problem over $\mathrm{G} = GL_k(\mathbb{F}_2) \times S_n$ he can break the McEliece-type cryptosystem.

The general procedure to solve this hidden shift problem is to reduce it to try and reduce it to a hidden subgroup problem. We can reduce the hidden shift problem with functions f_0 and f_1 defined above on the group $\mathrm{G} = GL_k(\mathbb{F}_2) \times S_n$ to the hidden subgroup problem over $(\mathrm{G} \times \mathrm{G}) \rtimes \mathbb{Z}_2$ [21, Section 2.2]. The hidden subgroup in this case is

$$\mathrm{K} = (((\mathrm{H}_0, s^{-1}\mathrm{H}_0 s), 0) \cup ((\mathrm{H}_0 s, s^{-1}\mathrm{H}_0), 1)) \tag{2.3}$$

where $\mathrm{H}_0 = \{(A, P) \in GL_k(\mathbb{F}_2) \times S_n : A^{-1}MP = M\}$ and s is a shift from f_0 to f_1.

In short, the scrambler-permutation problem is one of the key ways to attack a McEliece-type cryptosystems. This problem can be formulated as a hidden shift problem which further can be reduced to a hidden subgroup problem. So we can attack McEliece type cryptosystems by trying to solve a hidden subgroup problem over $(\mathrm{G} \times \mathrm{G}) \rtimes \mathbb{Z}_2$ with $\mathrm{G} = GL_k(\mathbb{F}_2) \times S_n$.

2.3 Successful Quantum Fourier Sampling

In the previous section we saw that solving the hidden subgroup problem as a standard way to attack the a McEliece-type cryptosystem. An interesting question is, when is the hidden subgroup problem hard to solve? This way we can ensure the security of a McEliece-type cryptosystem against known quantum attacks.

We briefly sketch thought behind effectiveness of QFS. Algorithm of QFS in a general scenario and its use for solving a hidden subgroup problem is very well explained in [19]. Arguments particular to McEliece-type cryptosystems and corresponding hidden subgroup problem are in [21]. The standard model of QFS yields a probability distribution as a function of the hidden subgroup. The basic idea here is if two subgroups H_1 and H_2 yield probability distributions $\mathsf{P}_{\mathrm{H}_1}$ and $\mathsf{P}_{\mathrm{H}_2}$ such that $\mathsf{P}_{\mathrm{H}_1}$ and $\mathsf{P}_{\mathrm{H}_2}$ are *very close* to each other then QFS will not give us enough information to solve the hidden subgroup problem. The concept of closeness of two probability distributions can be captured by setting up a norm on the space of probability functions. To our knowledge, J. Kempe and A. Shalev [22] were the first to introduce this beautiful idea. They used total variation norm. So, if two probability functions have total variation distance between them less than or equal to $log^{-\omega(1)}|\mathrm{G}|$ then we say that those two distributions are non-distinguishable. Using this definition, they provided a necessary condition to distinguish a subgroup of S_n from the trivial subgroup $\langle e \rangle$. Later Dinh, Moore and Russel [21] extended this result with keeping McEliece-type group structure under consideration. Their result can be viewed as an analysis of a hidden subgroup problem over the group $\mathrm{G} = (GL_k(\mathbb{F}_2) \times S_n)^2 \rtimes \mathbb{Z}_2$, the group structure for McEliece-type cryptosystems. Instead of using total variation distance, they use L_1 distance. Other key difference that can be considered between two definitions is Kempe and Shalev defined it for weak Fourier sampling while Dinh, Moore and Russel defined it for strong Fourier sampling. Also to account for all the conjugate subgroups Dinh, Moore and Russel took expectation over all the conjugate subgroups along with expectation over irreducible complex representations of group G denoted as ρ. Here they demonstrate a case when the hidden subgroup H can not be distinguished from either its conjugate subgroups $g\mathrm{H}g^{-1}$ or the trivial subgroup $\langle e\rangle$.

First note that weak Fourier sampling gives same distributions for all the conjugate subgroups that is P_{H} is same as $\mathsf{P}_{g\mathrm{H}g^{-1}}$. Hence weak Fourier sampling can not differentiate a subgroup from its conjugate subgroup and thus it suffices to look at strong Fourier sampling. Dinh, Moore and Russel [21], inspired from J. Kempe and A. Shalev [22] define distinguishability of a subgroup H by strong Fourier sampling.

Definition 2.3.1. *Distinguishability of a subgroup on strong QFS*

$$\mathcal{D}_{\mathrm{H}} := \mathbf{E}_{\rho,g}\|\mathsf{P}_{g\mathrm{H}g^{-1}}(\cdot|\rho) - \mathsf{P}_{\langle e\rangle}(\cdot|\rho)\|_1$$

A subgroup H *is called indistinguishable by strong Fourier sampling if* $\mathcal{D}_{\mathrm{H}} \leq log^{-\omega(1)}|\mathrm{G}|$.

The real $\mathcal{D}_{\mathrm{H}}$ is nothing but the expectation of L_1 distance between probability distribution of conjugate subgroups and the trivial subgroup.

Note that if a subgroup H is indistinguishable according to this definition then by Markov's inequality for all c, $\|\mathsf{P}_{g\mathrm{H}g^{-1}}(\cdot|\rho) - \mathsf{P}_{\{e\}}(\cdot|\rho)\|_{t.v.} \leq log^{-c}|\mathrm{G}|$; which is analogous to definition provided by J. Kempe and A. Shalev [22] for indistinguishability of a subgroup by weak Fourier sampling.

Now we state a few definitions which will be used to establish quantum security of McEliece-type cryptosystem.

Definition 2.3.2. $Aut(M) = \{P \in S_n$ *such that there exists* $A \in GL_k(\mathbb{F}_q),\ AMP = M\}$.

Definition 2.3.3. *The minimal degree of a* $G \leqslant S_n$ *acting on set of* n *symbols is defined to be minimum number of elements moved by a non-identity element of the group* G.

Definition 2.3.4. *Consider a* $k \times n$ *matrix* M, *we define* T_M *for matrix* $M = [I_k|M^*]$ *as*

$$T_M = \{\mathcal{P}_1 \in S_k \text{ such that there exists } \mathcal{P}_2 \in S_{n-k} \text{ with } \mathcal{P}_1 M^* \mathcal{P}_2 = M^*\ \}.$$

Theorem 2.3.1. *[21, Theorem 4]: Assume* $q^{k^2} \leqslant n^{an}$ *for some constant* $0 < a < 1/4$*. Let* m *be the minimal degree of the automorphism group* $Aut(M)$*. Then for sufficiently large* n*, the subgroup* K*,* $D_K \leqslant O(|K|^2 e^{-\delta m})$*, where* $\delta > 0$ *is a constant.*

In the above theorem, the subgroup K is the hidden subgroup for McEliece-type cryptosystems that we stated earlier in this chapter. Details of the proof can be found in [21]. For a matrix M of full column rank, $|K| = 2|Aut(M)|^2$ [21]. Hence if $|Aut(M)|^4 e^{-\delta m} \leq log^{-\omega(1)}|\mathrm{G}|$ then K is indistinguishable making scrambler-permutation attack using QFS infeasible. Thus if a $k \times n$ matrix M with minimal degree m is such that $|Aut(M)|^4 e^{-\delta m} \leq log^{-\omega(1)}|\mathrm{G}|$ then we can not find a scrambler-permutation pair and hence system remains secure against quantum Fourier sampling.Later we use this result for parity check matrix H to show that our Niederreiter cryptosystem is secure against this hidden subgroup attack.

Chapter 3

A McEliece Variant

In this chapter, we describe a new variant of McEliece cryptosystem based on quasi-cyclic codes. We then show that these codes have small automorphism group and large minimal degree. Due to a Dinhtheorem by Dinh, Moore and Russell [21], that we looked at in the previous chapter, it becomes a natural direction to look for codes having automorphism groups with small size and large minimal degree. We would like to point out that our system is not necessarily quantum secure as it does not follow the require condition for theorem to hold, that is, $q^{k^2} \leqslant n^{\alpha n}$ for some $0 < \alpha < 1/4$. Though our system is not necessarily quantum secure, it presents an interesting mathematical problem of combinatorial optimization where we are looking to reduce the automorphism group size and increase the minimal degree. Quasi-cyclic codes represent an important class of block linear codes and particularly, $\frac{m-1}{m}$ quasi-cyclic codes and $\frac{1}{m}$ codes have received some special attention [23]. So, it becomes an important question if we can set up McEliece variant based on these quasi-cyclic codes satisfying required bounds on automorphism group size and minimal degree. Apart from, mathematical interests, this variant has encryption rate much higher than original McEliece. We show such construction over codes of rate $\frac{m-1}{m}$ and then note that similar construction can be used to construct $\frac{1}{m}$ codes. In the next chapter, we provide a similar construction of a Niederreiter variant which is indeed quantum secure.

3.1 Quasi-Cyclic Codes

Definition 3.1.1. *Cyclic code: A code $\mathcal{C}$ is called a cyclic code if it is closed under right shifts i.e. for all $c = (c_0, c_1, c_2, \ldots, c_{n-1}) \in \mathcal{C}$ we have $c' = (c_n, c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}$.*

A quasi-cyclic code (QCC), a simple generalization of the cyclic code, is such that any cyclic shift of a codeword by m symbols gives another codeword of QCC. If $m = 1$ the code is a cyclic code. We are particularly interested in $\frac{m-1}{m}$ rate codes. More specifically, our system is based on $\frac{m-1}{m}$ rate codes over $\mathbb{F}_2$ in this chapter and over $\mathbb{F}_{2^l}$ in the next chapter. Such codes along with Quasi-cyclic codes of rate $\frac{1}{m}$ are studied in great detail in [23].

Definition 3.1.2. *Circulant matrix: A $p \times p$ matrix C is called circulant if every row, except for the first row, is a circular right shift of the row above that.*

A typical example of a circulant matrix is

$$\begin{bmatrix} c_0 & c_1 & \cdots & c_{p-1} \\ c_{p-1} & c_0 & \cdots & c_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{bmatrix}$$

Now we state a couple of relevant results about circulant matrices and cyclotomic polynomials. Each lemma is easy to check involving basic ring definitions and Chinese remainder theorem for rings so proofs are skipped.

Lemma 3.1.1. *Let $\mathcal{C}_n$ denote class of all circulant matrices of size n. The class $\mathcal{C}_n$ forms a commutative ring under usual matrix addition and multiplication.*

Lemma 3.1.2. *The Ring $\mathcal{C}_n$ is isomorphic to the ring $\frac{\mathbb{F}_q[x]}{(x^n-1)}$. Furthermore if n is prime and $p \neq q$ then this isomorphism decomposes as*

$$\mathcal{C}_p \xrightarrow{\sim} \frac{\mathbb{F}_q\left[x\right]}{(x-1)} \times \frac{\mathbb{F}_q\left[x\right]}{\Phi_p(x)}$$

where $\Phi_p(x)$ is p^{th} cyclotomic polynomial.

A rate $\frac{m-1}{m}$ systematic Quasi-Cyclic code has an $p \times mp$ parity check matrix of the form

$H = [I_p|C_1|C_2|\cdots|C_{m-1}]$ where each C_i is a circulant matrix of size p and I_p is identity matrix of size p. For compactness we denote this as $H = [I|C]$ with keeping in mind that C is an array of circulants and we denote $C = \textsc{Array}[C_1, C_2, \ldots, C_{m-1}]$. Alternatively, these codes can be defined using generator matrices [24]. In this case, generator matrix takes the following form:

$$G = \left[\begin{array}{c|c} & C'_1 \\ & C'_2 \\ \mathrm{I} & \vdots \\ & C'_{m-1} \end{array}\right]$$

Again for compactness we denote this as $G = [I|C]$ with understanding that C is a stack of circulants and we denote C as $C = \textsc{Stack}[C'_1, C'_2, \ldots, C'_{m-1}]$.

In a recent work [24] a way to generate generator matrices for such codes over $\mathbb{F}_2$ is presented. Since these generator matrices are in systematic form one can easily construct parity check matrix from generator matrix. Regarding the codes over extension fields [23, chapter 6] shows that Quasi-cyclic codes over extension fields can be MDS (maximum distance separable) codes. As the name suggest MDS codes can achieve large minimum distance and hence no low weight codewords. This plays an important role for classical security of the system against classical attacks such as Stern's attack and Lee-Brickell attack. Though [23] presents examples of MDS codes with rate $\frac{1}{m}$, this does present a case for study quasi-cyclic codes of rate $\frac{m-1}{m}$ with large minimum distance. For more details about quasi-cyclic codes a reader can refer to [23, 24].

3.1.1 Decoding

Quasi-cyclic codes are well studied and well established codes and depending on how one constructs them there are various decoders available. We briefly mention some of them here. [23, Appendix B] presents some ML (majority logic) decodable QCCs. Another new and interesting way of decoding quasi-cyclic codes using Gröbner basis formulation can be found in [25].

3.2 Our McEliece variant

Now we are ready to describe our McEliece variant over quasi-cyclic codes of rate $\frac{m-1}{m}$. Our McEliece variant over $\mathbb{F}_2$ has a generator matrix $M = [I|C]$ with $C =$Stack$[C_1, C_2, \ldots, C_{m-1}]$ satisfying following conditions:

(I) Size of each circulant is a prime p, i.e., each circulant is a $p \times p$ matrix for some prime p.

(II) At least one of the $C_i s$ is invertible i.e. there exists $i < m$ suth that $C_i \in GL_p(\mathbb{F}_2)$.

(III) Given any two columns c_{i_0}, c_{i_1} of C, there is at most one index j with $c_{i_0}[j] = c_{i_1}[j] = 1$; that is, both the columns can have non-zero entry simultaneously at maximum one position. Some authors refer to this condition as no more than one overlapping 1s.

(IV) Let t be the weight of a column of C and t_r be a weight of a row of C then $t \cdot t_r \leqslant p-1$.

Now we prove bounds on $Aut(M)$ and minimal degree using a sequence of lemmas. First we just point out some relation between columns of matrices.

Let $P \in Aut(M)$ then for some A we have $A\,[I|C]\,P = [A|AC]\,P = [I|C]$. Hence, $[A|AC]$ have same set of columns as $[I|C]$ possibly in different order.

Remark 3.2.1. *Every column of A and AC is either a column of C or a column of I. Also no column of A is same as column of AC, in fact, no two columns of $[A|AC]$ are identical.*

Assume for the whole discussion that every column of C has weight t.

Lemma 3.2.1. *Let $\{v_1, v_2, \ldots, v_t\}$ be set of t distinct columns where each v_i comes either from I or from C such that at least 2 columns are from C then $\Sigma_{i=1}^t v_i$ is of weight at least 2.*

Proof. Suppose v_1, v_2 are from C. Hence $v_1 + v_2$ has weight at least $2t-2$ from condition (III) on C (as at most one entry from each column can get converted to 0). Now each of the remaining $t-2$ columns $v_3, v_4, \ldots, v_t$ can reduce the weight by at most 2 as it can reduce weight of v_1 by at most 1 and weight of v_2 by at most 1. Thus weight of $\Sigma_{i=1}^t v_i$ is at least $2t-2-2(t-2) = 2$. $\square$

Lemma 3.2.2. *Let $\{v_1, v_2, \ldots, v_t\}$ be a set of t distinct columns where each v_i comes from either I or C such that $\Sigma_{i=1}^{t} v_i$ is weight 1. Then only possible combination is 1 column of weight t and $t-1$ columns of weight 1. Moreover, each column of weight 1, including the resultant $\Sigma_{i=1}^{t} v_i$, should have 1 in the same place as those of weight t column.*

Proof. Clearly, if all $v_i's$ are weight 1, then $\Sigma_{i=1}^{t} v_i$ would be weight t. Now, if at least two $v_i's$ are weight t then $\Sigma_{i=1}^{t} v_i$ can not be weight 1 from *Lemma* 3.2.1. The condition on position of 1 is easy to check so we skip the proof. □

Remark 3.2.2. *Since columns of C have weight t, any column of AC is an addition of t columns of A. Moreover, every column of A contributes to such t_r additions where t_r is the weight of a row of C.*

Theorem 3.2.3. *If $P \in Aut(M)$ with C satisfying condition (II) and (III) then for any corresponding A, AC can not have a column of weight 1.*

Proof. Suppose there is a column (say ac_1) of AC with weight 1. As columns of AC are obtained by adding t columns of A, there exists a set $\{a_1, a_2, \ldots, a_t\}$ of t distinct columns such that $\Sigma a_i = ac_1$.

From Lemma 3.2.2, one column should be weight t and rest of the columns must be weight 1. Let a_1 be weight t column and $a_2, a_3, \ldots, a_t$ be weight 1 and each weight 1 column matches with weight t column.

Now since a_1 must be involved in t_r such additions ($t_r \geq 2$), there exists another set $\{a_1, a_2', a_3', \ldots a_t'\}$ such that $a_1 + a_2' + a_3' + \cdots + a_t' = ac_2$ is another column of AC. As ac_2 is column of AC it must be of weight 1 or weight t.

Observe that ac_2 can not be weight 1. Because if ac_2 was weight 1, then from Lemma 3.2.2 it should match 1, with a_1. The only columns satisfying such condition are $a_2, a_3, \ldots, a_t, ac_1$. But ac_2 can not be equal to either of those since column of AC can not be equal to any column of A or any other column of AC, this follows from Remark 3.2.1.

Now only possibility is ac_2 has weight t. Now we analyze this possibility in two cases.

Case 1: All $a_2', a_3', \ldots, a_t'$ have weight t. This leads to a contradiction as now the columns in the addition are weight t, all of them come from C. Thus we have one column of C as sum of other columns of C. Contradiction to condition (II).

Case 2: There is a column of weight 1 in addition (say a'_2). Now we have

$a_1 + a'_2 + a'_3 + \cdots + a't = ac_2$

Therefore, $a_1 + ac_2 + a'_3 + a'_4 + \cdots + a'_t = a'_2$. Contradiction to Lemma 3.2.1 as a_1, ac_2 have weight t and a'_2 has weight 1.

Thus we have ruled out every possibility for ac_2. Hence our assumption that there was a column of weight 1 in AC is wrong. Hence, every column of AC is weight t. □

Corollary 3.2.4. *If C satisfies conditions (II),(II) and $M = [I|C]$ then every $P \in Aut(M)$ is a direct sum of permutation matrix of size mp and a permutation matrix of size p that is P can be expressed as $P = P_1 \oplus P_2$ where P_1 is a $mp \times mp$ permutation matrix and P_2 is a $p \times p$ permutation matrix.*

Proof. Since AC has no column of weight 1; A must be permuted to get identity matrix and AC must be permuted to get C. Thus P permutes first mp columns within themselves and last k columns within themselves. Therefore, $P = P_1 \oplus P_2$ with corresponding sizes. Moreover, $A = {P_1}^{-1}$ and $ACP_2 = C$. □

Lemma 3.2.5. *If M satisfies condition (II) and (III) then $|Aut(M)| = \#(\mathcal{P}_1, \mathcal{P}_2)$ satisfying $\mathcal{P}_1 C \mathcal{P}_2 = C$.*

Proof. From corollary 3.2.4; $A = P_1^{-1}$ and $ACP_2 = C$. Therefore, $P_1^{-1}CP_2 = C$ and satisfies required equation with $\mathcal{P}_1 = P_1^{-1}$ and $\mathcal{P}_2 = P_2$. Similarly, one can go back to show other side. Thus, $Aut(M)$ is in one-to-one correspondence with $(\mathcal{P}_1, \mathcal{P}_2)$ solutions for $\mathcal{P}_1 C \mathcal{P}_2 = C$. □

Corollary 3.2.6. *If M satisfies condition (II) and (III) then $|Aut(M)| = \#\mathcal{P}_2$ satisfying $\mathcal{P}_1 C \mathcal{P}_2 = C$.*

Proof. Notice that since no two rows of C are identical $C\mathcal{P}_2$ are identical as $\mathcal{P}_2$ just permutes columns. Hence there is at most one way to permute rows to get back C. Hence for every P_2 there is at most one $\mathcal{P}_1$. Since $\mathcal{P}_2$ satisfies the equation, we have a unique P_1 corresponding to $\mathcal{P}_2$. □

So due to corollary 3.2.6 problem of finding size of the automorphism group is reduced to finding number of $\mathcal{P}_2$ solutions to

$$\mathcal{P}_1 C \mathcal{P}_2 = C \tag{3.1}$$

Here, we state a theorem by Burnside [26, Theorem 3.5B].

Theorem 3.2.7. Burnside *Let* G *be a subgroup of* $Sym(\mathbb{F}_p)$ *containing a* $p-$*cycle* $\mu : \xi \mapsto \xi + 1$. *Then* G *is either 2-transitive or* $\mathrm{G} \leq AGL_1(\mathbb{F}_p)$ *where* $AGL_1(\mathbb{F}_p)$ *is the affine group over* p.

We use this theorem to prove our main result

Theorem 3.2.8. *If* M *satisfies conditions (I), (II) and (III) then* $|Aut(M)| \leq p(p-1)$

Proof. Let G be the set of $\mathcal{P}_2$ that satisfy equation (3.1). It is an easy check that G forms a group. Also we can check that $\mathcal{P}_2 = \mu \in \mathrm{G}$ as it satisfies the equation (3.1) with $\mathcal{P}_1$ being block diagonal with μ^{-1} as every block. So using Burnside's theorem we can say that G is a subgroup of $AGL_1(\mathbb{F}_p)$ if it is not doubly transitive and the its size is less that or equal to $p(p-1)$. Proof that G is not doubly transitive follows from condition (IV). □

Lemma 3.2.9. *If* $t \cdot t_r \leqslant p - 1$ *then* G *is not doubly transitive.*

Proof. Let $\mathcal{S}$ be the set of x such that there exists a row r in C which contains non-zero entries at positions 0 and x that is there exists a row r such that $r[0] = r[x] = 1$. Now clearly $|\mathcal{S}| \leqslant t \cdot t_r$. Hence, $|\mathcal{S}| < p$ and there exists $y \in \{0, 1, 2, \ldots, p-1\}$ such that $y \notin \mathcal{S}$. Denote first row of C by r_0. Let x_0, y_0 be distinct positions such that $r_0[x_0] = r_0[y_0] = 1$. Now one can notice that no element $\mathcal{P}_2 \in \mathrm{G}$ can send (x_0, y_0) to $(0, y)$ as $C\mathcal{P}_2$ needs to have same set of rows as C. After action of such $\mathcal{P}_2$ first row has 1's at positions 0 and y but no row can have 1's at positions 0 and y the reason being $y \notin \mathcal{S}$. Thus G is not 2 transitive. □

Lemma 3.2.10. *Minimal degree of* G *is at least* $p - 1$.

Proof. As $P = P_1 \oplus P_2$ number of points moved by P is greater than or equal to number of points moved by P_2. Now assume P_2 fixes at least two points. Then $P_2 = I_p$ as $P_2 \in AGL(\mathbb{F}_p)$. And as for every P_2 there is at most one corresponding P_1, $P_1 = I_{(m-1)p}$ making $P = I_{mp}$. Thus non identity P can not have corresponding P_2 fixing more than one point and hence every non identity P moves at least $p - 1$ points. □

Algorithm 2 An algorithm that generates required generator matrix

procedure *Generate_C*(p, m, t_r)
  $Available\_set = [0, 1, \ldots, p-1]$
  $\mathcal{A} = [\,]$
  repeat m **times** $\{$
  while size of Current_set $\leq t_r$ **do**
    randomly choose $\alpha_1 \in$ Available_set
    Current_set.append$[\alpha_1]$
    remove α_1 from Available_Set
    for $(\alpha_2, \alpha_3) \in Current\_set \cup \mathcal{A}$ **do**
      remove $\alpha_2 + \alpha_3 - \alpha_1$ *modulo* p from Available_set
    end for
  end while
  $\mathcal{A}.append\,(Current\_State)$ $\}$
  construct circulant C_i having $\mathcal{A}[i]$ as its first column
  return $C = \textsc{Stack}\ [C_0, C_1, \ldots, C_{m-1}]$
end procedure

We will now prove correctness of the algorithm. We claim that output of the algorithm is a C satisfying condition (III). Let $\mathcal{A} = \{\alpha_j\}$ denote positions such that $c_0[\alpha_j] = 1$ where c_0 is the first column of C.

Lemma 3.2.11. *Let $\mathcal{A}' = \{\alpha'_j\}$ denote positions such that $c_N[\alpha'_j] = 1$ where c_N denotes the $N+1\,th$ column of C then $\{\alpha_j + N \text{ mod } p \mid \alpha_j \in \mathcal{A}\} = \{\alpha'_j \mid \alpha'_j \in \mathcal{A}'\}$.*

Lemma 3.2.12. *If condition (III) is not satisfied by C then there exists $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ such that $\alpha_4 = \alpha_1 + \alpha_3 - \alpha_2 \text{ mod } p$ such that α_1, α_2 belong to the same circulant C_i and α_3, α_4 belong to the same circulant C_j* [1].

Proof. Without loss of generality we can assume that column c_0 and column c_N have more than one overlaps. So there exists α_1 such that $c_0\,[\alpha_1] = c_N\,[\alpha_1] = 1$. Now from second equality and above lemma $\alpha_1 = \alpha_2 + N \text{ mod } p$ for some $\alpha_2 \in \mathcal{A}$. Similarly second overlap will give you $\alpha_3 = \alpha_4 + N \text{ mod } p$. Solving both the equations we get $\alpha_4 = \alpha_1 + \alpha_3 - \alpha_2 \text{ mod } p$. $\square$

Our algorithm iteratively constructs circulants $C_0, C_1, \ldots, C_{m-1}$ so that $\alpha_4 \neq \alpha_1 + \alpha_3 - \alpha_2$. In this way we ensure that C generated as an output of algorithm satisfies condition (III).

[1] C_i can be same as C_j that is it is possible that $i = j$ where the case represents both the overlaps happening in the same circulant

We end this chapter by showing how to achieve other conditions. Condition (I) is trivial as we just need to choose a prime p. We now move towards condition (II). By Lemma 3.1.2 we know that $\mathcal{C}_p$ is isomorphic to direct product of two rings. First part $\frac{\mathbb{F}_q[x]}{(x-1)}$ is a field as $(x-1)$ is irreducible. So to ensure that image in this ring is non-zero we just have to ensure that weight of circulant is odd. So we get our first condition as t_r should be odd. For the second part of the product we use following lemma.

Lemma 3.2.13. *If 2 is a primitive root modulo p then $\Phi_p(x)$ is irreducible in $\mathbb{F}_2[x]$.*

Proof of this can be found in standard number theory text. So to make the second part a field we need to choose p such that 2 is primitive. One of the ways to do this is choose $p = 4q+1$ such that both p and q are prime. For condition (IV) notice that $t = m \cdot t_r$ which gives us $t \cdot t_r = {t_r}^2 \cdot m$. So we can satisfy condition (IV) by choosing $t_r \leqslant \sqrt{\frac{p}{m}}$.

In conclusion, we can generate a matrix C satisfying required conditions by running algorithm for prime p such that 2 is primitive mod p and choosing odd t_r less than $\sqrt{\frac{p}{m}}$ and thus we can construct $\frac{m-1}{m}$ codes with small automorphism group and minimal degree at least $p-1$.

Chapter 4

Quantum Secure Niederreiter Variant

This chapter is based on work done in [27]

4.1 Introduction

In chapter 2 we mentioned a way to ensure security of cryptosystem against quantum Fourier sampling. But before going there, we describe our variant and briefly go over its classical security. Our variant is based on Quasi-cyclic codes over $\mathbb{F}_{2^l}$ of rate $\frac{m-1}{m}$. The relevant definitions about Quasi-cyclic codes are in section 3.1 and more details are in [23].

4.1.1 Description of our Niederreiter cryptosystem

Description of the parity check matrix used for the proposed Niederreiter cryptosystem Recall that we are talking about $\frac{m-1}{m}$ quasi-cyclic codes over $\mathbb{F}_{2^l}$. For the cryptosystem to be quantum-secure the parity check matrix $\mathcal{H}$ for the $[n = mp, k = (m-1)p, d]$, $\frac{m-1}{m}$ quasi-cyclic code should satisfy the following conditions:

I Integers m, p, such that p is a prime and m is bounded above by a polynomial in p.

II The matrix $\mathcal{H}$ is of size $p \times mp$ over $\mathbb{F}_{2^l}$.

III The matrix $\mathcal{H}$ is of the form $[\,C_0 = I \,|\, C_1 \,|\, C_2 \,|\, \ldots \,|\, C_{m-1}\,]$, where each C_i is a circulant matrix of size p. Each C_i for $i > 0$ should contain an element from a proper extension of $\mathbb{F}_2$. Furthermore, we denote the matrix $\mathcal{H}$ as $[\,I \,|\, C\,]$ where C is the concatenation of the circulant matrices C_i, $i > 0$.

IV We define $T_{\mathcal{H}} = \{P_1 \in \mathrm{S}_p \mid \exists P_2 \in \mathrm{S}_{p(m-1)} \text{ such that } P_1 C P_2 = C\}$, where S_n is the symmetric group acting on n letters. It is easy to see that $T_{\mathcal{H}}$ is a permutation group action on p letters. The condition we impose on $\mathcal{H}$ is that $T_{\mathcal{H}}$ is not 2-transitive.

V No two columns of C are identical.

4.2 Classical Attacks

In this section we briefly go over the generic classical attacks against McEliece and Niederreiter cryptosystems. We also mention some attacks exploiting the circulant structures in the keys. Interestingly, Li *et al.* [12, Section III] proved that both McEliece and Niederreiter cryptosystems are equivalent in terms of classical security. The proof follows from the fact that the encryption equation for one can be reduced to the other. This implies the equivalence of security of both the cryptosystems for attacks that try to extract the plaintext from a ciphertext.

Most generic attacks over algebraic code based cryptosystems are *information set decoding attacks*(ISD). Two most popular ways of implementing ISD attacks are by Lee and Brickell [9] and Stern [8]. As mentioned by Baldi *et al.* [10] ISD attacks are the best known attacks with the least work factor as far as classical cryptanalysis is considered. Hence these work factors are considered as security levels for a McEliece and Niederreiter cryptosystems.

The basic idea behind one of the attacks was suggested by McEliece himself. Lee and Brickell [9] improved the attack and added an important verification step where attacker confirms that recovered message is the correct one. In this case, we are dealing with a McEliece cryptosystem over a $[n, k]$ linear code. The strategy is based on repeatedly selecting k bits at random from a n-bit ciphertext in hope that none of the selected bits are part of the error. Similar attacks can also be implemented over Niederreiter cryptosystems. Lee and Brickell also provided a closed-form equation for complexity of the attack. As our system is based on $(n = mp, k = (m-1)p, d_{min} = 2\mathcal{E} + 1)$ code the expression for minimal work factor (with $\alpha = \beta = 1$ as taken by Lee and Brickell) takes the following form

$$W_{min} = W_2 = T_2 \left((m-1)^3 p^3 + (m-1) p N_2 \right)$$

where $T_2 = \dfrac{1}{Q_0 + Q_1 + Q_2}$ and $Q_i = \binom{\mathcal{E}}{i} \binom{n-\mathcal{E}}{k-i} \, / \, \binom{n}{k}$ with $N_2 = 1 + k + \binom{k}{2}$.

In Table 4.1 we present numerical data for work factor for different values of parameters. Recently, Aylaj *et al.* [24] developed an algorithm to construct stack-circulant codes with high error correcting capacity which makes the proposed Niederreiter cryptosystem much more promising.

Other ISD attacks are based on a strategy given by Stern. To recover the intentional error vector e in a McEliece cryptosystem such strategies use an extension code C'' generated by generator matrix $M'' = \begin{bmatrix} M' \\ x \end{bmatrix}$. Bernstein *et al.* [11] later improved this attack. Probability of success and work factor for Stern's attack is described in [28]. In the Table 4.1 we also provide probability of success for parameters $l = 16$ and $A_w \approx n - k$. Both the parameters can be optimized further to obtain the least work factor but not much variation is seen as we change any of these parameters. With such low probabilities, it is clear that the work factor for Stern's attack is worse than the Lee-Brickell attack. Even when one considers improvements suggested by Bernstein *et al.* [11], Lee-Brickell's [9] attack seems to outperform the attack by Bernstein *et al.* as it produces speedup up to 12 times and hence the security of the system against the Lee-Brickell attack should be considered the security of the system. Key sizes should be devised according to that.

Another attack worth mentioning for quasi-cyclic codes is the attack on the dual code. This attack works only if the dual code has really low weight codewords and is often encountered only when sparse parity check matrices are involved. For example, McEliece with QC-LDPC [10]. Such attacks can easily be stopped by choosing codes that do not have low weight codewords. From the work of Aylaj *et al.* [24] this can be achieved.

4.3 Quantum security

After this discussion on classical security we now move towards quantum-security of the proposed McEliece and Niederreiter cryptosystems which is one of the major goal of this chapter. Before moving towards quantum security we recall definitions and theorem by Dinh, Moore and Russell.

Definition 4.3.1. *$Aut(M) = \{P \in S_n$ such that there exists $A \in GL_k(\mathbb{F}_q),\ AMP = M\}$*

Definition 4.3.2. *The minimal degree of a $G \leqslant S_n$ acting on set of n symbols is defined to be minimum number of elements moved by a non-identity element of the group G.*

Definition 4.3.3. *Consider a $k \times n$ matrix M , we define T_M for matrix $M = [I_k|M^*]$ as*

$$T_M = \{\mathcal{P}_1 \in S_k \text{ such that there exists } \mathcal{P}_2 \in S_{n-k} \text{ with } \mathcal{P}_1 M^* \mathcal{P}_2 = M^* \ \}$$

Theorem 4.3.1. *[21, Theorem 4]: Assume $q^{k^2} \leqslant n^{an}$ for some constant $0 < a < 1/4$. Let m be the minimal degree of the automorphism group $Aut(M)$. Then for sufficiently large n, the subgroup K, $D_K \leqslant O(|K|^2 e^{-\delta m})$, where $\delta > 0$ is a constant.*

The idea behind security of the system is when distinguishability of the hidden subgroup K denoted as D_k becomes less than $\log^{-\omega(1)} |\mathrm{G}|$, quantum Fourier sampling can not give us the hidden subgroup and hence an attacker can not find a scrambler permutation pair.

4.3.1 Proof of indistinguishability of the hidden subgroup

We prove indistinguishability in a sequence of lemmas.

Lemma 4.3.2. *Let $P \in \mathrm{Aut}(\mathcal{H})$ then $P = P_1 \oplus P_2$ where P_1 is a block of size p and P_2 is a block of size $(m-1)p$ and $P_1 \oplus P_2$ is a block diagonal matrix of size $mp \times mp$ with the top block P_1 and the bottom block P_2.*

Proof. Let $P \in \mathrm{Aut}(\mathcal{H})$, from the definition of automorphism there is an A such that $A\mathcal{H}P = \mathcal{H}$. This implies that

$$A\,[\,I\,|\,C\,]\,P = [\,A\,|\,AC\,]\,P = [\,I\,|\,C\,]\,.$$

As action of right multiplication by a permutation matrix permute columns, the above equality shows that $[\,A\,|\,AC\,]$ has same columns as $[\,I\,|\,C\,]$ possibly in different order. Now since every column of C contains an entry from a proper extension of $\mathbb{F}_q$, no column of A can be column of C. This forces A to have same columns as I and AC to have same columns as that of C. Hence P permutes first p columns within themselves and last $(m-1)p$ columns in themselves. Hence every $P \in \mathrm{Aut}(\mathcal{H})$ can be broken into $P_1 \oplus P_2$ so that P_1 acts on I and P_2 acts on C. □

An obvious corollary follows:

Corollary 4.3.3. *If $P \in \mathrm{Aut}(\mathcal{H})$ then the corresponding $A \in \mathrm{S}_k$.*

The next lemma is central to quantum-security. It gives us a way to move from $\mathcal{H}$ to C by noting, the P_1 from the $P \in \mathrm{Aut}(\mathcal{H})$ is actually a member of $T_{\mathcal{H}}$.

Lemma 4.3.4. *The cardinality of $\mathrm{Aut}(\mathcal{H})$ is the cardinality of the set $\{(P_1, P_2)\}$ that satisfy $P_1 C P_2 = C$ where $\mathcal{H} = [\,I\,|\,C\,]$ as defined earlier.*

Proof. The proof follows from the fact, if P belongs to $\mathrm{Aut}(\mathcal{H})$, then $P = P_1 \oplus P_2$. Then $A[\,I\,|\,C\,]P = [\,I\,|\,C\,]$ translates into $A[\,I\,|\,C\,](P_1 \oplus P_2) = [\,I\,|\,C\,]$. Keeping in mind the block diagonal nature of P, it follows that $[\,AIP_1\,|\,ACP_2\,] = [\,I\,|\,C\,]$. Then $A = P_1^{-1}$ and $P_1^{-1} C P_2 = C$. This proves the lemma. □

The next lemma proves that for each P_1 there is at most one P_2.

Lemma 4.3.5. *Cardinality of the set $\{(P_1, P_2)$ that satisfy $P_1 C P_2 = C\}$ equals $|T_{\mathcal{H}}|$.*

Proof. Recall that $T_{\mathcal{H}} = \{P_1$ that satisfy $P_1 C P_2 = C\}$. So it suffices to show that for every P_1 there is at most one P_2. Since no two columns of C are identical, no two columns of $P_1 C$ are identical. Hence, there is at most one way to re-order them to get back C. Thus for every P_1 there is at most one P_2. □

Theorem 4.3.6 (Burnside [26, Theorem 3.5B])**.** *Let G be a subgroup of $Sym(\mathbb{F}_p)$ containing a p-cycle $\mu : \xi \mapsto \xi + 1$. Then G is either 2-transitive or $G \leq \mathrm{AGL}_1(\mathbb{F}_\mathrm{p})$ where $\mathrm{AGL}_1(\mathbb{F}_\mathrm{p})$ is the affine group over p.*

We prove a theorem on the size of the Automorphism group of $\mathcal{H}$.

Theorem 4.3.7. *If $\mathcal{H}$ satisfies conditions I,II and III then* $|\mathrm{Aut}(\mathcal{H})| \leqslant \mathrm{p}(\mathrm{p}-1)$.

Proof. From Lemma 4.3.4 and Lemma 4.3.5, the group $\mathrm{Aut}(\mathcal{H})$ has same size as $T_{\mathcal{H}}$. It is now easy to check that the circulant matrix μ with first row $[0, 1, 0, \ldots, 0]$ of size p belongs to $T_{\mathcal{H}}$. The corresponding P_2 will be a block diagonal $(m-1)p$ matrix with blocks of size p and each consisting of μ^{-1}. Now notice that the circulant matrix μ corresponds to the p-cycle $\xi \mapsto \xi + 1$. By our condition III, $T_{\mathcal{H}}$ is not 2-transitive. Now by Burnside's theorem $T_{\mathcal{H}} \leq AGL_1(\mathbb{F}_\mathrm{p})$. Thus $|\mathrm{Aut}(\mathcal{H})| \leqslant \mathrm{p}(\mathrm{p}-1)$. □

After this bound on the size of the Automorphism group we move towards the minimal degree of the Automorphism group.

Lemma 4.3.8. *The minimal degree of* $\mathrm{Aut}(\mathcal{H})$ *is bounded below by* $p-1$.

Proof. Notice that any $P \in \mathrm{Aut}(\mathcal{H}) = \mathrm{P}_1 \oplus \mathrm{P}_2$. By the twist, from $P \in \mathrm{Aut}(\mathcal{H})$ to $P_1^{-1} \in T_{\mathcal{H}}$, it is easy to see that $P_1 \in AGL_\mathrm{k}(\mathbb{F}_\mathrm{q})$. Then $P_1(x) = ax + b \pmod{q}$ for some $a, b \in \mathbb{F}_q$. If P fixes two distinct points, then $a = 1$ and $b = 0$ is the only possible solution. This corresponds to the identity element and thus a non-identity element can not fix more that one point. So minimal degree of $\mathrm{Aut}(\mathcal{H})$ is bounded below by $p-1$. □

We now prove the main theorem of this chapter.

Theorem 4.3.9. *Let p be a prime and m a positive integer bounded above by a polynomial in p, such that, $p \leq \frac{1}{4}m(\log m + \log p)$. Then the subgroup K (Equation 2.3) defined above is indistinguishable.*

Proof. We will use Theorem 4.3.1 in this proof. First note, the minimal degree is bounded below by $p-1$. Now it is well known that $|K| = 2|H_0|^2$ and $|H_0| = |\mathrm{Aut}(\mathcal{H})| \times |\mathrm{Fix}(\mathcal{H})|$. We have shown that $|\mathrm{Aut}(\mathcal{H})| \leq \mathrm{p}(\mathrm{p}-1)$ and it is easy to see that $|\mathrm{Fix}(\mathcal{H})| = 1$. Putting all these together, we see that $|K|^2 e^{-\delta p} \leq 4p^8 e^{-\delta p}$ for some positive constant δ. However, from the bound on the size of m, it is obviously true that $4p^8 e^{-\delta p} \leq (mp\log(mp))^{-\omega(1)}$ for large enough p.

Now, if $p \leq am(\log m + \log p)$, then $p^2 \leq amp(\log m + \log p)$ which gives $2^{p^2} \leq (mp)^{amp}$ for $0 < a < \frac{1}{4}$. This satisfies the premise of Theorem 4.3.1 and hence K is indistinguishable. $\square$

4.4 Construction of the required parity check matrix

Now we address the last question about the proposed Niederreiter cryptosystem, how to construct a matrix $\mathcal{H}$ satisfying conditions I, II, III and IV? Clearly, conditions I, II and III are trivial to set up and deserve no special attention. We suggest a particular way for construction of parity check matrix $\mathcal{H}$ so that condition IV is satisfied. It should be noted that there may be other ways to meet condition IV as well.

Choose a pair of distinct elements $a, b \in F_{q^l}$. Now construct $\mathcal{H}$ such that C_1 contains both a and b exactly once in each column and no other C_i contains both a and b. We restate this condition as our condition IV$'$. We could have replaced C_1 by any other C_i for $i > 1$ and the proof remains the same. For sake of simplicity we stick with C_1.

IV$'$ Two distinct elements $a, b \in F_{q^l}$ occurs as entries of C_1 exactly once in each column and no other C_i contain both a and b.

Lemma 4.4.1. *If the matrix $\mathcal{H}$ satisfies* IV$'$*, it also satisfies* IV*.*

Proof. Let $\mathcal{P}_1 \in T_{\mathcal{H}}$. From $\mathcal{P}_1 C \mathcal{P}_2 = C$ it follows that $\mathcal{P}_1 C$ should have the same set of columns as C but possibly in a different order. Let α denote the row of a in the first column of C_1 and β denote the row of b in the same column. Now notice that every column in C that contains both a and b contains them such that difference between rows of a and b is $\alpha - \beta$ mod p where p is the size of each circulant matrix. Now let $\sigma \in T_{\mathcal{H}}$ such that it sends β to α and α to β. It then follows from the fact that p is a odd prime, $\alpha = \beta$ which contradicts our assumption. Hence, $T_{\mathcal{H}}$ is not 2 transitive. $\square$

Condition V can be easily satisfied using brute force and other means and this completes the construction of a parity check matrix $\mathcal{H}$ satisfying I, II, III, IV and V and hence, a **Niederreiter cryptosystem that resists quantum Fourier sampling** is found.

Algorithm 3 An algorithm that generates required parity check matrix

procedure *Generate_H*(p, m)
    choose $a, b \in \mathbb{F}_{2^l}^{\times}$
    generate an array v of length $p-2$ with entries from $\mathbb{F}_{2^l} - \{a, b\}$
    Construct an array c_1 by concatenating array $[a, b]$ with the array v
    randomly permute entries within c_1
    Let I denote the array of length p; $I = [1, 0, 0, \ldots, 0]$
    vector list $= [I, c_0]$
    Let R denote the right shift operator
    construct an array x of length p over $\mathbb{F}_{2^l}$
    while *vector list* has length less than m **do**
        for $y \in$ *vector list* **do**
            for $0 \leqslant i \leqslant n-1$ **do**
                if $R^i(x) = y$ **then**
                    choose next x
                else
                    add x in the *vector list*
                    choose next x
                end if
            end for
        end for
    end while
    for $v_i \in$ *vector list* ; $0 \leqslant i \leqslant m-1$ **do**
        Construct a circulant matrix C_i with v as its first column
    end for
    return parity check matrix $H = \textsc{Array}[C_0, C_1, \ldots, C_{m-1}]$
end procedure

4.5 Advantages of the proposed cryptosystem

One of the prime advantages of our proposed cryptosystem is quantum-security. Apart from that it has high transmission rate which translated into high encryption rate. Current McEliece cryptosystem built on Goppa codes has transmission rate of about 0.52. For a McEliece cryptosystem its rate is same as that of the transmission rate of the underlying code and is $\frac{k}{n}$. Niederreiter cryptosystems have slightly different rates due to difference in their encryption algorithm. For a general cryptosystem its encryption rate or information rate can be defined as follows [29]:

Let $\mathcal{S}(C)$ denote possible number of plaintext and $\mathcal{T}(C)$ denote possible number of ciphertexts then information rate of the system is defined by

$$\mathcal{R}(C) = \frac{log\,(\mathcal{S}(C))}{log\,(\mathcal{T}(C))}.$$

This information rate can be viewed as amount of information contained in one bit of ciphertext.

Our proposed Niederreiter cryptosystem have encryption rate on the higher side. This gives our variant an edge over once those constructed on classical Goppa codes or with GRS codes (generalized Reed-Solomon codes).

As discussed before another problem with McEliece and Niederreiter cryptosystems is large key size. Circulant matrices seems like a good choice when it comes to key-sizes. Matrices are 2-dimensional objects but circulant matrices behave like a 1-dimensional object as they can be described by their first row. Though this circulant structure is lost in public key due to the scrambler-permutation pair, the size of the key still remains smaller than the conventional Niederreiter cryptosystem. Our system is slightly better than original Niederreiter cryptosystem because of the less number of rows in the public key matrices. With $p = 101$, this number is less than one-tenth of the original Niederreiter cryptosystem. Though there are two factors that increase size of matrices in our variant compared to original McEliece, one, our matrices have large number of columns; and two, our system is based on extension field $\mathbb{F}_{q^l}$ which makes the effective size of the matrix l times compared to McEliece which is based on $\mathbb{F}_2$. However, in most cases due to very less number of rows the net result indicates that our system requires shorter keys than original McEliece. For instance, at 80-bit security with $p = 101$ and $l = 3$ our keys are almost half of the keys corresponding to original McEliece at same security level. While at 256-bit security level with $p = 211, t = 40$ and $l = 3$ our system key size of about $\frac{1}{4}^{th}$ of the original McEliece.

Table 4.1: Parameters for the proposed Niederreiter cryptosystem

Security in bits	p	t	m_c	m_Q	m	Probability of success	Public Key Size		Rate
							No. rows	No. cols	
80-bits	101	15	17	35	35	2^{-132}	101	3535	0.60
		20	9	35	35	2^{-190}	101	3535	0.77
	211	35	4	62	62	2^{-398}	211	13082	0.71
		40	3	62	62	2^{-465}	211	13082	0.80
100-bits	101	15	40	35	40	2^{-136}	101	4040	0.61
		20	17	35	35	2^{-190}	101	3535	0.77
	211	35	5	62	62	2^{-398}	211	13082	0.71
		40	5	62	62	2^{-465}	211	13082	0.80
120-bits	101	15	95	35	95	2^{-171}	101	9595	0.67
		20	32	35	35	2^{-190}	101	3535	0.77
	211	35	8	62	62	2^{-398}	211	13082	0.71
		40	6	62	62	2^{-465}	211	13082	0.80
128-bits	101	15	134	35	134	2^{-184}	101	13534	0.70
		20	42	35	42	2^{-199}	101	4242	0.79
	211	35	9	62	62	2^{-398}	211	13082	0.71
		40	7	62	62	2^{-465}	211	13082	0.80
256-bits	211	35	98	62	98	2^{-443}	211	20678	0.75
		20	55	62	62	2^{-465}	211	13082	0.80

Chapter 5

Results and Conclusion

This thesis looks into two different variants of McEliece cryptosystem and corresponding NP-hard scrambler permutation problem from the point of view of security against quantum computers. In chapter 3 we attempted a problem of combinatorial optimization where the underlying code needs to have small automorphism group and large minimal degree so that we get $|Aut(M)|^4 e^{-\delta(m)} \leqslant log^{-\omega(1)}|\mathrm{G}|$. This does not produce a McEliece cryptosystem as the premise of the theorem for quantum security requires $q^{k^2} \leqslant n^{\alpha n}$ for some $0 < \alpha < \frac{1}{4}$, the problem still is mathematically strong enough to deserve some attention. We provide a cryptosystem satisfying suggested bounds on automorphism group and minimal degree along with the algorithm to construct parity check matrices for this variant. It would be interesting to see if one can relax the requirement $q^{k^2} \leqslant n^{\alpha n}$ so that we get a quantum secure McEliece over quasi-cyclic codes. Another direction of improvement could be refining condition for the construction. In particular, if condition (III) can be relaxed it can increase the class of codes significantly.

In chapter 4 we give a Niederreiter variant that is classically and quantum secure against current known attacks. In particular, we show that for our system the hidden subgroup coming from natural reduction of corresponding scrambler-permutation problem is indistinguishable by quantum Fourier sampling. We also show that our system has higher encryption rate and shorter keys compared to classical McEliece systems. One of the important problem that needs to be addressed is finding QCCs that fit the suggested parameter size. It would be interesting to see if the system remains classically secure if we use sparse keys. It is clear that the system remains secure against quantum computers as the group structure for the system remains the same. This is important because it could reduce key sizes substantially. Other than classical security, the real question is, can we construct codes using sparse parity check matrices simultaneously satisfying required conditions $(I) - (V)$ and retain error correction capacity of the system as par suggested parameters? Another problem of interest could be optimizing values of p, t, l so that we get high rates and lower sized keys.

Bibliography

[1] R. J. McElice, "A public key cryptosystem based on algebraic coding theory," tech. rep., Communications system research centre, NASA, Jan-Feb 1978.

[2] R. E. Blahut, *Algebraic codes for data transmission*. Cambridge University Press, 2003.

[3] R. Roth, *Introduction to Coding Theory*. New York, NY, USA: Cambridge University Press, 2006.

[4] W. C. Huffman and R. A. Brualdi, *Handbook of Coding Theory*. New York, NY, USA: Elsevier Science Inc., 1998.

[5] E. Berlekamp, R. McEliece, and H. Van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, 1978.

[6] E. Petrank and R. M. Roth, "Is code equivalence easy to decide?," *IEEE Transactions on Information Theory*, vol. 43, no. 5, pp. 1602–1604, 1997.

[7] N. T. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece-based digital signature scheme," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 157–174, Springer, 2001.

[8] J. Stern, "A method for finding codewords of small weight," in *International Colloquium on Coding Theory and Applications*, pp. 106–113, Springer, 1988.

[9] P. J. Lee and E. F. Brickell, "An observation on the security of McEliece's public-key cryptosystem.," in *Eurocrypt*, vol. 88, pp. 275–280, Springer, 1988.

[10] M. Baldi, M. Bodrato, and F. Chiaraluce, "A new analysis of the McEliece cryptosystem based on QC-LDPC codes," *Security and Cryptography for Networks*, pp. 246–262, 2008.

[11] D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Post-Quantum Cryptography. PQCrypto 2008*, pp. 31–46.

[12] Y. X. Li, R. H. Deng, and X. M. Wang, "On the equivalence of McEliece's and Niederreiter's public-key cryptosystems," *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 271–273, 1994.

[13] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.

[14] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pp. 124–134, Ieee, 1994.

[15] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proc. R. Soc. Lond. A*, vol. 439, no. 1907, pp. 553–558, 1992.

[16] D. R. Simon, "On the power of quantum computation," *SIAM journal on computing*, vol. 26, no. 5, pp. 1474–1483, 1997.

[17] N. D. Mermin, *Quantum computer science: an introduction.* Cambridge University Press, 2007.

[18] P. Kaye, R. Laflamme, and M. Mosca, *An introduction to quantum computing.* Oxford University Press, 2007.

[19] M. Grigni, L. Schulman, M. Vazirani, and U. Vazirani, "Quantum mechanical algorithms for the nonabelian hidden subgroup problem," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pp. 68–74, ACM, 2001.

[20] C. Lomont, "The hidden subgroup problem-review and open problems," *arXiv preprint quant-ph/0411037*, 2004.

[21] H. Dinh, C. Moore, and A. Russell, "Mceliece and niederreiter cryptosystems that resist quantum fourier sampling attacks," in *Annual Cryptology Conference*, pp. 761–779, Springer, 2011.

[22] J. Kempe and A. Shalev, "The hidden subgroup problem and permutation group theory," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1118–1125, Society for Industrial and Applied Mathematics, 2005.

[23] T. A. Gulliver, *Construction of quasi-cyclic codes.* PhD thesis, University of Victoria, 1989.

[24] B. Aylaj, M. Belkasmi, S. Nouh, and H. Zouaki, "Good quasi-cyclic codes from circulant matrices concatenation using a heuristic model," *International journal of advanced computer science and applications*, vol. 7, no. 9, pp. 63–68, 2016.

[25] A. Zeh and S. Ling, "Decoding of quasi-cyclic codes up to a new lower bound on the minimum distance," in *Information Theory (ISIT), 2014 IEEE International Symposium on*, pp. 2584–2588, IEEE, 2014.

[26] J. Dixon and B. Mortimer, *Permutation Groups.* Graduate Texts in Mathematics, Springer New York, 1996.

[27] U. Kapshikar and A. Mahalanobis, "A Quantum-Secure Niederreiter Cryptosystem using Quasi-Cyclic Codes," *ArXiv e-prints*, Mar. 2018.

[28] M. Hirotomo, M. Mohri, and M. Morii, "A probabilistic computation method for the weight distribution of low-density parity-check codes," in *International Symposium on Information Theory*, 2005.

[29] H. Niederreiter and C. Xing, *Algebraic Geometry in Coding Theory and Cryptography.* Princeton University Press, 2009.