QUANTUM ERROR CORRECTING CODES AND FAULT-TOLERANT QUANTUM COMPUTATION OVER NICE RINGS

A Dissertation

by

SANGJUN LEE

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Andreas Klappenecker
Committee Members,	Anxiao Jiang
	Eun Jung Kim
	M. Suhail Zubairy
Head of Department,	Dilma Da Silva

August 2017

Major Subject: Computer Science

ABSTRACT

Quantum error correcting codes play an essential role in protecting quantum information from the noise and the decoherence. Most quantum codes have been constructed based on the Pauli basis indexed by a finite field. With a newly introduced algebraic class called a nice ring, it is possible to construct the quantum codes such that their alphabet sizes are not restricted to powers of a prime.

Subsystem codes are quantum error correcting schemes unifying stabilizer codes, decoherence free subspaces and noiseless subsystems. We show a generalization of subsystem codes over nice rings. Furthermore, we prove that free subsystem codes over a finite chain ring cannot outperform those over a finite field. We also generalize entanglement-assisted quantum error correcting codes to nice rings. With the help of the entanglement, any classical code can be used to derive the corresponding quantum codes, even if such codes are not self-orthogonal. We prove that an R-module with antisymmetric bicharacter can be decomposed as an orthogonal direct sum of hyperbolic pairs using symplectic geometry over rings. So, we can find hyperbolic pairs and commuting generators generating the check matrix of the entanglement-assisted quantum code.

Fault-tolerant quantum computation has been also studied over a finite field. Transversal operations are the simplest way to implement fault-tolerant quantum gates. We derive transversal Clifford operations for CSS codes over nice rings, including Fourier transforms, SUM gates, and phase gates. Since transversal operations alone cannot provide a computationally universal set of gates, we add fault-tolerant implementations of doubly-controlled Z gates for triorthogonal stabilizer codes over nice rings.

Finally, we investigate optimal key exchange protocols for unconditionally secure key distribution schemes. We prove how many rounds are needed for the key exchange between any pair of the group on star networks, linear-chain networks, and general networks.

ACKNOWLEDGEMENTS

To be honest, I could never have finished this thesis without the support of many people. I would like to reflect on the people who have supported and helped me so much throughout my time at Texas A&M.

First and foremost I would like to express my sincere gratitude to my advisor Dr. Andreas Klappenecker for his support, motivation and patience. His guidance has always steered me toward the right direction whenever I struggled to make progress in research. I would also like to thank the rest of my committee members, Dr. Anxiao (Andrew) Jiang, Dr. Eun Jung (EJ) Kim, and Dr. M. Suhail Zubairy for their support and assistance through this period.

I am also grateful to Aunt Susan and Uncle Ken for their continuous encouragement and prayers. Last but not the least, I would like to thank my family: my mom and to my wife Min Sun and my son Hyun for supporting me spiritually throughout these past years.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Conventional computing systems are usually implemented using VLSI circuits in silicon technology. The size of silicon-based processing units is getting smaller each year. But this technology is going to reach its physical limits within the next few years [1, 2]. So we need to explore alternative technologies. New materials such as carbon-nano tubes [3] promise faster transistors. Optical processing units [4] promise higher bandwidths. Thus, the new circuit technologies offer a speed up compared to silicon-based devices. How large is this speed up going to be? We don't know yet. Likely, it is going to be a constant speed up compared to the current silicon-based technology. Are there other ways to obtain a much faster speed up?

Quantum computation is well known for outperforming conventional computation for certain problems. In fact, it is able to provide quantum algorithms to solve some hard problems. A famous example is the integer factorization problem. The classical computation systems cannot solve this problem in a polynomial time. By contrast, Shor's quantum algorithm [5] runs in polynomial time to find the prime factors of an integer. Another example is Pell's equation, one of the oldest problems in number theory. Given a positive nonsquare integer d, Pell's equation is $x^2 - dy^2 = 1$. The goal is to find the pair (x_1, y_1) that minimizes $x + y\sqrt{d}$ among infinitely many pair of integers satisfying the equation. Hallgren [6] showed how a quantum computer can find the solution in polynomial time. We do not yet have any polynomial time classical algorithm for this problem.

So what gives the quantum computation such powerful computational enhancements? Conventional computation is based on laws of classical mechanics. On the other hand, quantum computation exploits quantum physics for computation. A quantum bit, or shortly a qubit, is a unit to store the information. It has a form of a superposition of two basis, $|0\rangle$ and $|1\rangle$. The qubit gives us the information of the probabilities to observe $|0\rangle$ and $|1\rangle$.

The problem here is that building quantum information systems is quite hard. Quantum bits are so vulnerable to interactions with the environment surrounding them. They tend to yield errors due to decoherence. In other words, coupling with other quantum states corrupts the quantum information. To protect quantum system from such errors, we need to use quantum error correcting codes. Quantum error correcting codes are essential in quantum communication and quantum computation [7, 8, 9]. They can protect quantum information against noise in a quantum channel. Also, they can remedy decoherence effects in quantum memory. Quantum codes encode the physical quantum information in a mathematically-defined way. So, we could restore the information corrupted during the transmission or the computation.

We can also use quantum error correcting codes for fault-tolerant quantum computation. Suppose that we have a good quantum error correcting code. First, we encode quantum bits using this quantum code. Then, we perform logical quantum operations on the encoded quantum bits without decoding. In such a way, we can compute quantum information without any decoherence issues. It implies that we could perform quantum computation in a fault-tolerant way.

One way to construct quantum codes is the so-called Calderbank-Shor-Steane (CSS) code construction [10, 11]. This gives us a systematic way to construct the quantum code using classical codes. For constructing quantum versions of error correcting codes, Gottesman introduced the stabilizer formalism [12]. Currently, the stabilizer formalism is indispensable for the construction of most quantum codes. Stabilizer codes also allow us to derive quantum codes from classical error correcting codes. CSS codes are a special subclass of stabilizer codes. Stabilizer codes are simultaneous eigenspaces with $+1$ eigenvalues of all elements in stabilizer group. One drawback of stabilizer codes is that the corresponding classical codes need to be self-orthogonal codes.

Quantum error correcting scheme is the key technique for a fault-tolerant quantum computation [7, 9]. Suppose that the level of the error can keep being under a threshold. Then, we could perform any quantum operations without any concerns about the errors [13]. One major difficulty here is that the quantum gate could spread errors to other parts. Those affected parts might be in the same block or in the other block. To avoid these undesirable events, we need to consider a transversal operation. The transversal operation is in a simple form because it is a bit-wise operation. Unfortunately, having only transversal operations are not enough for a computational universality [14]. So we need an alternative approach to detour this issue like the one introduced in [15].

A finite field is the dominant algebraic class for constructions of quantum codes. A famous example of a finite field is a binary. Recently, Klappenecker [16] has introduced a wider algebraic class, a nice nearring. It is well suitable for the generalized Pauli basis. So we can use this ring to construct the quantum error correcting codes. A nice nearring has some distinct advantages over a finite field. This wider algebraic class has no restriction on the order, so it is not limited to powers of primes. Note that the order of a finite field is a prime power. Besides, a nice nearring can have a simpler arithmetic. A nice nearring has one special subclass, which is a nice ring. We have considered to use a nice ring to generalize a couple of quantum error correcting schemes so far. The first attempt was constructing the stabilizer codes over Frobenius rings [17]. We have put more effort into constructing other quantum schemes over a nice ring. In this thesis, we develop the theory of subsystem codes over nice rings [18] and entanglement-assisted quantum error correcting codes over nice rings

[19]. Also, we investigated fault-tolerant quantum computation over nice rings [20]. Those generalizations are the main results of this thesis. We will show them in the following chapters.

A fault-tolerant quantum computing system on its own may not be good enough in a real world. These days, connecting high performance computers with others offers a strong computing power. We may need to consider a distributed quantum computing system, too. This can give us more computational power based on the quantum operations. Moreover, it may ease one of the difficulties, a scaling-up issue. Many researchers have been trying to overcome this issue to build a quantum machine. Suppose we have some quantum computing systems communicating each other. One of the key features that we need to set up is a secure key exchange protocol. There have been a lot of efforts to investigate the secure key exchange protocols. BB84 is the widespread secure key exchange protocol on the quantum communication systems [21, 22]. There is also the protocol on the classical communication systems using Kirchhoff-Law-Johnson-Noise (KLJN) [23]. We focus on the optimal numbers of rounds for the secure key exchange on various networks [24]. In this thesis, we will give tight bounds for the optimal number of rounds.

This thesis has the following structure. In the next chapter, we provide the fundamental concepts and the mathematical notations. In Chapter 3, we show the generalization of the subsystem code over a nice ring. Furthermore, in Chapter 4, we prove the generalization of the entangled-assisted quantum error correcting code. Then, we discuss how to generalize a fault-tolerant quantum computation with transversal operations in Chapter 5. Finally, we show the optimal numbers of rounds for the secure key exchange on three types of networks in Chapter 6. Those are a star network, a network with a general topology, and a linear-chain network.

2. PRELIMINARIES

The fundamental concepts and basic notations are discussed in this chapter, so that the readers would have sufficient background knowledge and be ready to go through the following chapters without any hassles.

2.1 Nearrings and Rings

First of all, the definition of a nearring is introduced as follows.

A set N under an addition operation $+$ and a multiplication operation $\cdot$ is called a (left) nearring if it satisfies

1. $(N, +)$ is a (not necessarily abelian) group,
2. $(N, \cdot)$ is a semigroup,
3. and the left-distributive law $x(y + z) = xy + xz$ holds for all $x, y, z \in N$.

Its subclass, a ring, has the following definition.

A set R under an addition operation $+$ and a multiplication operation $\cdot$ is called a ring if it satisfies

1. $(R, +)$ is an abelian group,
2. $(R, \cdot)$ is a monoid,
3. and the distributive laws on both sides, $x(y + z) = xy + xz$ and $(y + z)x = yx + zx$, hold for all $x, y, z \in R$.

A nearring and a ring have fewer constraints than a finite field, so some nearrings and rings offer an arithmetic that is simpler to implement than finite field arithmetic.

2.2 Qudits

The qudits are the generalization of the qubits for the higher dimensions. Let R be a finite ring with q elements. We fix an orthonormal basis B of $\mathbb{C}^q$ and denote it by

$$B = \{|x\rangle | x \in R\}.$$

A quantum state $|\psi\rangle$ is denoted by a superposition of the orthonormal basis,

$$|\psi\rangle = \sum_{x \in R} c_x |x\rangle,$$

where $c_x \in \mathbb{C}$ for all $x \in R$ and $\sum_{x \in R} |c_x|^2 = 1$. The absolute square $|c_x|^2$ is the probability to observe $|x\rangle$ when measuring $|\psi\rangle_B$ in the computational basis B.

The qubit is an example of the qudit since $\mathbf{F}_2$ is a finite field, so a ring.

Example 1. *A qubit has an orthonormal basis* $\{|0\rangle, |1\rangle\}$ *of* $\mathbb{C}^2$*. A quantum state* $|\psi\rangle = c_0|0\rangle + c_1|1\rangle$*, where* $c_0, c_1 \in \mathbb{C}$ *and* $|c_0|^2 + |c_1|^2 = 1$.

A ring of integer modulo n, where n is an integer, is a good example for the qudit.

Example 2. *Let R be the ring of integer modulo 6, denoted* $\mathbf{Z}/6\mathbf{Z}$*. Then, an orthonormal basis of* $\mathbb{C}^6$ *is given by* $\{|0\rangle, |1\rangle, |2\rangle, |3\rangle, |4\rangle, |5\rangle\}$*. A quantum state $|\psi\rangle$ is of the form*

$$|\psi\rangle = \sum_{x \in \mathbf{Z}/6\mathbf{Z}} c_x |x\rangle,$$

where $c_x \in \mathbb{C}$ *for all* $x \in \mathbf{Z}/6\mathbf{Z}$ *and* $\sum_{x \in \mathbf{Z}/6\mathbf{Z}} |c_x|^2 = 1$.

2.3 Nice Nearrings and Nice Rings

In this section, we provide fundamental information about nice nearrings and their properties, see [16] for more details.

Let N be a nearring with q elements, where $2 \leq q < \infty$. We denote by

$$\{|x\rangle | x \in N\}$$

a fixed orthonormal basis of $\mathbb{C}^q$. Let χ be a character of $(N, +)$. For all $a, b \in N$, we define a shift operator $X(a) : \mathbb{C}^q \to \mathbb{C}^q$ and a multiplication operator $Z(b) : \mathbb{C}^q \to \mathbb{C}^q$ by

$$X(a)|x\rangle = |x + a\rangle, \qquad Z(b)|x\rangle = \chi(bx)|x\rangle,$$

for all $x \in N$.

A finite nearring is called nice if and only if there exists a character χ of $(N, +)$ such that $\mathcal{E} = \{X(a)Z(b)|a, b \in N\}$ is a nice error basis. The set $\mathcal{E}$ is called a nice error basis if and only if it contains the identity, is closed under multiplication up to scalars, and is an orthogonal basis with respect to the Hilbert-Schmidt inner product. If N is a nice nearring and $\mathcal{E}$ is a nice error basis indexed by N with respect to the character χ, then we call χ a generating character.

If a finite nearring is nice, then it satisfies the following properties.

Proposition 3 ([16]). *If N is a nice nearring, then*

1. *a generating character χ of $(N, +)$ is a linear and irreducible,*

2. *$(N, +)$ is an abelian group,*

3. *and $(N, \cdot)$ has a unique left identity.*

Proof. See [16, Proposition 3 and 6]. □

So a nice nearring has considerably more structure than a general nearring, but does not need to be right-distributive.

A nice error basis $\mathcal{E}$ can be extended to n components by tensoring,

$$\mathcal{E}^{\otimes n} = \{M_1 \otimes \cdots \otimes M_n \,|\, M_k \in \mathcal{E}, 1 \leq k \leq n\}.$$

This yields a nice error basis of $\mathbb{C}^{q^n}$. The elements in $\mathcal{E}$ generalize the Pauli matrices, and the nice error basis $\mathcal{E}^{\otimes n}$ is the generalization of the Pauli basis on n qubits.

A nice error basis is not a group, since it is not closed under multiplication. The group generated by a nice error basis is called an error group. In our case, the error group of $\mathcal{E}^{\otimes n}$ is given by

$$E_n = \{\omega^c X(a)Z(b) | a, b \in N^n, c \in \mathbf{Z}\}, \tag{2.1}$$

where $\omega = \exp(2\pi i/m)$ is a primitive mth root of unity and m is the exponent of the group $(N, +)$. This is a finite group with center $Z(E_n) = \{\omega^c 1 | c \in \mathbf{Z}\}$, where 1 is the identity.

If R is a distributively generated nice nearring, then R is called a nice ring. A distributively generated nearring is nice if and only if it is a finite Frobenius ring [16]. Finite Frobenius rings play a prominent role in classical coding theory.

Example 4. *For a qubit with a basis $\{|0\rangle, |1\rangle\}$ of $\mathbb{C}^2$, a shift operator $X(1)$ is called a bit-flip operator X and a multiplication operator $Z(1)$ is called a phase-flip operator Z, such that*

$$X|0\rangle = |1\rangle, \qquad X|1\rangle = |0\rangle,$$
$$Z|0\rangle = |0\rangle, \qquad Z|1\rangle = -|1\rangle.$$

Those operators can be represented in the form of matrices,

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Example 5. *Let R be the ring of integer modulo 6, denoted $\mathbf{Z}/6\mathbf{Z}$. For a qudit with a basis $\{|0\rangle, |1\rangle, ..., |5\rangle\}$ of $\mathbb{C}^6$, the shift operators and multiplication operators are defined by*

$$X(a)|x\rangle = |x + a \pmod 6\rangle, \qquad Z(b)|x\rangle = \exp(2\pi bx/6)|x\rangle,$$

where $x \in \mathbf{Z}/6\mathbf{Z}$.

2.4 Classical Error Correction

In this and subsequent sections, we provide some background information about classical and quantum error correction, see also [25].

Before discussing the theory of quantum error correction, we first recall a few facts from the theory of classical error correction. Classical linear codes can be used to construct a variety of quantum error correction codes. For understanding the stabilizer formalism, we provide some fundamental ideas about classical linear codes in this section.

We say that a linear code C encoding k bits of messages into n bits of codewords is an $[n, k]$ code. A way to encode information is specified by an n by k generator matrix G. Given the k bit message x, the encoded codeword is Gx. The columns of G are linearly independent, so the set of codewords is the vector space spanned by the columns of G.

For the error-correction of the code, we need to consider a parity check matrix.

The parity check matrix H is an $n-k$ by n matrix such that $Hx = 0$ for all codewords x of a code C. The rows of H are linearly independent. Suppose that we encode the message x such that the encoded codeword is $x' = Gx$. If an error e corrupts the codeword x' during the transmission, the receiver will get $x' + e$. Since x' is a codeword, $Hx' = 0$. When we multiply the received $x' + e$ by the parity check matrix H, we obtain $H(x' + e) = He$. We call this the error syndrome. The error syndrome is important for the error correction since it contains information about the error occurred such as which bits the error corrupted.

We provide an example of a $[7, 4]$ Hamming code for better understanding the procedure of the error correction.

Example 6. *A* $[7, 4]$ *Hamming code encodes* 4 *bits of information into* 7 *bits of a codeword, and is able to correct an error on any single bit. Its parity check matrix is*

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

For instance, let us suppose that we transmit a codeword $(1, 0, 1, 0, 1, 0, 1)^T$. *We can easily check that this is a valid codeword in a* $[7, 4]$ *Hamming code by multiplying the codeword by the parity check matrix* H. *Suppose that an error occur such that the fourth bit is flipped. Now the corrupted codeword is* $(1, 0, 1, 1, 1, 0, 1)^T$. *We now need to calculate the error syndrome, which is* $(1, 0, 0)^T$. *In this case, the error syndrome says that the fourth bit needs to be flipped to correct the error.*

2.5 Theory of Quantum Error Correction

We need quantum error correcting codes to protect the quantum information against noise. However, there are some aspects that we never had to consider when constructing classical codes. Classical information can be duplicated, but quantum information cannot be replicated by the no-cloning theorem. This means that we cannot construct a repetition code that replicates the message several times.

We also need to deal with the fact that errors are continuous. So detecting which error occurred seems to require infinite precision and resources. Luckily, however, it turns out that quantum errors can be discretized. To show this, we provide the quantum error correction conditions. The quantum error correction conditions determine whether a quantum error correcting code protects against a particular type of noise.

Theorem 7 (Quantum error correction conditions)**.** *Let C be a quantum code, P be the projector onto C, $\{E_i\}$ be the set of quantum errors. A necessary and sufficient condition for the existence of an error-correction operation is*

$$PE_i^\dagger E_j P = \alpha_{ij} P,$$

for some Hermitian matrix α of complex numbers.

Proof. See the proof of Theorem 10.1 in [25]. □

Suppose the quantum error is a linear combination of the E_i's. Then, this error is also correctable by the error correction operation that can recover from $\{E_i\}$. Therefore, we can say that it is sufficient to deal with a finite set of errors, the Pauli matrices, instead of the continuum of errors.

In classical error correction, we can observe the output and calculate the error syndrome to correct the errors. Observation in quantum system, however, may destroy the information stored in the quantum state. So, we have to calculate the error syndrome without observing the quantum state with the information directly.

With a help of ancilla states with the conditional quantum operations, it is possible to obtain the error syndrome without knowing any information of the quantum states.

2.6 Stabilizer Formalism and Stabilizer Codes

Stabilizer codes are an important class of quantum codes. To understand stabilizer codes, we need first to review the stabilizer formalism. The stabilizer formalism gives us a way to express the quantum code using the stabilizer, instead of the actual code space.

Suppose S is a subgroup of the error group E_n. Let $\text{Fix}(S)$ be the set of n qudit states that are fixed by every element of S. Then, we say that $\text{Fix}(S)$ is the code space stabilized by S, and S is the stabilizer of $\text{Fix}(S)$. In other words, $\text{Fix}(S)$ is the intersection of eigenspaces with $+1$ eigenvalue of every elements of S. There are some conditions that the stabilizer should satisfy for a non-trivial code space. Obviously, $-I$ should not be an element of S, otherwise $-I|\psi\rangle = |\psi\rangle$, which is a contradiction. In addition, the elements of S commute each other. If there are some elements of S which do not commute, then it leads to a contradiction, too.

The stabilizer S is described by its generators. Thus, if the quantum state is stabilized by the generators of S, then that state is stabilized by S. Let S be generated by $n-k$ independent and commuting elements from E_n, and $-I \notin S$. Then, the code space $\text{Fix}(S)$ stabilized by S has a dimension of k. We define an $[[n,k]]$ stabilizer code as the code space $\text{Fix}(S)$ stabilized by a subgroup S of E_n such that $-I \notin S$ and S has $n-k$ generators. For example, the Steane seven qubit code is a $[[7,1]]$ stabilizer code, and its stabilizer has 6 generators, $IIIXXXX, IXXIIXX, XIXIXIX, IIIZZZZ, IZZIIZZ$, and $ZIZIZIZ$. This code is indeed the CSS code, and we will see how to construct this using the CSS code construction in the next section.

To present $n-k$ generators of the stabilizer S, we construct an $n-k$ by $2n$ matrix. This matrix is called the check matrix. Each row of the check matrix corresponds to each generator of S. If the generator contains an I on some qudit, then the entries in the corresponding positions on both hand sides are 0. If it contains an $X(a)$, then the entry in the corresponding position on the left hand side is a and one on the right hand side is 0. If it contains a $Z(b)$, then the entry in the corresponding position on the left hand side is 0, and one on the right hand side is b. For instance, the generators of the stabilizer of the Steane seven qubit code is equivalent to the following check matrix.

$$\left[\begin{array}{ccccccc|ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right]$$

Suppose a stabilizer code Fix(S) is corrupted by an error $E \in E_n$. We need to distinguish three possible cases. (i) If the error E does not commute with an element of the stabilizer S, then Fix(S) is taken to an orthogonal subspace, and the error will be detected by measuring the generators of S. (ii) If $E \in S$, then there is nothing to worry, since this error does not affect the code space Fix(S) at all. (iii) The last case is when E commutes with all the elements of S, but is not a scalar multiple of an element in S. This type of errors is not correctable.

2.7 CSS Code Construction

The Calderbank-Shor-Steane codes, or the CSS codes, are a subclass of stabilizer codes. The CSS codes are very important since their constructions give us a systematic way to build stabilizer codes.

Suppose C_1 and C_2 are $[n, k_1]$ and $[n, k_2]$ classical linear codes such that $C_2 \subset C_1$ and C_1 and $C_2^{\perp}$ both correct t errors. An $[[n, k_1 - k_2]]$ CSS code is constructed as follows. Suppose x is any codeword in C_1. Then,

$$|x + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle.$$

Since the state $|x + C_2\rangle$ corresponds to the coset of C_1/C_2, this CSS is an $[[n, k_1 - k_2]]$ stabilizer code. The error correcting property of the classical code C_1 is utilized to detect and correct the addition errors, the bit flip errors in binary. The error correcting property of $C_2^{\perp}$ is to detect and correct the multiplication errors, the phase flip errors in binary.

The check matrix corresponding to the generators of the CSS code is defined as

$$\left[\begin{array}{c|c} H(C_2^{\perp}) & 0 \\ 0 & H(C_1) \end{array}\right].$$

We give an example of the Steane code for the CSS code construction.

Example 8. *We construct the Steane code using the* $[7, 4]$ *Hamming code* C*. Let* $C_1 = C$ *and* $C_2 = C^{\perp}$*. Since* $C^{\perp} \subset C$ *and* $C^{\perp}$ *is the* $[7, 3]$ *code, we can construct a*

$[[7,1]]$ *CSS code. The check matrix is*

$$\left[\begin{array}{c|c} H(C) & 0 \\ 0 & H(C) \end{array}\right] = \left[\begin{array}{ccccccc|ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right].$$

The corresponding generators are $IIIXXXX, IXXIIXX, XIXIXIX, IIIZZZZ$, $IZZIIZZ$, *and* $ZIZIZIZ$. *Thus, the stabilizer of the Steane code is generated by these generators.*

3. SUBSYSTEM CODES OVER NICE RINGS

In this chapter, we will show one of the contributions by the author about generalization of subsystem codes over nice rings, which was published in [18][1].

Subsystem codes are one of the promising quantum error correction schemes for the fault-tolerant quantum computation. The subsystem code formalism unifies the features of stabilizer codes, decoherence free subspaces, and noiseless subsystems [26], [27], [28]. The main idea of the subsystem code is to decompose a subspace C of the Hilbert space H into a tensor product of the subsystem A and the co-subsystem B and encode the quantum information in the subsystem [26], [27], [29] such that

$$H = C \oplus C^{\perp} = (A \otimes B) \oplus C^{\perp}.$$

Since all quantum information is stored in the subsystem A, there is no need to correct any errors affecting only the co-subsystem B.

As one of efforts to utilize nice error bases indexed by a nearring, the construction of stabilizer codes over finite Frobenius rings was studied in [17]. In this chapter, we provide a construction of subsystem codes over nice nearrings [18]. Thus, we generalize the results of [17] by (a) allowing for a more general nice nearring arithmetic, and (b) generalizing from stabilizer codes to subsystem codes.

Before discussing the main results, some notations used in this chapter are given below.

Let G be a group, and H a subgroup of G. The centralizer $\{g \in G \,|\, gh = hg \text{ for all } h \in H\}$ of H in G is denoted by $C_G(H)$. The center $C_G(G)$ of G is

[1] 2013 IEEE. Reprinted, with permission, from Sangjun Lee and Andreas Klappenecker, Subsystem codes over nice rings, 2013 IEEE International Symposium on Information Theory Proceedings (ISIT), July 2013

denoted by $Z(G)$. For $x, y \in G$, the commutator $[x, y]$ is defined as $x^{-1}y^{-1}xy$. Given subgroups X and Y of G, we define $[X, Y] = \langle [x, y] | x \in X, y \in Y \rangle$. As usual, we denote the commutator subgroup $[X, X]$ by X'.

Let N be a nice nearring. Suppose an error e in E_n is given by $\omega^c X(a)Z(b)$ for $a, b \in N^n$ and $c \in \mathbf{Z}$. Then the weight $\mathrm{wt}(e)$ of the error e is defined as the number of its tensor components which are not scalar multiples of the identity. Similarly, given $(a|b) \in N^{2n}$, we can introduce its symplectic weight by

$$\mathrm{swt}(a|b) = |\{i | a_i \neq 0 \text{ or } b_i \neq 0, 1 \leq i \leq n\}|.$$

Thus, we have the relation $\mathrm{wt}(\omega^c X(a)Z(b)) = \mathrm{swt}(a|b)$ between these two types of weights.

3.1 Construction of Subsystem Codes over Nice Nearrings

In this section, we give the construction of subsystem codes over nice nearrings. For this purpose, we use the more general theory of Clifford subsystem codes, see [29].

Let M be a normal subgroup of an error group E_n and χ an irreducible character of M. The inertia group of χ is given by

$$I_{E_n}(\chi) = \{g \in E_n \,|\, \chi(gng^{-1}) = \chi(n) \text{ for all } n \in M\}.$$

The inertia group of a character determines the parameters of the subsystem code in the following theorem from [29]:

Theorem 9 ([29]). *Let E_n be an error group such that $E_n' \subseteq Z(E_n)$, M a normal subgroup of E_n, and χ an irreducible character of M chosen such that the orthogonal* *projector*

$$P = \frac{\chi(1)}{|N|} \sum_{n \in N} \chi(n^{-1}) n$$

is nonzero.[2] *Then $C = image(P)$ is the corresponding Clifford code. The inertia group of χ is given by $I_{E_n}(\chi) = C_{E_n}(Z(M))$. If $C_{E_n}(Z(M)) = LM$ for some subgroup L of E_n such that $[L, M] = 1$, then C is a subsystem code $C = A \otimes B$ such that*

1. $\dim A = |Z(E_n) \cap M||E_n : Z(E_n)|^{1/2}|M : Z(M)|^{1/2}/|M|$,

2. $\dim B = |M : Z(M)|^{1/2}$.

An error e in E_n is detectable by subsystem A if and only if e is contained in the set $E_n \setminus (C_{E_n}(Z(M)) \setminus Z(L)M)$.

Proof. See [29, Theorem 2]. □

We will show that Theorem 9 can be used to construct subsystem codes over nice nearrings. For this purpose, we will prove that (a) the error group given in (2.1) satisfies $E_n' \subseteq Z(E_n)$ and (b) give a convenient condition on M which ensures that the inertia group factorizes into a central product of the form required by Theorem 9. The next two lemmas will establish the required facts.

Lemma 10. *Let N be a finite nice nearring. The error group $E_n = \{\omega^c X(a)Z(b) | a, b \in N^n, c \in \mathbf{Z}\}$ satisfies $E_n' \subseteq Z(E_n)$.*

Proof. From the definitions of the shift operator and the multiplication operator, we know that $X(a)X(b) = X(a+b)$ and $Z(a)Z(b) = Z(a+b)$ for $a, b \in N$. Furthermore, we have $\chi(ba)X(a)Z(b) = Z(b)X(a)$. Since $Z(E_n) = \{\omega^c 1 | c \in \mathbf{Z}\}$, the quotient group $E_n/Z(E_n)$ is isomorphic to N^{2n}. By Proposition 3, a nice nearring is an

[2] We can always find such a character. Notice that the parameters of the resulting code do not depend on the particular choice of χ.

abelian group under addition. In other words, $E_n/Z(E_n)$ is abelian, and it follows that $E_n' \subseteq Z(E_n)$, as claimed. □

Our next concern is to find a convenient sufficient condition which ensures that $C_{E_n}(Z(M))$ factors into a central product of the form LM for some subgroup L of E_n.

A subgroup H of a group is called c-closed if and only if $H = C_G(C_G(H))$ holds. This terminology is motivated by the fact that the centralizer map $x \mapsto C_G(x)$ yields a Galois connection on the lattice of subsets of the group G.

Lemma 11. *Let E_n be an error group. Let M be a c-closed normal subgroup of the error group E_n such that $MC_{E_n}(M)$ is c-closed. Then*

$$C_{E_n}(Z(M)) = MC_{E_n}(M).$$

Proof. Since $MC_{E_n}(M)$ and M are c-closed, we have

$$\begin{aligned} MC_{E_n}(M) &= C_{E_n}(C_{E_n}(MC_{E_n}(M))) \\ &= C_{E_n}(C_{E_n}(M) \cap C_{E_n}(C_{E_n}(M))) \\ &= C_{E_n}(C_{E_n}(M) \cap M) \\ &= C_{E_n}(C_M(M)) \\ &= C_{E_n}(Z(M)), \end{aligned}$$

which proves the claim. □

The existence of normal subgroups satisfying the hypothesis of the previous lemma follows from a well-known group-theoretic result by Chermak and Delgado, see [30]. We can now combine the previous results and formulate the construction of subsystem codes over nice nearrings in the following form:

Corollary 12. *Let E_n be the error group given in (2.1) over a nice nearring N. Let M be a c-closed normal subgroup of E_n such that $MC_{E_n}(M)$ is c-closed. Then there exists a subsystem code $C = A \otimes B$ such that*

1. $\dim A = |Z(E_n) \cap M||E_n : Z(E_n)|^{1/2}|M : Z(M)|^{1/2}/|M|$,
2. $\dim B = |M : Z(M)|^{1/2}$.

An error e in E_n is detectable by subsystem A if and only if e is contained in the set $E_n \setminus (MC_{E_n}(M) \setminus M)$.

Proof. By Lemma 10, the error group satisfies $E_n' \subseteq Z(E_n)$. By Theorem 9, the inertia group is of the form $C_{E_n}(Z(M)$. This inertia group factorizes into the central product

$$C_{E_n}(Z(M)) = MC_{E_n}(M),$$

with $[M, C_{E_n}(M)] = 1$, by Lemma 11. Furthermore, we have

$$\begin{aligned} Z(C_{E_n}(M)) &= C_{E_n}(C_{E_n}(M)) \cap C_{E_n}(M) \\ &= M \cap C_{E_n}(M) \\ &= Z(M) \\ &\leq M. \end{aligned}$$

Thus, the claim is obtained from Theorem 9 using $L = C_{E_n}(M)$ and the fact that $Z(L)$ is a subset of M. □

In the next section, we will remove the assumption on the normal subgroup M in the case of rings.

3.2 Construction of Subsystem Codes over Nice Rings

In this section, we will assume that N is a distributively generated nice nearring. A distributively generated nearring is nice if and only if it is a finite Frobenius ring [16]. Finite Frobenius rings play a prominent role in classical coding theory. We will relate the construction of subsystem codes over nice rings to classical codes over rings. We will use R instead of N to denote a nice ring.

Let R be a nice ring. For $u = (a|b)$ and $v = (a'|b')$ in R^{2n}, we define

$$\langle u|v\rangle = \chi(b \cdot a' - b' \cdot a).$$

We write $u \perp v$ if and only if $\langle u|v\rangle = 1$ holds. Thus, $X(a)Z(b)$ and $X(a')Z(b')$ commute if and only if $u \perp v$. For a subset $S \subseteq R^{2n}$, we define

$$\begin{aligned} S^{\perp} &= \{u \in R^{2n} \mid \langle s|u\rangle = 1 \text{ for all } s \in S\} \\ {}^{\perp}S &= \{u \in R^{2n} \mid \langle u|s\rangle = 1 \text{ for all } s \in S\} \end{aligned}$$

One can show that for a subgroup C of R^{2n}, we have

$$|C||C^{\perp}| = |{}^{\perp}C||C| = |R^{2n}|,$$

see [17, Lemma 6].

For a subgroup G of the error group E_n, we use the bar notation $\overline{G}$ to denote $G/Z(E_n)$.

A key element in the construction of subsystem codes is the decomposition of the inertia group of the character χ into a central product (the groups L and M in Theorem 9). The next lemma shows that this is always possible when R is a nice ring.

Without loss of generality, we may assume that the normal subgroup M of the error group E_n contains the center $Z(E_n)$ of the error group. If M does not contain $Z(E_n)$, then simply use the larger group $MZ(E_n)$,

Lemma 13. *Let R be a finite nice ring. If E_n is a set $\{\omega^c X(a)Z(b)|a, b \in R^n, c \in \mathbf{Z}\}$ and M is a normal subgroup of E_n, then $C_{E_n}(Z(M)) = MC_{E_n}(M)$.*

Proof. We first notice that $MC_{E_n}(M)$ is a subgroup of $C_{E_n}(Z(M))$, and that both groups contain the center $Z(E_n)$ of the error group. Thus, it suffices to show that the cardinality of the quotient groups $\overline{MC_{E_n}(M)}$ and $\overline{C_{E_n}(Z(M))}$ are the same. We have

$$\begin{aligned}
|\overline{MC_{E_n}(M)}| =& |\overline{M} + \overline{M}^{\perp}| \\
=& |\overline{M}||\overline{M}^{\perp}|/|\overline{M} \cap \overline{M}^{\perp}| \\
=& |\overline{M}||\overline{M}^{\perp}|/|\overline{Z(M)}| \\
=& |R^{2n}|/|\overline{Z(M)}| \\
=& |\overline{Z(M)}^{\perp}| \\
=& |\overline{C_{E_n}(Z(M))}|,
\end{aligned}$$

which proves our claim. □

From the previous two results, we can conclude the following theorem about the construction of subsystem codes over a nice ring.

Theorem 14. *Let R be a finite nice ring. Suppose the error group E_n is a set $\{\omega^c X(a)Z(b)|a, b \in R^n, c \in \mathbf{Z}\}$. If C is a Clifford code with data (E_n, ρ, M, χ) with $M \neq 1$, then C is a subsystem code $C = A \otimes B$ such that*

1. $\dim A = |Z(E_n) \cap M||E_n \colon Z(E_n)|^{1/2}|M \colon Z(M)|^{1/2}/|M|$,

2. $\dim B = |M : Z(M)|^{1/2}$.

An error e in E_n is detectable by subsystem A if and only if e is contained in the set $E_n - (MC_{E_n}(M) - M)$.

Proof. By Lemma 10, the error group E_n satisfies $E_n' \subseteq Z(E_n)$. Thus, the inertia group $I_{E_n}(\chi) = C_{E_n}(Z(M))$. By Lemma 13, we have $C_{E_n}(Z(M)) = MC_{E_n}(M)$. Since $C_{E_n}(M) \le E_n$ and $[C_{E_n}(M), M] = 1$, the resulting Clifford code is a subsystem code $C = A \otimes B$ with $\dim A$ and $\dim B$ as given in Theorem 9.

In addition, since $Z(E_n) \le M$ and $[C_{E_n}(M), M] = 1$, $\overline{Z(C_{E_n}(M))} \subseteq \overline{C_{E_n}(M)} \cap \overline{C_{E_n}(M)}^{\perp} \subseteq \overline{M}^{\perp} \cap \overline{M} \subseteq \overline{M}$. Thus, we have the relation $M \subseteq Z(C_{E_n}(M))M \subseteq M$, which means $Z(C_{E_n}(M))M = M$. Therefore, an error e in E_n is detectable if and only if $e \in E_n - (MC_{E_n}(M) - M)$. □

Now we are ready to construct subsystem codes from classical additive codes over a nice ring.

Theorem 15. *Let R be a nice ring with q elements. Let X be a classical additive subcode of R^{2n} such that $X \neq \{0\}$ and let Y denote its subcode $Y = X \cap X^{\perp}$. Let $x = |X|$ and $y = |Y|$. Then, there exists a subsystem code $C = A \otimes B$ such that*

1. $\dim A = q^n/(xy)^{1/2}$,

2. $\dim B = (x/y)^{1/2}$.

The minimum distance of subsystem A is given by $d = \operatorname{swt}((X + X^{\perp}) - X) = \operatorname{swt}(Y^{\perp} - X)$. Thus, the subsystem A can detect all errors in E_n of weight less than d, and can correct all errors in E_n of weight $\le \lfloor (d-1)/2 \rfloor$.

Proof. Let E_n be the nice error group given earlier, and let M be the full preimage of $\overline{M} = X$ in E_n under the canonical quotient map. Then, we can apply Theorem 14 to prove the theorem.

Since $\overline{Z(M)} = X \cap X^{\perp} = Y$, $|M : Z(M)| = |\overline{M} : \overline{Z(M)}| = x/y$. Thus, $\dim B = (x/y)^{1/2}$. From the fact that $Z(E_n) \leq M$ by definition, $|Z(E_n) \cap M|/|M| = 1/\overline{M} = 1/x$. Therefore, $\dim A = |E_n : Z(E_n)|^{1/2}/(xy)^{1/2} = q^n/(xy)^{1/2}$.

Since $\mathrm{wt}(e) = \mathrm{swt}(\overline{e})$ for an error $e \in E_n$, the minimum distance of subsystem A is $\mathrm{wt}(MC_{E_n}(M) - M) = \mathrm{swt}(\overline{MC_{E_n}(M) - M}) = \mathrm{swt}((X + X^{\perp}) - X)$. Equivalently, $\mathrm{wt}(C_{E_n}(Z(M)) - M) = \mathrm{swt}(\overline{C_{E_n}(Z(M)) - M}) = \mathrm{swt}(Y^{\perp} - X)$. □

Let R be a nice ring. Suppose $K = \dim A$, $L = \dim B$ and d is a minimum distance of subsystem A. Then, a subsystem code Q over a nice ring is called an $((n, K, L, d))_R$ subsystem code. We also write $[[n, k, l, d]]_R$ for an $((n, q^k, q^l, d))_R$ subsystem code, where q is the number of elements in R. By slight abuse of language, we will also refer to d as the minimum distance of the subsystem code Q, cf. [28].

Next, we will derive a special case of Theorem 15, which constructs a subsystem code over a nice ring with the help of two classical linear codes over a nice ring of the same length n. This generalizes a result from [28] by allowing finite rings instead of finite fields.

Let a, b, a', b' be in R^n. We define a form $\langle \cdot | \cdot \rangle_s : R^{2n} \times R^{2n} \to R$ by

$$\langle (a|b) | (a'|b') \rangle_s = b \cdot a' - b' \cdot a.$$

Suppose u and v are in R^{2n} such that $u = (a|b)$ and $v = (a'|b')$. Then, we define the orthogonality $u \perp_s v$ if and only if $\langle u|v \rangle_s = 0$.

Lemma 16. *Let C_1 and C_2 be two linear codes over a nice ring R such that $C_1 \leq R^n$ and $C_2 \leq R^n$. The product code $C_1 \times C_2 = \{(a|b) | a \in C_1, b \in C_2\}$ has length $2n$ and its dual is given by*

$$(C_1 \times C_2)^{\perp_s} = C_2^{\perp_s} \times C_1^{\perp_s}$$

Proof. If $(a|b) \in C_1 \times C_2$ and $(b'|a') \in C_2^{\perp_s} \times C_1^{\perp_s}$, then $\langle (a|b)|(b'|a') \rangle = b \cdot b' - a' \cdot a = 0$. Thus, $C_2^{\perp_s} \times C_1^{\perp_s} \subseteq (C_1 \times C_2)^{\perp_s}$. Since $|(C_1 \times C_2)^{\perp_s}| = |R|^{2n}/|C_1 \times C_2| = |R|^{2n}/|C_1||C_2| = |R|^n/|C_2| \cdot |R|^n/|C_1| = |C_2^{\perp_s} \times C_1^{\perp_s}|$, we can conclude that $(C_1 \times C_2)^{\perp_s} = C_2^{\perp_s} \times C_1^{\perp_s}$. □

Corollary 17. *Let R be a nice ring with q elements. Let C_i be $[n, k_i]$ linear codes in R^n for $i \in \{1, 2\}$. Then, there exists an $((n, K, L, d))_R$ subsystem code with*

1. $K = q^{n-(k_1+k_2)/2}/|D|^{1/2}$,
2. $L = q^{(k_1+k_2)/2}/|D|^{1/2}$,
3. $d = \min\{\mathrm{wt}((C_1^{\perp_s} \cap C_2)^{\perp_s} \backslash C_1), \mathrm{wt}((C_2^{\perp_s} \cap C_1)^{\perp_s} \backslash C_2)\}$,

where $|D| = |C_1 \cap C_2^{\perp_s}||C_2 \cap C_1^{\perp_s}|$.

Proof. Let $C = C_1 \times C_2$, then by Lemma 16, $C^{\perp_s} = C_2^{\perp_s} \times C_1^{\perp_s}$. Using C and $C^{\perp_s}$, we get $D = C \cap C^{\perp_s} = (C_1 \cap C_2^{\perp_s}) \times (C_2 \cap C_1^{\perp_s})$. Since $|C| = |C_1||C_2| = q^{k_1+k_2}$, we can obtain that $K = q^{n-(k_1+k_2)/2}/|D|^{1/2}$ and $L = q^{(k_1+k_2)/2}/|D|^{1/2}$ by Theorem 15. The distance of the code is provided in Theorem 15 by

$$\begin{aligned} d =& \,\mathrm{swt}(D^{\perp_s} \backslash C) \\ =& \,\mathrm{swt}((C_2 \cap C_1^{\perp_s})^{\perp_s} \times (C_1 \cap C_2^{\perp_s})^{\perp_s} \backslash (C_1 \times C_2)), \end{aligned}$$

which can be simplified to

$$d = \min\{\mathrm{wt}((C_2 \cap C_1^{\perp_s})^{\perp_s} \backslash C_1), \mathrm{wt}((C_1 \cap C_2^{\perp_s})^{\perp_s} \backslash C_2)\}.$$

Therefore, we can have an $((n, K, L, d))_R$ subsystem code from C_1 and C_2. □

In particular, an subsystem code $((n, K, L, d))_R$ with simplified code parameters can be obtained by setting $C_1 = C_2$, where $K = q^{n-k}/|D|$, $L = q^k/|D|$, $|D| = |C_1 \cap C_1^{\perp_s}|$ and $d = wt((C_1 \cap C_1^{\perp_s})^{\perp_s} \backslash C_1)$.

3.3 Subsystem Codes over Rings and Fields

Now, we will discuss a finite chain ring to show how to derive the subsystem code over a field from the subsystem code over a ring. A finite chain ring is a well-known ring structure for a classical code [31], [32]. Since ideals of a finite chain ring construct a form of a chain, it has a unique maximal ideal. A finite chain ring is Frobenius ring, which implies that a finite chain ring is a nice distributively generated nearring [16]. In order to discuss the relation between the subsystem code over a ring and a finite field, we restrict our focus into a finite chain ring.

Let R be a finite chain ring with the Jacobson radical $J(R)$ and the nilpotency index ν. Then, $J(R)$ is the maximal ideal of R since a finite chain ring is a local ring and it has a unique maximal ideal. Thus, the quotient ring $R/J(R)$ becomes a field. Let F be such a residue field. Suppose that a field $R/J(R)$ has q elements. Then, we have $|R| = q^\nu$ and $|J(R)| = q^{\nu-1}$ [31].

Let $c \in R^n$. We denote by $\overline{c}$ the image of c under the canonical projection from R^n to F^n. Let $C \subseteq R^n$. Then, we denote $\overline{C} = \{\overline{c} | c \in C\}$.

We define the submodule quotient $(C : r)$ as a set $\{e \in R^{2n} | re \in C\}$ for any code $C \leq R^{2n}$ and any $r \in R$. The submodule quotient and its image under the projection have following chain conditions [31]

$$C = (C : \gamma^0) \subseteq \cdots \subseteq (C : \gamma^i) \subseteq \cdots \subseteq (C : \gamma^{\nu-1}),$$

and its projection to F,

$$\overline{C} = \overline{(C : \gamma^0)} \subseteq \cdots \subseteq \overline{(C : \gamma^i)} \subseteq \cdots \subseteq \overline{(C : \gamma^{\nu-1})}.$$

In order to know the relation between the subsystem code over a ring and that over a field, we need to first show the relation of the symplectic weights of a code over a ring and that over its residue field.

Lemma 18. *Let R be a finite chain ring, γ a generator of the Jacobson radical $J(R)$ and ν the nilpotency index. Let X be a classical additive code of R^{2n} and Y its subcode $Y = X \cap X^{\perp_s}$. Then,*

$$\mathrm{swt}(Y^{\perp_s} - X) \leq \mathrm{swt}(\overline{Y^{\perp_s} - (X : \alpha)}),$$

where $\alpha = \gamma^{\nu-1}$.

Proof. Let us consider $x \in Y^{\perp_s} - (X : \alpha)$. Then, we know $\alpha x \in Y^{\perp_s} - X$. This means that $\alpha(Y^{\perp_s} - (X : \alpha)) \subseteq Y^{\perp_s} - X$. Therefore, $\mathrm{swt}(Y^{\perp_s} - X) \leq \mathrm{swt}(\alpha(Y^{\perp_s} - (X : \alpha)))$.

We define the map $\varphi : \alpha R^{2n} \to F^{2n}$ given by $\varphi(\alpha x) = \overline{x}$. Since φ is an isomorphism and preserves the weight [32], we can say that this map also preserves the symplectic weight. Thus, we have the relation that $\mathrm{swt}(\alpha(Y^{\perp_s} - (X : \alpha))) = \mathrm{swt}(\overline{Y^{\perp s} - (X : \alpha)})$. Then,

$$\begin{aligned} \mathrm{swt}(Y^{\perp_s} - X) \leq & \,\mathrm{swt}(\alpha(Y^{\perp_s} - (X : \alpha))) \\ = & \,\mathrm{swt}(\overline{Y^{\perp_s} - (X : \alpha)}) \end{aligned}$$

Therefore, we can conclude that $\mathrm{swt}(Y^{\perp_s} - X) \leq \mathrm{swt}(\overline{Y^{\perp_s} - (X : \alpha)})$. □

For the simpler analysis of the relation between the subsystem code over a ring and a field, we will focus on the free code. The free code over a ring has the following properties [31].

Let C be a free code over R. Then, the dual code $C^{\perp_s}$ is also a free code. The number of rows in a generator matrix in standard form $k(C)$ is $k_0(C)$, which is the number of rows not divisible by γ^i for $1 \leq i \leq \nu - 1$. In addition, $\overline{C} = \overline{(C:\gamma)} = \cdots = \overline{(C:\gamma^{\nu-1})}$.

Theorem 19. *If an $((n, K, L, d))_R$ free subsystem code exists over a finite chain ring with the Jacobson radical $J(R)$ and the nilpotency index ν, then there exists an $((n, K^{1/\nu}, L^{1/\nu}, \geq d))_q$ subsystem code over a field $R/J(R)$ with q elements.*

Proof. By Theorem 15, there are the classical additive code $X \subseteq R^{2n}$ and its subcode $Y = X \cap X^{\perp_s}$, which are associated with an $((n, K, L, d))_R$ subsystem code. Since an $((n, K, L, d))_R$ subsystem code is a free code, X and Y are free codes. Suppose that generator matrices of X and Y have the numbers of rows k_0 and k_0', respectively. Then, $|X| = q^{\nu k_0}$ and $|Y| = q^{\nu k_0'}$ [31]. Therefore, $K = |R|^n / q^{\nu(k_0+k_0')/2} = q^{\nu(n-(k_0+k_0')/2)}$ and $L = q^{\nu(k_0-k_0')/2}$.

Now we consider $\overline{X}$ and $\overline{Y}$, which are the subset of F^{2n}. Since $dim(\overline{X}) = k_0$ and $dim(\overline{Y}) = k_0'$ [31], we have $|\overline{X}| = q^{k_0}$ and $|\overline{Y}| = q^{k_0'}$. By [29, Theorem 5], we can have an $((n, K', L', d'))_q$ subsystem code using $\overline{X}$ and $\overline{Y}$, where $K' = q^{n-(k_0+k_0')/2}$ and $L' = q^{(k_0-k_0')/2}$. Since $K' = K^{1/\nu}$ and $L' = L^{1/\nu}$, we conclude that there exists an $((n, K^{1/\nu}, L^{1/\nu}, d'))_q$ subsystem code over a finite field with q elements.

Since a free code has a property that $\overline{X} = \overline{(X:\alpha)}$ [31], the minimum distance $d' = \mathrm{swt}(\overline{Y^{\perp_s} - X}) = \mathrm{swt}(\overline{Y^{\perp_s} - (X:\alpha)}) \geq \mathrm{swt}(Y^{\perp_s} - X) = d$, where the inequality is from Lemma 18. □

Changing a notation to a simpler one, the derived subsystem code over a finite field has same parameters as the free subsystem code over a finite chain ring except for their distances.

Corollary 20. *If an* $[[n,k,l,d]]_R$ *free subsystem code exists over a finite chain ring, then there exists an* $[[n,k,l,\geq d]]_q$ *subsystem code over a finite field.*

Proof. From the proof of Theorem 19, an $((n,K,L,d))_R$ free subsystem code can be called an $[[n,n-(k_0+k_0')/2,(k_0-k_0')/2,d]]_R$ free subsystem code since $|R| = q^\nu$. Using the fact that $|F| = q$, an $((n,K^{1/\nu},L^{1/\nu},\geq d))_q$ subsystem code can be called an $[[n,n-(k_0+k_0')/2,(k_0-k_0')/2,\geq d]]_q$ subsystem code. By letting $k = n-(k_0+k_0')/2$ and $l = (k_0-k_0')/2$, we can conclude that an $[[n,k,l,\geq d]]_q$ code can be derived from an $[[n,k,l,d]]_R$ code. □

4. ENTANGLEMENT-ASSISTED QUANTUM ERROR CORRECTING CODES OVER NICE RINGS

In this chapter, the generalization of the other quantum error correcting scheme, entanglement-assisted quantum error correcting code over nice rings, which was published in [19][1], will be given.

Entanglement-assisted quantum error correcting codes (EAQECCs), which are quantum codes with the pre-shared entanglement between the sender and the receiver, have been introduced [33, 34]. Unlike the stabilizer codes, these codes do not have a restriction that classical codes should be dual-containing codes for constructing the quantum codes. Thus, using this construction with a help of the entanglement, any classical codes can be used in order to derive the corresponding quantum codes. The constructions of the binary and quaternary EAQECCs were well-established [33, 34] and the formalism of the generalized EAQECCs combining EAQECCs and subsystem quantum error correcting codes was investigated [35].

What we now show is that the entanglement-assisted quantum error correcting codes can be generalized over nice rings, so that these codes can also be not restricted to power prime dimensions [19].

We will use the two types of forms to connect the commutativity with the orthogonality. From [17], we have a form to use over R-module instead of the symplectic inner product over a finite field. For each additive character χ, a unique function ψ

[1] 2014 IEEE. Reprinted, with permission, from Sangjun Lee and Andreas Klappenecker, Entanglement-assisted quantum error correcting codes over nice rings, 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), October 2014

in $Hom(R, \mathbb{Q}/\mathbb{Z})$ exists such that

$$\chi(x) = \exp(2\pi i \psi(x)).$$

Using the function ψ, we define a form $\langle \cdot | \cdot \rangle_\chi : R^{2n} \times R^{2n} \to \mathbb{Q}/\mathbb{Z}$ such that

$$\langle (a|b) | (a'|b') \rangle_\chi = \psi(b \cdot a' - b' \cdot a)$$

for all $(a|b), (a'|b') \in R^{2n}$.

Let $u = (a|b)$ and $v = (a'|b')$ in R^{2n}. We say that $u \perp v$ if and only if $\langle u|v \rangle_\chi = 0$. This implies that $X(a)Z(b)$ commutes with $X(a')Z(b')$ if and only if $u \perp v$ [17].

There is an another inner product [18], which may be preferred to use in this setting, which is defined by

$$\langle (a|b) | (a'|b') \rangle = \chi(b \cdot a' - b' \cdot a)$$

for all $(a|b), (a'|b') \in R^{2n}$.

Since this form uses the additive character χ directly, $X(a)Z(b)$ and $X(a')Z(b')$ commute if and only if $\chi(b \cdot a' - b' \cdot a) = 1$ [17]. Thus, this inner product has an orthogonal property such that $u \perp v$ if and only if $\langle u|v \rangle = 1$. It is easily seen that $\langle u|v \rangle = 1$ is equivalent to $\langle u|v \rangle_\chi = 0$ since ψ is in $Hom(R, \mathbb{Q}/\mathbb{Z})$. Both forms share the same orthogonal condition.

We would like to recall the definitions of the weight and the symplectic weight. Given an error $e = \omega^c X(a)Z(b)$ in an error group for $a, b \in R^n$ and $c \in \mathbf{Z}$, the weight $\mathrm{wt}(e)$ is defined as the number of its tensor components which are not scalar multiples of the identity. The symplectic weight of $(a|b) \in R^{2n}$ is defined by

$$\mathrm{swt}(a|b) = |\{i|a_i \neq 0 \text{ or } b_i \neq 0, 1 \leq i \leq n\}|.$$

It is easily seen that $\mathrm{wt}(\omega^c X(a)Z(b)) = \mathrm{swt}(a|b)$.

4.1 Symplectic Abelian Groups

In [36], an antisymmetric bicharacter on finite abelian groups is introduced. The inner product $\langle\cdot|\cdot\rangle$ is indeed one of such antisymmetric bicharacters since $\langle(a|b)|(a|b)\rangle$ $= \chi(0) = 1$. A finite abelian group with an antisymmetric bicharacter is called a finite symplectic abelian group. It is known that a finite symplectic abelian group can have a decomposition as an orthogonal direct sum of Sylow p-subgroups and a primary symplectic abelian group can be splitted as an orthogonal direct sum of its homogeneous components [37]. A primary abelian group is called homogeneous if all of its invariants are equal. It is known that any finite abelian group is a direct product of homogeneous subgroups such that no two distinct factors have a common invariant. Such subgroups are called the homogeneous components of a group. Each homogeneous component can indeed be splitted into an orthogonal direct sum of hyperbolic subgroups [37, 36]. A subgroup of a symplectic abelian group is called hyperbolic if the subgroup is nonsingular and $H \cong \mathbb{Z}_{p^n} \times \mathbb{Z}_{p^n}$ for some prime p and some integer $n \geq 1$.

R^{2n} with the inner product $\langle\cdot|\cdot\rangle$ is considered as a symplectic abelian group since $\langle\cdot|\cdot\rangle$ is the antisymmetric bicharacter [36], which defines a symmetric relation of orthogonality, and R^{2n} is a R-module. Then, we can have

Theorem 21. *If R^{2n} is a nondegenerate abelian group with $\langle\cdot|\cdot\rangle$ and $R^{2n} = H_1 \oplus \cdots \oplus H_m$ is any decomposition as the direct sum of homogeneous components such* *that each homogeneous component has an even rank of $r_i = 2s_i$ for $1 \leq i \leq m$, then R^{2n} admits a decomposition into the orthogonal direct sum of $\sum_i s_i$ hyperbolic subgroups.*

Proof. Each homogeneous component H_i is nondegenerate since R^{2n} is nondegenerate, and a nondegenerate, homogeneous, symplectic, abelian group of even rank $2s$ is splitted into the orthogonal direct sum of s hyperbolic subgroups [36]. □

The following lemmas provide proofs that R^{2n} is decomposed as an orthogonal direct sum of hyperbolic subgroups like the symplectic linear spaces.

Lemma 22. *Let R^{2n} be a nondegenerate abelian group with $\langle\cdot|\cdot\rangle$. Then, there is homogeneous components of R^{2n} such that an orthogonal direct sum of those homogeneous components is R^{2n}.*

Proof. Any finite abelian group can be decomposed as an orthogonal direct sum of its Sylow p-subgroups [37]. A primary symplectic abelian group can be decomposed as an orthogonal direct sum of homogeneous components [37, 36]. Therefore, it is clear that R^{2n} can be decomposed as an orthogonal direct sum of homogeneous components. □

Lemma 23. *Let R^{2n} be a nondegenerate abelian group with $\langle\cdot|\cdot\rangle$. A homogeneous component of R^{2n} can be decomposed as an orthogonal direct sum of hyperbolic subgroups.*

Proof. Since a homogeneous component of R^{2n} is a homogeneous nondegenerate abelian p-group, it is decomposed as an orthogonal direct sum of two isomorphic isotropic cyclic subgroups, which are called hyperbolic subgroups [37, 36]. □

Using two lemmas above, we now have the following result.

Theorem 24. *Let R^{2n} be a nondegenerate abelian group with $\langle\cdot|\cdot\rangle$. Then, R^{2n} can be decomposed as an orthogonal direct sum of hyperbolic subgroups.*

Proof. From Lemmas 22 and 23, R^{2n} can be decomposed as an orthogonal direct sum of homogeneous components, and each homogeneous component can be decomposed as an orthogonal direct sum of hyperbolic subgroups. □

4.2 Entanglement-Assisted Quantum Codes over Nice Rings

It was known that $(\mathbb{Z}_2)^{2n}$ is a symplectic subspace of itself [33, 34]. This implies that there exists a symplectic basis of $(\mathbb{Z}_2)^{2n}$ consisting of n hyperbolic pairs. From previous section, we now know that R^{2n} also can be decomposed as an orthogonal direct sum of hyperbolic subgroups like a vector space. In addition, the form $\langle\cdot|\cdot\rangle = 1$ is equivalent to $\langle\cdot|\cdot\rangle_\chi = 0$, so they both have the same orthogonal condition.

We will show how to find a symplectic basis and an isotropic basis generating a free submodule of R^{2n}. A subgroup is called symplectic if it is generated by a set of anti-commuting generator pairs, a symplectic basis, and a subgroup is isotropic if it is generated by a set of commuting generators, an isotropic basis [38]. The form $\langle\cdot|\cdot\rangle_\chi$, not $\langle\cdot|\cdot\rangle$, will be used as an inner product in order to follow the approach similarly, which was provided for a vector space over a finite field [33, 34].

We denote an element in $\mathbb{Q}/\mathbb{Z}$ by $[\frac{a}{b}]$, such that (a, b) is co-prime and $0 \leq \frac{a}{b} < 1$. Then, we choose $\frac{a}{b}$ as a representative of $[\frac{a}{b}]$.

Theorem 25. *Let R be a commutative nice ring. Let M be a free submodule of R^{2n} with a rank m. Then, there exists a symplectic basis of R^{2n} with hyperbolic pairs $(\mathbf{u_i}, \mathbf{v_i})$ for $1 \leq i \leq n$ such that $\langle \mathbf{u_i}|\mathbf{u_j}\rangle_\chi = 0$ for all i, j, $\langle \mathbf{v_i}|\mathbf{v_j}\rangle_\chi = 0$ for all i, j, $\langle \mathbf{u_i}|\mathbf{v_j}\rangle_\chi = 0$ for all $i \neq j$ and $\langle \mathbf{u_i}|\mathbf{v_i}\rangle_\chi \neq 0$ for all i. A basis $\{\mathbf{u_1}, \cdots, \mathbf{u_{c+s}}, \mathbf{v_1}, \cdots, \mathbf{v_c}\}$ generates M for some $c, s \geq 0$ with $2c + s = m$.*

Proof. Since M is a free submodule of R^{2n} with a rank m, we can pick a set of basis $\{\mathbf{w_1}, \cdots, \mathbf{w_{2n}}\}$ for R^{2n} such that $\{\mathbf{w_1}, \cdots, \mathbf{w_m}\}$ is a basis for M.

The following algorithm runs n rounds, and one pair $(\mathbf{u_i}, \mathbf{v_i})$ is calculated in each round.

Initially, we have $i = 1$, $m' = m$ and $U = V = \emptyset$. In ith round,

1. For $\mathbf{w_1}, \cdots, \mathbf{w_{2(n-i+1)}}$ and $\mathbf{u_1}, \cdots, \mathbf{u_{i-1}}, \mathbf{v_1}, \cdots, \mathbf{v_{i-1}}$, the followings are satisfied.

   (a) $\{\mathbf{w_1}, \cdots, \mathbf{w_{2(n-i+1)}}, \mathbf{u_1}, \cdots, \mathbf{u_{i-1}}, \mathbf{v_1}, \cdots, \mathbf{v_{i-1}}\}$ is a basis for R^{2n}.

   (b) $\langle \mathbf{u}_j | \mathbf{w}_k \rangle_\chi = 0$ and $\langle \mathbf{v}_j | \mathbf{w}_k \rangle_\chi = 0$ for $1 \leq j \leq i-1$ and $1 \leq k \leq 2(n-i+1)$.

   (c) $M = \text{span}\{\mathbf{u_j} : j \in U\} \oplus \text{span}\{\mathbf{v_l} : l \in V\} \oplus \text{span}\{\mathbf{w_k} : 1 \leq k \leq m'\}$.

2. Choose $\mathbf{u_i} = \mathbf{w_1}$. If $m' \geq 1$, then add i to U. Let $t \geq 2$ be the smallest index such that $\langle \mathbf{w_1} | \mathbf{w_t} \rangle_\chi \neq 0$. Set $\mathbf{v_i} = \mathbf{w_t}$.

3. If $t \leq m'$:

   The hyperbolic pair $(\mathbf{u_i}, \mathbf{v_i})$ are in a set of generators generating M. Add i to V, and swap $\mathbf{w_t}$ with $\mathbf{w_2}$.

   For $k = 3, \cdots, 2(n-i+1)$, a new basis that is orthogonal to the hyperbolic pair $(\mathbf{u}_i, \mathbf{v}_i)$ is computed.

   Let $\langle \mathbf{v}_i | \mathbf{u}_i \rangle_\chi = [\frac{x_i}{y_i}]$, $\langle \mathbf{v}_i | \mathbf{w}_k \rangle_\chi = [\frac{a_k}{b_k}]$, and $\langle \mathbf{u}_i | \mathbf{w}_k \rangle_\chi = [\frac{c_k}{d_k}]$, where $\frac{x_i}{y_i}$, $\frac{a_k}{b_k}$, and $\frac{c_k}{d_k}$ are representatives of $[\frac{x_i}{y_i}], [\frac{a_k}{b_k}]$, and $[\frac{c_k}{d_k}]$, respectively. Let $\frac{a_k}{b_k} \cdot \frac{y_i}{x_i} = \frac{g_{k,i}}{h_{k,i}}$ and $\frac{c_k}{d_k} \cdot \frac{y_i}{x_i} = \frac{r_{k,i}}{s_{k,i}}$, where $(g_{k,i}, h_{k,i})$ and $(r_{k,i}, s_{k,i})$ are co-prime, respectively. In order to have all coefficients in the following equation as integers, the least common multiple of two denominators is calculated. Let $e_{k,i} = \text{lcm}(h_{k,i}, s_{k,i})$. Then,

$$\mathbf{w}'_{k-2} = e_{k,i}\mathbf{w}_k - e_{k,i}\frac{g_{k,i}}{h_{k,i}}\mathbf{u}_i - e_{k,i}\frac{r_{k,i}}{s_{k,i}}\mathbf{v}_i,$$

so that

$$\langle \mathbf{w}'_{k-2} | \mathbf{u}_i \rangle_\chi = \langle \mathbf{w}'_{k-2} | \mathbf{v}_i \rangle_\chi = 0.$$

Then, set $m' = m - 2$ since we found two generators of M.

If $t > m'$:

One of the hyperbolic pair $\mathbf{u_i}$ is in a set of generators generating M, but another one $\mathbf{v_i}$ is out of a set of generators of M. Swap $\mathbf{w_t}$ with $\mathbf{w_{2(n-i+1)}}$.

For $k = 2, \cdots, 2(n-i)+1$, a new basis that is orthogonal to the hyperbolic pair $(\mathbf{u}_i, \mathbf{v}_i)$ is computed.

Let $\langle \mathbf{v}_i | \mathbf{u}_i \rangle_\chi = [\frac{x_i}{y_i}]$, $\langle \mathbf{v}_i | \mathbf{w}_k \rangle_\chi = [\frac{a_k}{b_k}]$, and $\langle \mathbf{u}_i | \mathbf{w}_k \rangle_\chi = [\frac{c_k}{d_k}]$, where $\frac{x_i}{y_i}$, $\frac{a_k}{b_k}$, and $\frac{c_k}{d_k}$ are representatives of $[\frac{x_i}{y_i}], [\frac{a_k}{b_k}]$, and $[\frac{c_k}{d_k}]$, respectively. Let $\frac{a_k}{b_k} \cdot \frac{y_i}{x_i} = \frac{g_{k,i}}{h_{k,i}}$ and $\frac{c_k}{d_k} \cdot \frac{y_i}{x_i} = \frac{r_{k,i}}{s_{k,i}}$, where $(g_{k,i}, h_{k,i})$ and $(r_{k,i}, s_{k,i})$ are co-prime, respectively. To have all coefficients as integers, the least common multiple is calculated. Let $e_{k,i} = \mathrm{lcm}(h_{k,i}, s_{k,i})$. Then,

$$\mathbf{w}'_{\mathbf{k-1}} = e_{k,i}\mathbf{w}_k - e_{k,i}\frac{g_{k,i}}{h_{k,i}}\mathbf{u}_i - e_{k,i}\frac{r_{k,i}}{s_{k,i}}\mathbf{v}_i,$$

so that

$$\langle \mathbf{w}'_{k-1} | \mathbf{u}_i \rangle_\chi = \langle \mathbf{w}'_{k-1} | \mathbf{v}_i \rangle_\chi = 0.$$

Then, set $m' = m - 1$ if $m' \geq 1$, since we only found one generator of M.

4. $\mathbf{w}_k = \mathbf{w_k}'$ for $1 \leq k \leq 2(n-i)$.

Reordering $\mathbf{u}'_j s$ and $\mathbf{v}'_j s$ may be required to put the hyperbolic pairs in a set of generators of M first and the remaining $\mathbf{u}_j's$ following the hyperbolic pairs.

□

Let $u_i = (a_1, \cdots, a_n | b_1, \cdots, b_n) \in R^{2n}$, where $a_i, b_i \in R$ for $1 \leq i \leq n$. The corresponding quantum error is defined as $X(a_1)Z(b_1) \otimes \cdots \otimes X(a_n)Z(b_n)$.

Lemma 26. *Let G_n be a nice error basis indexed by a nice ring R with q elements. Let $\mathcal{S}$ be a subgroup in G_n with q^m elements up to scalar. Then, there exists a set of m independent generators $\{Z_1, Z_2, \cdots, Z_c, \cdots, Z_{c+s}, X_1, \cdots, X_c\}$ for $\mathcal{S}$ such that $[Z_i, Z_j] = 0$ for all i, j, $[X_i, X_j] = 0$ for all i, j, $[Z_i, X_j] = 0$ for all $i \neq j$ and $[Z_i, X_i] \neq 0$ for all i.*

Proof. From Theorem 25, it is guaranteed that we can have such a set of generators.

□

Suppose that we have a subgroup S in G_n generated by the noncommuting set of generators. From Lemma 26, we can obtain a new set of generators consisting of c hyperbolic pairs and s commuting generators, where c is the number of entangled-qudits and s is the number of ancilla qudits, which are ebits and ancilla bits for binary quantum codes [33]. We label two generators in each hyperbolic pair as Z_i and X_i for $1 \leq i \leq c$, and commuting generators as Z_i for $c+1 \leq i \leq c+s$. As [33] has done, we can find a new group generated by commuting generators by extending the generators. The only different thing from [33] is that this is not just a binary case, but the generalized one over a nice ring.

For a hyperbolic pair $(a|b)$ and $(a'|b')$ in R^{2n}, let $Z_i = X(a)Z(b)$ and $X_i = X(a')Z(b')$. Then, we know $\chi(b \cdot a' - b' \cdot a) \neq 1$ since Z_i and X_i do not commute. Let $Z(s)$ and $X(t)$ be operators to be appended to Z_i and X_i, respectively, to obtain the commuting pair for some $s, t \in R$. The extended pairs $Z'_i = X(a|0)Z(b|s)$ and

$X_i' = X(a'|t)Z(b'|0)$ commute if and only if $\chi((b|s)\cdot(a'|t)-(b'|0)\cdot(a|0))=1$. This implies that $\chi(b\cdot a'-b'\cdot a)\chi(st)=1$, thus $st=b'\cdot a-b\cdot a'$. We can easily pick s and t in R satisfying $st=b'\cdot a-b\cdot a'$. The trivial case may be to pick $s=1$ or $t=1$. By appending c entangled-qudits, we now have the extended set of commuting generators. It is assumed that the original qudits are possessed by Alice, and the appended qudits are possessed by Bob and are error-free.

With the generators with appended qudits, EAQECC over a nice ring can be constructed. Initially, a state $|\Phi\rangle^{AB}|0\rangle^{\otimes s}|\psi\rangle^{\otimes k}$ is prepared, where $|\Phi\rangle^{AB}=\frac{1}{\sqrt{|R|}}\sum_{x\in R}|x\rangle|x\rangle$ is a generalized Bell state, which is a maximally entangled state shared between Alice and Bob, $|\psi\rangle^{\otimes k}$ contains the quantum information to transmit, and k is a number of logical qudits. After encoding process on qudits possessed by Alice, the code space is now stabilized by the generators. A half of the entangled pair is possessed by Bob, and that is assumed to be error-free, so that Bob can measure the extended generators on the transmitted qudits along with his entangled pair.

Let n be a number of physical qudits, k a number of logical qudits, d a minimum distance, and c a number of entangled-qudits or a number of hyperbolic pairs. Then, an EAQECC over a nice ring is called an $[[n,k,d;c]]_R$ EAQECC.

Proposition 27. *Let R be a nice ring. If a classical $[n,k,d]_R$ code exists, then an $[[n,2k-n+c,d;c]]_R$ EAQECC exists, where c is the number of hyperbolic pairs in the symplectic basis.*

Proof. The classical $[n,k,d]_R$ code has a $(n-k)\times n$ parity-check matrix H. Using H, we can construct a $2(n-k)\times 2n$ parity-check matrix $\tilde{H}$ for CSS construction. From Lemma 25 and Theorem 26, we know it is possible to find generators $\{Z_1,Z_2,\cdots,Z_c,\cdots,Z_{c+s},X_1,\cdots,X_c\}$, where $2c+s=2(n-k)$. By appending appropriate qudits for extended commuting generators, we finally set the entanglement-assisted quantum error correcting code. For any nonzero $(a|b) \in R^{2n}$ such that $\mathrm{swt}(a|b) < d$, the syndrome of $(a|b)$ is nonzero by the definition of the minimum distance. It means that any error $e = \chi(c)X(a)Z(b)$ such that $\mathrm{wt}(e) < d$ can be detected. Therefore, the constructed EAQECC is the $[[n, 2k - n + c, d; c]]_R$ code, where c is the number of hyperbolic pairs in the symplectic basis. □

4.3 Examples

One of good examples for constructing EAQECCs is quantum LDPC code. The classical LDPC codes are well-known as capacity-approaching codes, so there has been many investigations on them. But because they are not generally dual-containing codes, which are usually used to construct quantum codes, there has been many attempts to find dual-containing classical LDPC codes for CSS construction [39]. It has been relatively easy to find dual-containing LDPC codes with a short length, but difficult to find long LDPC codes. The entanglement-assisted quantum code construction, however, has no dual-containing constraints, so constructing quantum LDPC codes directly from classical LDPC codes can be possible [40].

Here, we provide some examples to show that it is possible to employ any LDPC codes directly to construct quantum codes with an entanglement-assistance.

4.3.1 [[6,1;1]] Quantum Binary LDPC Code

The classical irregular [6,3] binary LDPC code has a following parity-check matrix.

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

For a CSS construction, we have the parity-check matrix from above.

$$\tilde{H} = \left(\begin{array}{c|c} H & 0 \\ \hline 0 & H \end{array} \right).$$

From $\tilde{H}$, the stabilizer group has the following noncommuting generators.

$$\begin{array}{rcllllll} M_1 & = & Z & Z & Z & Z & I & I \\ M_2 & = & I & I & Z & Z & I & Z \\ M_3 & = & Z & I & I & Z & Z & I \\ M_4 & = & X & X & X & X & I & I \\ M_5 & = & I & I & X & X & I & X \\ M_6 & = & X & I & I & X & X & I \end{array}$$

Using Lemma 25, we obtain the symplectic basis and the isotropic basis to generate the stabilizer group. The function $\psi \in Hom(R, \mathbb{Q}/\mathbb{Z})$ is $\psi(x) = x/2$. The additive character χ is $\chi(x) = exp(2\pi i \psi(x)) = exp(\pi i x)$.

$$\begin{array}{rcllllll} Z_1 & = & I & I & Z & Z & I & Z \\ X_1 & = & I & I & X & X & I & X \\ Z_2 & = & Z & Z & Z & Z & I & I \\ Z_3 & = & X & X & X & X & I & I \\ Z_4 & = & I & Z & I & Z & Z & Z \\ Z_5 & = & I & X & I & X & X & X \end{array}$$

It is shown that a pair (Z_1, X_1) is only a hyperbolic pair in a generator set, and they commute with any generators in an isotropic space. The number of hyperbolic pairs is just one, and the number of generators in an isotropic space is four, which imply that c=1 and s=4, so the number of logical qubits is k=1. Thus, this quantum LDPC code is [[6,1;1]] code. The following is the generators with an appended qubit belonging to a receiver Bob.

$$\begin{array}{ccccccc|c}
\overline{Z_1} & = & I & I & Z & Z & I & Z & Z \\
\overline{X_1} & = & I & I & X & X & I & X & X \\
\overline{Z_2} & = & Z & Z & Z & Z & I & I & I \\
\overline{Z_3} & = & X & X & X & X & I & I & I \\
\overline{Z_4} & = & I & Z & I & Z & Z & Z & I \\
\overline{Z_5} & = & I & X & I & X & X & X & I
\end{array}$$

By appending Z to Z_1, X to X_1, and I to the remainder, the generators with the appended become commuting each other.

4.3.2 [[8,3;3]] Quantum LDPC Code over $\mathbf{F}_{16}$

The finite field $\mathbf{F}_{16}$ with 16 elements can be understood as the residue class ring $\mathbf{F}_{16} = \mathbf{F}_2[x]/\langle x^4 + x + 1\rangle$, so a residue class can be represented by a polynomial $a + bx + cx^2 + dx^3$ over $\mathbf{F}_2$. Since we do not have much space, we choose to represent the elements of $\mathbf{F}_{16}$ by integers from 0 to 15 as follows:

$$\begin{gathered}
0 = (0,0,0,0),\ 1 = (1,0,0,0),\ 2 = (0,1,0,0),\ 3 = (0,0,1,0), \\
4 = (0,0,0,1),\ 5 = (1,1,0,0),\ 6 = (0,1,1,0),\ 7 = (0,0,1,1), \\
8 = (1,1,0,1),\ 9 = (1,0,1,0), 10 = (0,1,0,1), 11 = (1,1,1,0), \\
12 = (0,1,1,1), 13 = (1,1,1,1), 14 = (1,0,1,1), 15 = (1,0,0,1),
\end{gathered}$$

where (a,b,c,d) consists of the coefficients of the polynomial. The addition and multiplication tables are provided below in Table 4.1.

The parity-check matrix of the classical (2,4)-regular [8,4] LDPC code over $\mathbf{F}_{16}$ is

Table 4.1: The addition and multiplication tables over $\mathbf{F}_{16}$

+	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	5	9	15	2	11	14	10	3	8	6	13	12	7	4
2	2	5	0	6	10	1	3	12	15	11	4	9	7	14	13	8
3	3	9	6	0	7	11	2	4	13	1	12	5	10	8	15	14
4	4	15	10	7	0	8	12	3	5	14	2	13	6	11	9	1
5	5	2	1	11	8	0	9	13	4	6	15	3	14	7	12	10
6	6	11	3	2	12	9	0	10	14	5	7	1	4	15	8	13
7	7	14	12	4	3	13	10	0	11	15	6	8	2	5	1	9
8	8	10	15	13	5	4	14	11	0	12	1	7	9	3	6	2
9	9	3	11	1	14	6	5	15	12	0	13	2	8	10	4	7
10	10	8	4	12	2	15	7	6	1	13	0	14	3	9	11	5
11	11	6	9	5	13	3	1	8	7	2	14	0	15	4	10	12
12	12	13	7	10	6	14	4	2	9	8	3	15	0	1	5	11
13	13	12	14	8	11	7	15	5	3	10	9	4	1	0	2	6
14	14	7	13	15	9	12	8	1	6	4	11	10	5	2	0	3
15	15	4	8	14	1	10	13	9	2	7	5	12	11	6	3	0

×	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1
3	0	3	4	5	6	7	8	9	10	11	12	13	14	15	1	2
4	0	4	5	6	7	8	9	10	11	12	13	14	15	1	2	3
5	0	5	6	7	8	9	10	11	12	13	14	15	1	2	3	4
6	0	6	7	8	9	10	11	12	13	14	15	1	2	3	4	5
7	0	7	8	9	10	11	12	13	14	15	1	2	3	4	5	6
8	0	8	9	10	11	12	13	14	15	1	2	3	4	5	6	7
9	0	9	10	11	12	13	14	15	1	2	3	4	5	6	7	8
10	0	10	11	12	13	14	15	1	2	3	4	5	6	7	8	9
11	0	11	12	13	14	15	1	2	3	4	5	6	7	8	9	10
12	0	12	13	14	15	1	2	3	4	5	6	7	8	9	10	11
13	0	13	14	15	1	2	3	4	5	6	7	8	9	10	11	12
14	0	14	15	1	2	3	4	5	6	7	8	9	10	11	12	13
15	0	15	1	2	3	4	5	6	7	8	9	10	11	12	13	14

generated by the method that selects the coefficients of the matrices using a Monte Carlo method [41]:

$$H = \begin{pmatrix} 0 & 15 & 2 & 0 & 0 & 6 & 0 & 4 \\ 15 & 9 & 0 & 11 & 0 & 13 & 0 & 0 \\ 0 & 0 & 3 & 0 & 1 & 0 & 14 & 5 \\ 5 & 0 & 0 & 7 & 3 & 0 & 9 & 0 \end{pmatrix}.$$

From the parity-check matrix $\tilde{H}$ for a CSS construction, the stabilizer group has the following eight noncommuting generators:

$$\begin{array}{rcccccccccc} M_1 & = & I & Z(15) & Z(2) & I & I & Z(6) & I & Z(4) \\ M_2 & = & Z(15) & Z(9) & I & Z(11) & I & Z(13) & I & I \\ M_3 & = & I & I & Z(3) & I & Z(1) & I & Z(14) & Z(5) \\ M_4 & = & Z(5) & I & I & Z(7) & Z(3) & I & Z(9) & I \\ M_5 & = & I & X(15) & X(2) & I & I & X(6) & I & X(4) \\ M_6 & = & X(15) & X(9) & I & X(11) & I & X(13) & I & I \\ M_7 & = & I & I & X(3) & I & X(1) & I & X(14) & X(5) \\ M_8 & = & X(5) & I & I & X(7) & X(3) & I & X(9) & I \end{array}$$

From Lemma 25, the symplectic basis and the isotropic basis generating the stabilizer group can be calculated. The function $\psi(x)$ is $\frac{tr_{2^4/2}(x)}{2}$, where $tr_{p^m/p}(x)$ is the absolute trace from $\mathbf{F}_{p^m}$ to $\mathbf{F}_p$ defined as $tr_{p^m/p}(x) = \sum_{k=0}^{m-1} x^{p^k}$. The additive character χ is given by $\chi(x) = \exp(\pi i tr_{2^4/2}(x))$.

$$\begin{array}{rcccccccccc}
Z_1 &=& I & Z(15) & Z(2) & I & I & Z(6) & I & Z(4) \\
X_1 &=& X(15) & X(9) & I & X(11) & I & X(13) & I & I \\
Z_2 &=& I & I & Z(3) & I & Z(1) & I & Z(14) & Z(5) \\
X_2 &=& I & I & X(3) & I & X(1) & I & X(14) & X(5) \\
Z_3 &=& I & X(15) & X(2) & I & I & X(6) & I & X(4) \\
X_3 &=& Z(15) & Z(9) & I & Z(11) & I & Z(13) & I & I \\
Z_4 &=& Z(5) & Z(15) & Z(6) & Z(7) & Z(9) & Z(6) & Z(4) & Z(8) \\
Z_5 &=& X(5) & X(15) & X(6) & X(7) & X(9) & X(6) & X(4) & X(8)
\end{array}$$

There are three pairs (Z_i, X_i) for $1 \le i \le 3$ and two generators Z_4 and Z_5 in the isotropic basis. Thus, we can construct the [[8,3;3]] LDPC code over $GF(16)$ having the following generators with appended qudits.

$$\begin{array}{rl cccccccc|ccc}
\overline{Z_1} &=& I & Z(15) & Z(2) & I & I & Z(6) & I & Z(4) & Z(13) & I & I \\
\overline{X_1} &=& X(15) & X(9) & I & X(11) & I & X(13) & I & I & X(1) & I & I \\
\overline{Z_2} &=& I & I & Z(3) & I & Z(1) & I & Z(14) & Z(5) & I & Z(15) & I \\
\overline{X_2} &=& I & I & X(3) & I & X(1) & I & X(14) & X(5) & I & X(1) & I \\
\overline{Z_3} &=& I & X(15) & X(2) & I & I & X(6) & I & X(4) & I & I & Z(13) \\
\overline{X_3} &=& Z(15) & Z(9) & I & Z(11) & I & Z(13) & I & I & I & I & X(13) \\
\overline{Z_4} &=& Z(5) & Z(15) & Z(6) & Z(7) & Z(9) & Z(6) & Z(4) & Z(8) & I & I & I \\
\overline{Z_5} &=& X(5) & X(15) & X(6) & X(7) & X(9) & X(6) & X(4) & X(8) & I & I & I
\end{array}$$

5. GENERALIZED FAULT-TOLERANT QUANTUM COMPUTATION OVER NICE RINGS

In this chapter, generalization of fault-tolerant quantum computation over nice rings, which was published in [20][1], will be discussed.

Quantum error correcting codes can protect the quantum information system against errors that are due to the decoherence. However, when building a fault-tolerant quantum computing device, it is not enough to merely deploy quantum error correcting codes. A large part of the errors stem from operational errors due to imperfect gate operations. Furthermore, controlled-not gates likely spread the errors to other qudits. Unless we take great care, the resulting cascade of errors will quickly exceed the error-correction capability of the quantum code. Therefore, it is very important to have a good scheme to avoid such a desperate situation for constructing the fault-tolerant quantum computing devices.

A simple way for realizing fault-tolerant quantum operations are transversal operations. A transversal operation acts in a bit-wise fashion, so that the fault-tolerant gate does not spread errors within the same block of qudits. Thus, it is easily seen that transversal operations are fault-tolerant. However, one cannot achieve computational universality with transversal operations alone, as was shown by Eastin and Knill [14].

Recently, it has been shown that universal fault-tolerant quantum computation is possible in the binary case with transversal gates and quantum error correction [15]. Using the binary triorthogonal matrix [42], the transversal version of Controlled-

[1] 2016 IEEE. Reprinted, with permission, from Sangjun Lee and Andreas Klappenecker, Generalized fault-tolerant quantum computation over nice rings, 2016 IEEE International Symposium on Information Theory Proceedings (ISIT), July 2016

Controlled Z (CCZ) operations are indeed logical operations on the binary triorthogonal stabilizer codes [15]. In this chapter, we will show how to generalize the triorthogonal matrix and the triorthogonal stabilizer code over a nice ring [20]. Based on these tools, it will be shown that the transversal CCZ operation is the logical CCZ operation on the triorthogonal stabilizer codes over nice rings.

5.1 Transversal Clifford Operations over Nice Rings

5.1.1 Transversal Fourier Transform

In the d dimensions, where d is a prime, the Hadamard operation is defined as the d-dimensional discrete Fourier transform [43]

$$H|x\rangle = \frac{1}{\sqrt{d}} \sum_{b=0}^{d-1} \omega^{xb}|b\rangle,$$

where $x \in \mathbb{F}_d$ and $\omega = e^{\frac{2\pi i}{d}}$. More generally for $\mathbb{F}_q$, where q is a power of a prime,

$$H|x\rangle = \frac{1}{\sqrt{q}} \sum_{b=0}^{q-1} \omega^{tr(xb)}|b\rangle,$$

where $x \in \mathbb{F}_q$, $\omega = e^{\frac{2\pi i}{p}}$ and tr is the absolute trace of the finite field $\mathbb{F}_q$ to its prime subfield.

With the help of the character, the form can be described more simply. By substituting the primitive root of the unity ω with the character χ, the Hadamard operation can be generalized for a nice ring.

Definition 28. *Let R be a nice ring. Then, for $x \in R$,*

$$H|x\rangle = \frac{1}{\sqrt{|R|}} \sum_{y \in R} \chi(xy)|y\rangle$$

is a Hadamard operation, where χ is a generating irreducible character of the additive group $(R, +)$, meaning that all other irreducible characters of $(R, +)$ can be expressed in the form $x \mapsto \chi(ax)$ for $a \in R$.

The adjoint of the Hadamard operation $H^\dagger$ also can be defined as follows.

Definition 29. *Let R be a nice ring. Then, for $x \in R$,*

$$H^\dagger|x\rangle = \frac{1}{\sqrt{|R|}} \sum_{z \in R} \chi(-(zx))|z\rangle,$$

where $H^\dagger$ is a complex conjugate of transpose Hadamard operation, where χ is the irreducible character of the additive group $(R, +)$.

We recall the CSS code construction over nice rings as follows.

Proposition 30 ([44])**.** *Let R be a nice ring. Let C_1 and C_2 denote two classical linear codes with parameters $[n, k_1, d_1]_R$ and $[n, k_2, d_2]_R$ such that $C_2^\perp \leq C_1$. Then there exists an $[[n, k_1 + k_2 - n, d]]_R$ stabilizer code with minimum distance $d = \min\{wt(c) | c \in (C_1 \setminus C_2^\perp) \cup (C_2 \setminus C_1^\perp)\}$ that is pure to min$\{d_1, d_2\}$.*

Proof. See the proof of Lemma 9 in [44]. □

We will call a CSS code self-orthogonal if and only if there exists a classical code C containing its dual code, $C^\perp \subseteq C$, such that $C_1 = C_2 = C$.

According to [8], given the arbitrary unitary transformation U and the operator M, $UM|\psi\rangle = UMU^\dagger U|\psi\rangle$. This implies that $|\psi\rangle$ is an eigenvector of M if and only if $U|\psi\rangle$ is an eigenvector of the $UMU^\dagger$. When we restrict our attention to the stabilizer codes, it turned out that the transformed $UMU^\dagger$ should be in the stabilizer S for all $M \in S$ [8]. When it comes to the transversal Hadamard operation for the fault-tolerant quantum computation, we need to check what $H^{\otimes n} M H^{\dagger \otimes n}$ would be for $M \in S$. When you consider CSS quantum codes, the generators of stabilizer consists of X operators or Z operators. Thus, it would be needed to check if $HMH^\dagger$ is either X or Z for $M \in \{X, Z\}$.

Using the shift and multiplication operators, we will determine $HMH^\dagger$ for $M \in \{X(a), Z(b)\}$. To simplify computations, we will assume that the nice ring R is commutative, so $ab = ba$ holds for all $a, b \in R$.

Lemma 31. *Let R be a nice ring. For all $a, b \in R$, we have*

$$HX(a)H^\dagger = Z(a).$$

Proof. It suffices to show that $HX(a)H^\dagger$ acts on the computational basis $\{|x\rangle \mid x \in R\}$ in the same way as $Z(a)$. Indeed,

$$\begin{aligned} HX(a)H^\dagger|x\rangle =& HX(a)\left(\frac{1}{\sqrt{|R|}}\sum_{z\in R}\chi(-(zx))|z\rangle\right) \\ =& \frac{1}{|R|}\sum_{z\in R}\sum_{y\in R}\chi(z(-x+y))\chi(ay)|y\rangle \end{aligned}$$

The character values cancel unless $x = y$, so we get

$$HX(a)H^\dagger|x\rangle = \frac{1}{|R|}\sum_{z\in R}\chi(ax)|x\rangle = \chi(ax)|x\rangle$$

The latter expression is nothing but $Z(a)|x\rangle$. Therefore, we can conclude that

$$HX(a)H^\dagger|x\rangle = Z(a)|x\rangle,$$

holds for all $x \in R$, which proves the claim. □

Lemma 32. *Let R be a nice ring. For $a, b \in R$, we have*

$$HZ(b)H^{\dagger} = X(-b).$$

In general, $X(-b)$ is not equal to $X(b)$ over a nice ring.

Proof. For all $x \in R$, we have

$$\begin{aligned} HZ(b)H^{\dagger}|x\rangle =& HZ(b)\left(\frac{1}{\sqrt{|R|}}\sum_{z\in R}\chi(-(zx))|z\rangle\right) \\ =& \frac{1}{|R|}\sum_{z\in R}\sum_{y\in R}\chi(z(-x+b+y))|y\rangle \end{aligned}$$

The character values cancel unless $y = x - b$, so we get

$$HZ(b)H^{\dagger}|x\rangle = \frac{1}{|R|}\sum_{z\in R}|x-b\rangle = |x-b\rangle.$$

Since the right-hand side is equal to $X(-b)|x\rangle$, the claim follows. □

The stabilizer S is the abelian subgroup of the error group E_n such that the codewords are fixed by all elements of S. For a self-orthogonal CSS code, the stabilizer has a symmetry, that is, it is possible to have the same set of generators consisting of X operators by replacing Z's in the set of generators consisting of Z operators by X's or vice versa. From two previous lemmas, we can conclude that applying the generalized Hadamard operators to the codeword does not change the stabilizer in the following.

Theorem 33. *A self-orthogonal CSS code over a nice ring affords a transversal Hadamard operation.*

Proof. We know that $HX(a)H^{\dagger} = Z(a)$ from Lemma 31 and $HZ(b)H^{\dagger} = X(-b)$

from Lemma 32. The first one implies that any generator in the stabilizer consisting of X operators can be converted to the generator in the stabilizer consisting of Z operators by a conjugation of the Hadamard operation. When any generator in the stabilizer consisting of Z operators is converted by the Hadamard conjugation, the transformed operator is the inverse of a generator in the stabilizer consisting of X operators since all indices after the transformation are the inverses of the indices before the transformation and $X(b)X(-b) = X(-b)X(b) = I$. Since the stabilizer is a group, there should be an inverse of any elements in S, which means that the transformed operator is still a generator in S. Therefore, the stabilizer is invariant under the transversal Hadamard operation. □

5.1.2 Transversal SUM Gate

In order to construct the Clifford group, we also need to consider two-qudit operations, and one of the good candidates is the generalized controlled-NOT operation, or the SUM gate [43].

Definition 34. *Let R be a nice ring. Then, for $x, y \in R$,*

$$\mathrm{SUM}\,|x\rangle|y\rangle = |x\rangle|y+x\rangle$$

*is a generalized controlled-*NOT *operation, or a* SUM *gate.*

The following is the adjoint of the generalized CNOT defined above.

Definition 35. *Let R be a ring. Then, for $x, y \in R$,*

$$\mathrm{SUM}^{\dagger}\,|x\rangle|y\rangle = |x\rangle|y-x\rangle,$$

where $\mathrm{SUM}^{\dagger}$ *is a complex conjugate of transpose* SUM *gate.*

It is well-known that any self-orthogonal CSS code over a binary field affords a transversal CNOT, cf. [8]. We will now show that this can be generalized to self-orthogonal CSS codes over nice rings.

Lemma 36. *Let R be a nice ring. For $a, b \in R$,* SUM *gate maps by a conjugation*

$$\begin{aligned} X(a) \otimes I &\to X(a) \otimes X(a), \\ I \otimes X(a) &\to I \otimes X(a), \\ Z(b) \otimes I &\to Z(b) \otimes I, \\ I \otimes Z(b) &\to Z(-b) \otimes Z(b). \end{aligned}$$

Proof. For all elements x, y in R, the SUM gate acts by conjugation on an X gate in the form

$$\begin{aligned} \mathrm{SUM}(X(a) \otimes I)\,\mathrm{SUM}^\dagger\,|x\rangle|y\rangle &= \mathrm{SUM}\,|x+a\rangle|y-x\rangle \\ &= |x+a\rangle|y+a\rangle \\ &= X(a)|x\rangle \otimes X(a)|y\rangle \end{aligned}$$

and

$$\begin{aligned} \mathrm{SUM}(I \otimes X(a))\,\mathrm{SUM}^\dagger\,|x\rangle|y\rangle &= \mathrm{SUM}\,|x\rangle|y-x+a\rangle \\ &= |x\rangle|y+a\rangle \\ &= |x\rangle \otimes X(a)|y\rangle. \end{aligned}$$

The action of the SUM gate on a Z gate is given by

$$\begin{aligned} \mathrm{SUM}(Z(b) \otimes I)\,\mathrm{SUM}^\dagger\,|x\rangle|y\rangle &= \mathrm{SUM}\,\chi(bx)|x\rangle|y-x\rangle \\ &= \chi(bx)|x\rangle|y\rangle \\ &= Z(b)|x\rangle \otimes |y\rangle \end{aligned}$$

and

$$\begin{aligned} \mathrm{SUM}(I \otimes Z(b))\,\mathrm{SUM}^\dagger\,|x\rangle|y\rangle &= \mathrm{SUM}\,\chi(b(y-x))|x\rangle|y-x\rangle \\ &= \chi(by)\cdot\chi(-bx)|x\rangle|y\rangle \\ &= Z(-b)|x\rangle \otimes Z(b)|y\rangle, \end{aligned}$$

which proves the claim. □

The above result is similar to the conjugation of the SUM gate in d dimension, where d is a prime [43]. Thus, applying the transversal SUM gates will work well fault-tolerantly on CSS codes over nice rings. These operations keep the stabilizer $S \times S$ invariantly.

In a two dimensional quantum system, the Clifford group can be generated by Hadamard gate, the phase gate, and the controlled-NOT gate. In order to generate the Clifford group over a nice ring, we may need to have the generalized phase gate.

5.1.3 Transversal Phase Gate

We already know that a multiplication operator $Z(b)$ and a phase gate S commute, that is, $Z(b) \longmapsto_S Z(b)$. When it comes to an addition operator $X(a)$ and a phase gate S, we want to have the following conjugation action, $X(a) \longmapsto_S X(a)Z(a)$ or $Z(a)X(a)$. For the finite fields of odd prime, Gottesman [43] showed $X \longmapsto_S XZ$ using the phase gate defined by $|j\rangle \to \omega^{j(j-1)/2}|j\rangle$, and Clark [45] proved $X \longmapsto_S ZX$ using the phase gate defined by $|j\rangle \to \omega^{j(j+1)/2}|j\rangle$. We will extend our scope to a nice ring R.

To investigate the phase gate over a nice ring, we use Weyl-Heisenberg operators. The definition of Weyl-Heisenberg operators [46] is

$$\omega(p,q,t) = \chi(t)\omega(p,q) = \chi(t - 2^{-1}pq)Z(p)X(q).$$

Since 2 is in general not a unit in a nice ring, we cannot define its inverse 2^{-1}. To avoid such an issue, we introduce new phase factor $\chi'(a) = \chi(a)^{1/2}$, which can be defined since $\chi(a) \in \mathbb{C}$. From this new factor, the Weyl-Heisenberg operators are

$$\omega(p, q, t) = \chi'(2t - pq)Z(p)X(q) = \omega'(p, q, 2t).$$

Those Weyl operators are closed under multiplication up to phase factors, which is shown below

$$\omega'(p_1, q_1, 2t_1)\omega'(p_2, q_2, 2t_2) = \omega'(p_1 + p_2, q_1 + q_2, 2t_1 + 2t_2 + p_1q_2 - q_1p_2).$$

Now, we define the phase gate S by

$$S|j\rangle = \chi'(j(j+1))|j\rangle,$$

for all $j \in R$. Then, the phase gate S acts by conjugation on the Weyl operator as follows.

Lemma 37. *Let R be a nice ring. For $a, b \in R$, the phase gate S maps by a conjugation*

$$\begin{aligned} Z(b) &\longmapsto Z(b), \\ X(a) &\longmapsto \chi'(c)Z(a)X(a), \end{aligned}$$

where $c = -a^2 + a$.

Proof. We will determine the conjugation action of the phase gate S on a Weyl operator. For all $j, t \in R$, we have

$$\begin{aligned}
S\omega'(b, a, 2t)|j\rangle &= \chi'(2t - ab)SZ(b)X(a)|j\rangle \\
&= \chi'(2t - a^2 + a)Z(a)\chi'(-ab)Z(b)X(a)S|j\rangle \\
&= \chi'(2t - a^2 + a)\omega(a, 0)\omega(b, a)S|j\rangle \\
&= \chi'(2t - a^2 + a)\chi'(a^2)\omega(a + b, a)S|j\rangle \\
&= \chi'(2t + a)\omega(a + b, a)S|j\rangle \\
&= \omega'(a + b, a, 2t + a)S|j\rangle.
\end{aligned}$$

By respectively substituting $(1, 0)$ and $(0, 1)$ for (b, a), we obtain the claim. □

With the similar definition of the phase gate [45], we can obtain same conjugation actions with some phase factors over a nice ring R.

The conjugate action of the phase gate on $X(a)$ produces the phase factor $\chi'(-a^2 + a)$. Thus, given $(c_1, \cdots, c_n) \in R^n$ corresponding to the stabilizer generator, the conjugating action of the transversal phase gate on this stabilizer generator produces the phase factor $\chi'(\sum_{i=1}^{n}(-c_i^2 + c_i))$. For the self-orthogonal CSS codes, $\sum_{i=1}^{n} c_i^2 = 0$. This means that applying the transversal phase gate does not change the stabilizer like the transversal Hadamard operations if the classical codewords corresponding to the stabilizer generators of the self-orthogonal CSS code satisfy the condition that $\sum_{i=1}^{n} c_i = 0$. In other words, the transversal phase gate is the logical operation on the self-orthogonal CSS codes over nice rings when the sum of all entries of each row in the parity-check matrix of the classical code is zero. For the logical $X(a)$ and $Z(b)$, it should be satisfied that given $(a_1, \cdots, a_n) \in R^n$ corresponding to $\overline{X(a)}$, $\sum_{i=1}^{n}(-a_i^2 + a^i) = -a^2 + a$ for all $a \in R$. $Z(b)$ and S gate commute, but $\overline{Z(b)}$ has the same condition as $\overline{X(b)}$ does since they have the same entries.

5.2 Transversal Non-Clifford Operations over Nice Rings

All previous operations belong to the Clifford group, so they cannot provide a universal set of gates. An example of a non-Clifford operation is given by the generalized Toffoli gate that we introduce here (which generalizes the Toffoli gates given in [43] and [7]). If the nice rings is a field, then Nebe, Rains, and Sloane [47] showed that the Clifford group is a projective maximal finite subgroup of the unitary group, so adding the Toffoli gate gives a universal set of gates. It seems likely that a similar result holds for nice rings in general. We will study the controlled-controlled Z gate from which the generalized Toffoli gate can be obtained by conjugation.

5.2.1 Transversal Controlled-Controlled Z Gates

In order to implement the transversal Controlled-Controlled Z gates over nice rings, we introduce a triorthogonal matrix defined over a nice ring R, which is a generalized version of a binary triorthogonal matrix [42].

We define two forms $|a \cdot b| : R^n \times R^n \to R$ and $|a \cdot b \cdot c| : R^n \times R^n \times R^n \to R$ for all $a, b, c \in R^n$ by

$$|a \cdot b| = \sum_{i=1}^{n} a_i b_i, \qquad |a \cdot b \cdot c| = \sum_{i=1}^{n} a_i b_i c_i,$$

where $a = (a_1, \cdots, a_n), b = (b_1, \cdots, b_n), c = (c_1, \cdots, c_n)$.

For the transversal CCZ operations to work as the logical CCZ operations on the encoded triorthogonal stabilizer codes over nice rings, we need to make the triorthogonal matrix have stronger restrictions or conditions than those in the definition provided before. But, it can be easily seen that the new definition of the triorthogonal matrix is still a generalized version of the definition of the binary triorthogonal matrix.

Definition 38. *Let R be a nice ring. Let G be a $m \times n$ matrix with elements in R. Let $g^1, \cdots, g^m$ are the rows of G. Then, G is triorthogonal if and only if*

1. $|g^i \cdot g^j| = 0$ *for all pairs of rows* $1 \le i < j \le m$,

2. $|g^i \cdot g^i| = 0$ *or is a unit for all* $1 \le i \le m$, *and*

3. $|g^i \cdot g^j \cdot g^k| = \begin{cases} 1, & \textit{if } i = j = k \textit{ and } |g^i \cdot g^i| \textit{ is a unit}, \\ 0, & \textit{otherwise.} \end{cases}$

While the binary triorthogonal matrix just had the condition that $|g^i \cdot g^j \cdot g^k| = 0$ for all distinct triples of rows, the triorthogonal matrix defined above has more restrictions on triples of rows of the triorthogonal matrix as seen above. For binary triorthogonal matrix, it is shown that those conditions hold naturally.

For a binary triorthogonal matrix [42], G_0 and G_1 are denoted by submatrices of G formed by even-weight and odd-weight rows, respectively. Linear subspaces $\mathcal{G}_0, \mathcal{G}_1$, and $\mathcal{G} \subseteq \mathbb{F}_2^n$ are defined as linear subspaces spanned by the rows of G_0, G_1, and G respectively.

For a generalized triorthogonal matrix defined above, the submatrices G_0 and G_1 are formed by self-orthogonal and non-self-orthogonal rows, respectively. Then, $\mathcal{G}_0, \mathcal{G}_1$, and $\mathcal{G} \subseteq R^n$ are defined as submodules by the rows of G_0, G_1, and G respectively. Here, we assume that the first k rows $g^1, \cdots, g^k$ are non-self-orthogonal ones, and the remainders $g^{k+1}, \cdots, g^m$ are the self-orthogonal rows.

Lemma 39. *Suppose G is triorthogonal over R. Then, (i) $\mathcal{G}_1$ is a free submodule of $\mathcal{G}$, (ii) $\mathcal{G}_0 \cap \mathcal{G}_1 = 0$, and (iii) $\mathcal{G}_0 = \mathcal{G} \cap \mathcal{G}^\perp$.*

Proof. Let $f \in \mathcal{G}_1$ such that $f = \sum_{i=1}^{k} x_i g^i$ for $x_i \in R$. If $f = 0$ or $f \in \mathcal{G}_0$, then $|f \cdot g^i| = x_i |g^i \cdot g^i| = 0$ for $1 \leq i \leq k$. Since $|g^i \cdot g^i|$ is a unit for $1 \leq i \leq k$ by definition, $x_i = 0$ for $1 \leq i \leq k$, which proves (i) and (ii).

We know that $f_0 \cdot f = 0$ for all $f_0 \in \mathcal{G}_0$ and $f \in \mathcal{G}$, since rows of G_0 are self-orthogonal and orthogonal to any row of G_1 by definition. Thus, $\mathcal{G}_0 \subseteq \mathcal{G} \cap \mathcal{G}^{\perp}$. Let $f = \sum_{i=1}^{m} x_i g^i \in \mathcal{G} \cap \mathcal{G}^{\perp}$ for $x_i \in R$. Then, $|f \cdot g^i| = x_i |g^i \cdot g^i| = 0$ for $1 \leq i \leq k$. Since $|g^i \cdot g^i|$ is a unit, x_i is 0 for all $1 \leq i \leq k$. Thus, $f \in \mathcal{G}_0$. This proves (iii). □

We define a CSS code $(X, \mathcal{G}_0; Z, \mathcal{G}^{\perp})$ with X-type stabilizers $X(f)$ for $f \in \mathcal{G}_0$, and Z-type stabilizers $Z(g)$ for $g \in \mathcal{G}^{\perp}$.

Lemma 40. *Let R be a commutative nice ring. The CSS code $(X, \mathcal{G}_0; Z, \mathcal{G}^{\perp})$ has k logical qudits, and its logical X and Z operators can be chosen as*

$$\overline{X_i(s)} = X(s \cdot g^i), \qquad \overline{Z_i(t)} = Z(t \cdot g^i),$$

where $g^1, \cdots, g^k$ are the rows of G_1, and s, t are in R.

Proof. Since $\overline{X_i(s)}\ \overline{Z_j(t)} = \chi(st|g^i \cdot g^j|)\overline{Z_j(t)}\ \overline{X_i(s)}$, logical X and Z operators commute if $i \neq j$, and does not commute if $i = j$. This means that those logical operators obey the commutation rules.

We also need to show that the logical X and Z operators commute with all stabilizers. Given Z-type stabilizer $Z(g)$ for $g \in \mathcal{G}^{\perp}$, $X(s \cdot g^i)Z(g) = \chi(s|g^i \cdot g|)Z(g)X(s \cdot g^i) = Z(g)X(s \cdot g^i)$ since $g^i \in \mathcal{G}_1 \subseteq \mathcal{G}$ and $g \in \mathcal{G}^{\perp}$. Given X-type stabilizer $X(f)$ for $f \in \mathcal{G}_0$, $Z(t \cdot g^i)X(f) = \chi(t|g^i \cdot f|)X(f)Z(t \cdot g^i) = X(f)Z(t \cdot g^i)$ since $g^i \in \mathcal{G}_1 \subseteq \mathcal{G}$ and $f \in \mathcal{G}_0 = \mathcal{G} \cap \mathcal{G}^{\perp}$ from Lemma 39 □

Since $Z(g)|f\rangle = \chi(gf)|f\rangle = |f\rangle$ for any $f \in \mathcal{G}_0$ and any $g \in \mathcal{G} + \mathcal{G}^{\perp}$, the encoded state $|0^{\otimes k}\rangle$ is defined as

$$\overline{|0^{\otimes k}\rangle} = |G_0\rangle = \frac{1}{\sqrt{|\mathcal{G}_0|}} \sum_{g \in \mathcal{G}_0} |g\rangle.$$

For any $x = (x_1, \cdots, x_k) \in R^k$, the encoded state $|x\rangle$ is obtained by

$$\begin{aligned} \overline{|x\rangle} &= \overline{X_1(x_1)} \cdots \overline{X_k(x_k)} |G_0\rangle \\ &= \frac{1}{\sqrt{|\mathcal{G}_0|}} \sum_{f \in \mathcal{G}_0 + x_1 g^1 + \cdots + x_k g^k} |f\rangle. \end{aligned}$$

In [15], transversal CCZ operations are implemented on the triorthogonal stabilizer codes over $\mathbb{F}_2$. Now, we will try to show the generalized transversal CCZ operations on the triorthogonal stabilizer codes over nice rings. It was shown that $HZ(b)H^{\dagger} = X(-b)$, so that the transversal generalized Toffoli gate can be implemented if the transversal generalized CCZ operation is implemented. We assume that the triorthogonal stabilizer code to be considered has a single encoded qudit with a non-self-orthogonal row g^1 for simplicity. Let $\mathcal{G}_x = \mathcal{G}_0 + x \cdot g^1$. Then, for $x \in R$, $\overline{|x\rangle} = \frac{1}{\sqrt{|\mathcal{G}_x|}} \sum_{g \in \mathcal{G}_x} |g\rangle$.

$$\begin{aligned} \mathrm{CCZ}(b)^{\otimes n} \overline{|x, y, z\rangle} &= \sum_{l \in \mathcal{G}_x, m \in \mathcal{G}_y, n \in \mathcal{G}_z} \mathrm{CCZ}(b)^{\otimes n} |l, m, n\rangle \\ &= \sum_{l \in \mathcal{G}_x, m \in \mathcal{G}_y, n \in \mathcal{G}_z} \chi(b|l \cdot m \cdot n|) |l, m, n\rangle \end{aligned}$$

In order to check if it is possible to implement the transversal generalized CCZ operations on the triorthogonal stabilizer codes, we need to calculate the value of $|l \cdot m \cdot n|$.

$$|l \cdot m \cdot n| = |(l' + xg^1) \cdot (m' + yg^1) \cdot (n' + zg^1)|,$$

for $l', m', n' \in \mathcal{G}_0$. By definition, the only term that does not vanish is $|g^1 \cdot g^1 \cdot g^1|$. So, $|l \cdot m \cdot n| = xyz|g^1 \cdot g^1 \cdot g^1| = xyz$.

Therefore, the transversal CCZ operation is the logical CCZ operation on the triorthogonal stabilizer codes as follows.

$$\mathrm{CCZ}(b)^{\otimes n}\overline{|x, y, z\rangle} = \chi(bxyz)\overline{|x, y, z\rangle}$$

Using the transversal Fourier transform with a help of the quantum error correcting code, we can construct the transversal Toffoli gate working on the triorthogonal stabilizer codes [15].

6. OPTIMAL KEY EXCHANGE PROTOCOLS FOR UNCONDITIONALLY SECURE KEY DISTRIBUTION SCHEMES

Finally, optimal key exchange protocols will be discussed for unconditionally secure key distribution schemes[1].

The key exchange protocols by Bennett and Brassard [21], Ekert [22], and Kish [23] offer unconditional security for establishing a common secret between two parties. The common secret allows the two parties to establish a shared key. These three protocols achieve their remarkable feat by resting their security on physical principles rather than on computational ones. However, all three protocols assume a dedicated communication channel between the two parties, which can be impractical when the number of parties grows. In practice, a communication network will connect the different parties. Therefore, it is natural to ask how quickly a group of people can establish shared secrets between any two parties of the group using the unconditionally secure point-to-point key exchange protocols [21], [22] or [23] over a communication network? Naturally, the answer to this question depends on the topology of the network and the number of key exchanges that can be done in parallel by a party.

Gonzalez, Balog, and Kish [48] started to investigate this question and gave asymptotic results for daisy chain, fully connected, and star network topologies. We will improve upon their results and furthermore give optimal or near optimal result for arbitrary network topologies.

[1] This chapter contains the unpublished paper "Optimal Key Exchange Protocols for Unconditionally Secure Key Distribution Schemes" by S. Lee and A. Klappenecker [24].

6.1 Switched Star Networks

Suppose that m parties communicate over a star network. Each of the participants is connected to a central crossbar switch. If parties a and b are exchanging a key, then no one else can exchange a key with either a or b at the same time, but other pairs in $\{1, 2, \ldots, m\} \setminus \{a, b\}$ can do a key exchange in parallel.

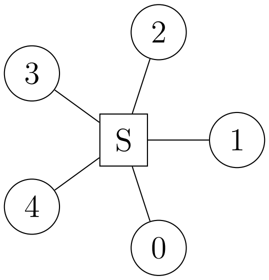

Figure 6.1: A star network connecting 5 participants to one central crossbar switch S. Two key exchanges can be done in parallel per round, but not more. A total of five rounds are needed so that all 5 participants can exchange keys.

A round consists of key exchanges that can be done in parallel (e.g. in a star network with five nodes, we could have the following key exchanges in parallel $\{1, 2\}$ and $\{3, 4\}$, but 0 would have to sit out this round). The next theorem shows that m parties can exchange all keys over a star network in m rounds when m is odd, and in $m - 1$ rounds when m is even.

Theorem 41. *Suppose that m nodes are connected by a star network, where m is an integer greater than 1. Then $2\left\lfloor \frac{m-1}{2} \right\rfloor + 1$ rounds of point-to-point key exchanges (using the protocols [21] or [23]) are necessary and sufficient to exchange all $\binom{m}{2}$ pairs of keys among the m nodes.*

The proof of this theorem will follow from Proposition 43 for the case m even and Proposition 44 for the case m odd. We will first need to prove a simple lemma.

Lemma 42. *Let G be an abelian group of order $2n-1$. Then $f\colon G \to G$ given by $f(x) = 2x$ is an injective function.*

Proof. If g and h are elements of G such that $2g = 2h$, then

$$g = g + (2n-1)g = n(2g) = n(2h) = h + (2h-1)h = h,$$

so the function f is injective, as claimed. □

Proposition 43. *Suppose that $m = 2n$ nodes are connected by a star network. Then it is possible to establish $\binom{2n}{2}$ pairwise key-exchanges in $2n-1$ rounds. No protocol can establish this in fewer rounds.*

Proof. Let us label the $2n$ nodes of the network by the elements of an abelian group G of order $2n-1$ and an additional element ∞ that is not contained in G. In a single round, each node can establish a key exchange with exactly one other node. It is allowed that disjoint pairs of nodes can exchange their keys in parallel, so at most n pairs of nodes can exchange a key in one round.

We will parameterize the round by an element of the group G. A round with parameter $g \in G$ consists of the following pairs of key exchanges:

$$\Gamma_g = \{\{\infty, g\}\} \cup \{\{h, k\} \in G \times G \mid h + k = 2g, h \neq k\}.$$

An element $h \in G$ is paired with the element $k = 2g - h$. We have $h = k$ if and only if $2g = 2h$. In an abelian group of odd order, $2g = 2h$ implies that $g = h$ by the previous lemma. Therefore, Γ_g contains n sets of pairs.

If g and g' are two distinct elements of G, then $\Gamma_g \cap \Gamma_{g'} = \emptyset$. Seeking a contradiction, let us assume that there exists a pair $\{h, k\}$ that is contained in both Γ_g and $\Gamma_{g'}$. The pair $\{h, k\}$ cannot contain ∞, so both h and k must be elements of G. Since $\{h, k\}$ is contained in both sets, we must have $2g = 2g'$, whence $g = g'$, contradicting the fact that the elements g and g' are distinct.

Therefore,

$$\bigcup_{g \in G} \Gamma_g$$

contains $n(2n - 1) = \binom{2n}{2}$ distinct pairs of nodes. We can conclude that after executing the $2n - 1$ rounds Γ_g with $g \in G$, we have established a key exchange between any pair of the $2n$ nodes. Since each round permits at most n key exchanges, we cannot have fewer rounds to exchange $\binom{2n}{2}$ keys, so the protocol is optimal. □

Proposition 44. *Suppose that $m = 2n - 1$ nodes are connected by a star network. Then it is possible to establish $\binom{2n-1}{2}$ pairwise key-exchanges in $2n - 1$ rounds. No protocol can establish this in fewer rounds.*

Proof. Let us label the $2n - 1$ nodes of the network by the elements of an abelian group G of order $2n - 1$. In a single round, each node can establish a key exchange with exactly one other node. It is allowed that disjoint pairs of nodes can exchange their keys in parallel, so at most $n - 1$ pairs of nodes can exchange a key in one round.

We will parameterize the round by an element of the group G. A round with parameter $g \in G$ consists of the following pairs of key exchanges:

$$\Delta_g = \{\{h, k\} \in G \times G \mid h + k = 2g, h \neq k\}.$$

An element $h \in G$ is paired with the element $k = 2g - h$. We have $h = k$ if and only if $2g = 2h$. In an abelian group of odd order, $2g = 2h$ implies that $g = h$ by the previous lemma. Therefore, Δ_g contains $n-1$ sets of pairs.

If g and g' are two distinct elements of G, then $\Delta_g \cap \Delta_{g'} = \emptyset$, since $\Delta_g \subset \Gamma_g$ and $\Delta_{g'} \subset \Gamma_{g'}$ and $\Gamma_g \cap \Gamma_{g'} = \emptyset$, using the notation of the proof of the previous theorem.

Therefore,

$$\bigcup_{g \in G} \Delta_g$$

contains $(n-1)(2n-1) = \binom{2n-1}{2}$ distinct pairs of nodes. We can conclude that after executing the $2n-1$ rounds Δ_g with $g \in G$, we have established a key exchange between any pair of the $2n-1$ nodes. Since each round permits at most $n-1$ key exchanges, we cannot have fewer rounds to exchange $\binom{2n-1}{2}$ keys, so the protocol is optimal. □

Example 45. *Consider a star network with 5 participants. According to the previous proposition, we will need at least five rounds to exchange the keys among all participants. The proof details a way to accomplish these key changes. They are explicitly given by the following rounds of key exchanges:*

$$\begin{aligned}
\Delta_0 &= \{\{1,4\},\{2,3\}\},\\
\Delta_1 &= \{\{0,2\},\{3,4\}\},\\
\Delta_2 &= \{\{0,4\},\{1,3\}\},\\
\Delta_3 &= \{\{0,1\},\{2,4\}\},\\
\Delta_4 &= \{\{0,3\},\{1,2\}\},
\end{aligned}$$

where we have used the cyclic group $G = \mathbf{Z}/5\mathbf{Z}$ with addition modulo 5.

6.2 General Network Topologies

In the previous section, we worked out the key exchange protocol for the star network. Since every participant can exchange a key with every other participant, the underlying communication structure can be represented by a complete graph. However, one limitation of the star network is that each participant can exchange *at most one key per round.*

In this section, we will discuss a much more general setup. We assume that a participant p has several ports that allow her to exchange keys with up to $m(p)$ other participants in parallel per round. The number $m(p)$ of communication ports is a positive integer that can vary depending on the participant p. For instance, a participant p_1 may have $m(p_1) = 3$ different communication ports, whereas participant p_2 may have just $m(p_2) = 2$ communication ports. A larger number of ports increases the cost, but may reduce the number of rounds until all keys are exchanged.

In general, we might not need to secure communication between all parties of a communication network. The key exchange multigraph models each participant by a node and contains an edge between two nodes if and only if the participants are going to exchange keys. We even allow several edges between nodes, so that we can also model the parallel exchange of several keys between two participants in a single round. Every finite multigraph can occur as a key exchange multigraph, but we assume that the multigraph does not contain loops, since no one needs to exchange a key with herself.

It is instructive to look at some small examples. Figure 6.2 shows a network with three participants that are connected with multiple edges.

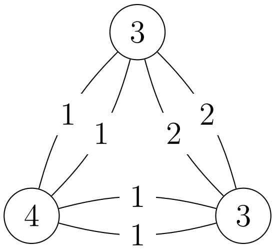

Figure 6.2: A key exchange multigraph for three participants. The vertices are labeled with the number of ports. Each edge is labeled with the round in which the key exchange is performed. Two rounds of key exchanges are needed, since one node v is of degree 4, but has only 3 ports. Therefore, $\lceil d(v)/m(v) \rceil = \lceil 4/3 \rceil = 2$ rounds are needed by any key exchange protocol. The given two-round key exchange protocol is optimal.

We can contrast this with the key exchange protocol for three participants shown in Figure 6.3.

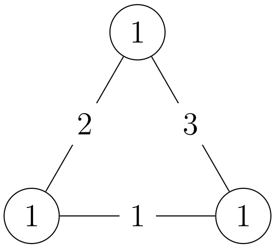

Figure 6.3: A key exchange multigraph for three participants. Each participant has a single communication port and can exchange just a single key per round. In this configuration, it is not possible to exchange more than one key per round overall, so three rounds are needed. This three-round key exchange protocol is optimal as well.

Let us fix some notation. Let $G = (V, E)$ be a multigraph with vertex set V and edge set E. Let

$$\binom{V}{2} = \{\{x, y\} \mid x, y \in V, x \neq y\}$$

be the set of unordered pairs in V. We chose to represent each edge by a unique identifier e in the set of edges E. We denote by $b\colon E \to \binom{V}{2}$ the map that associates to the edge the pair of incident vertices. In other words, $b(e) = \{u, v\}$ means that the edge e is incident to the vertices u and v.

Let $\mu(u, v)$ be the number of edges between the nodes u and v, that is,

$$\mu(u, v) = |\{e \in E \mid b(e) = \{u, v\}\}|.$$

We denote by $N(v)$ the neighborhood of the node v, that is, the set of all nodes in V that are connected by some edge to v. We write $\mu(u)$ for the maximal multiplicity of any edge incident with the vertex u; thus,

$$\mu(u) = \max\{\mu(u, v) \mid v \in N(u)\}.$$

The number of edges that are incident with a vertex v is called the degree $d(v)$ of the vertex. We denote the maximal degree of the graph $G = (V, E)$ by $\Delta(G)$. In other words, $\Delta(G) = \max\{d(v) \mid v \in V\}$.

Let $R_m(G)$ denote the minimal number of rounds needed to exchange all keys between any parties that are connected by an edge in the multigraph G.

Theorem 46. *Let $G = (V, E)$ denote the key exchange multigraph, and let $m\colon V \to \mathbf{N}^*$ be the function assigning the number of communicator ports to each vertex. Then the minimal number $R(G, m)$ of rounds that any protocol needs to exchange keys between all parties that are connected by an edge in G satisfies*

$$\max_{v \in V} \left\lceil \frac{d(v)}{m(v)} \right\rceil \leq R(G, m) \leq \max_{v \in V} \left\lceil \frac{d(v) + \mu(v)}{m(v)} \right\rceil .$$

If $\mu(v) \leq m(v)$, then the lower and upper bounds on $R(G, m)$ differ by at most 1.

Proof. Let $C\colon E \to \mathbf{N}^*$ denote a function that assigns a positive integer to each edge. For an edge e in E, the number $C(e)$ denotes the round during which a key exchange between the vertices u and v is performed, where $\{u, v\} = b(e)$. Thus, the assignment of rounds is an edge coloring problem of sorts.

Not all possible colorings C will lead to a feasible key exchange schedule. One serious restriction is that a node v can perform at most $m(v)$ key exchanges per round. Let $n_i(v)$ denote the number of edges e of color i that are incident with v, that is,

$$n_i(v) = |\{e \in E \mid C(e) = i, v \in b(e)\}|.$$

We call C a proper m-coloring for the multigraph G if and only if for each vertex v in V and each color i, we have $n_i(v) \leq m(v)$. A coloring C provides a feasible key exchange schedule if and only if it is a proper m-coloring.

We are interested in the minimal number of rounds $R(G, m)$ of any key exchange protocol for the key exchange multigraph G. In terms of coloring, this means that we want to find the smallest integer k such that there exist a proper m-coloring with k colors, but no such coloring exists with fewer colors. In other words, the minimal number of rounds $R(G, m)$ in a key exchange protocol for G and the fewest possible colors k of a proper m-coloring of G coincide.

By the generalized pigeonhole principle, if $d(v)$ edges are incident to a vertex v, and the node v has $m(v)$ communication ports, then at least $\lceil d(v)/m(v) \rceil$ rounds of key exchanges are needed before v is able to complete the key exchanges with all its $d(v)$ neighbors. Therefore, the minimal number $R(G, m)$ of key exchange rounds satisfies

$$\max_{v \in V} \left\lceil \frac{d(v)}{m(v)} \right\rceil \leq R(G, m).$$

It was shown in [49, Theorem 3] that

$$k = \max_{v \in V} \left\lceil \frac{d(v) + \mu(v)}{m(v)} \right\rceil$$

colors suffice to give the multigraph G a proper m-coloring. Therefore,

$$R(G, m) \leq \max_{v \in V} \left\lceil \frac{d(v) + \mu(v)}{m(v)} \right\rceil .$$

If the number of ports is equal to 1 for all vertices v, then this is know as Vizing's theorem. The bound ensures that $km(v) \geq d(v) + \mu(v)$, so at each node there are always $\mu(v)$ colors available when coloring with k colors. The proof given in [49, Theorem 3] shows that this gives enough freedom to ensure the existence of a proper m-coloring with k colors. □

Let us consider some examples. Let us denote by C_n the cycle graph on n vertices. Each vertex of the cycle graph is of degree 2. A cycle graph has n edges.

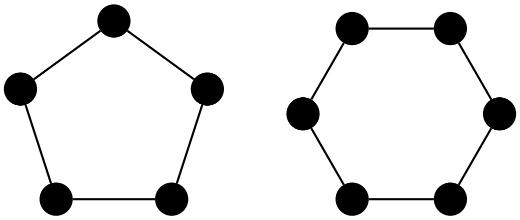

Figure 6.4: The cycle graphs with 5 and 6 vertices.

The next proposition determines the minimal number of key exchange rounds that are needed in the cycle network for all possible configurations of ports.

Proposition 47. *Consider a network of n nodes that form a cycle graph C_n, so each node performs a key exchange with precisely two neighbors. Then the minimal number $R(C_n, m)$ of key exchange rounds for a given configuration m of communication ports is given as follows.*

(i) If n is even and $m(v) = 1$ for some vertex v, then $R(C_n, m) = 2$.

(ii) If n is odd and $m(v) = 1$ for all vertices v, then $R(C_n, m) = 3$.

(iii) If n is odd and $m(v) = 1$ for some vertex v and $m(u) \geq 2$ for some other vertex u, then $R(G, m) = 2$.

(iv) If $m(v) \geq 2$ for all vertices v, then $R(C_n, m) = 1$.

Note that the cases (i) and (iv) cover all situations for cycle graphs with an even number of vertices, and (ii)–(iv) cover all situations for cycle graphs with an odd number of vertices.

Proof. (i) If n is even, then coloring the edges of C_n by alternating the colors 1 and 2 yields a proper m-coloring for any m, so $R(G, m) \leq 2$. If $m(v) = 1$ for some vertex v, then $R(G, m) \geq \lceil d(v)/m(v) \rceil = \lceil 2/1 \rceil = 2$. Therefore, the minimal number of key exchange rounds is $R(G, m) = 2$.

(ii) If n is odd and $m(v) = 1$ for all vertices v, then two colors do not suffice. Indeed, each vertex has degree 2, so at least two colors are needed, but the edges would have to alternate in the two colors, which is impossible for an odd number of edges.

(iii) Suppose that n is odd, $m(v) = 1$ for some vertex v, and $m(u) = 2$ for another. We need at least $\lceil d(v)/m(v) \rceil = \lceil 2/1 \rceil = 2$ colors. However, two colors suffice. Indeed, we can color the edges incident with u in the color 1, and alternate the color for the remaining $n - 2$ edges (starting and ending with the color 2). Thus, we can conclude that $R(G, m) = 2$.

(iv) If $m(v)$ is at least two for each vertex v, then coloring each edge with the color 1 is a valid edge m-coloring, so $R(G, m) = 1$. □

6.3 Pass-Through Networks

In this section, we make a small but very significant change to our network model by allowing that communication can pass through nodes. This is best explained with a small example. Consider the linear chain network with four nodes shown in Figure 6.5.

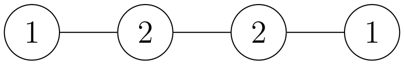

Figure 6.5: Linear chain network with four nodes. The two middle nodes have two communication ports, and the nodes at the ends have just one communication port each.

The obvious drawback of the linear chain network is that only neighboring nodes can establish a key exchange, but the two nodes at the end cannot exchange a key. Let us now assume that a node in the middle can either exchange key with its two neighbors or pass-through the communication and allow neighboring nodes to exchange keys, see Figures 6.6 and 6.7.

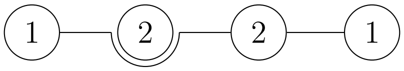

Figure 6.6: Linear chain network with four nodes. The second node from the left is in pass-through mode, so that the first and third nodes can exchange keys.

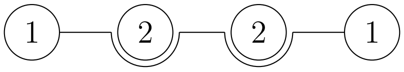

Figure 6.7: Linear chain network with four nodes. Both nodes in the middle are in pass-through mode, so that the first and fourth nodes can exchange keys.

We can now ask the question: How many rounds of key exchanges does it take so that all pairs of nodes in a linear chain network have exchanged keys assuming that pass-through is allowed?

The interesting aspect of this setup is that a simple network structure, such as the linear chain network, can still provide the connectivity of a complete graph. However, the scheduling of the key exchanges is now complicated by the fact that two or more communication links might be involved for key exchanges between non-neighboring nodes.

We will use a hypergraph rather than a graph to model the key exchanges. The hypergraph allows us to model each communication port as a node, so that we can include in a hyperedge all communication ports that are involved in (or disabled by) a key exchange.

For simplicity, let us assume that we are given a linear chain network with n nodes. The node 1 has a single communication port $v_{1,r}$ that connects him to nodes to the right. Each node k in the range $1 < k < n$ has two communication ports, $v_{k,l}$ that allows to communicate with nodes less than k, and $v_{k,r}$ that allows to communicate with nodes greater than k. Finally, the node n has a single communication port $v_{n,l}$.

We can model the linear chain network by a hypergraph as follows. The set of vertices V_n consists of the communication ports

$$V_n = \{v_{1,r},\, v_{2,l},\, v_{2,r},\, \ldots, v_{n-1,l},\, v_{n-1,r},\, v_{n,l}\}.$$

The key exchange between nodes i and j for $1 \leq i < j \leq n$ is modeled by the hyperedge e_{ij} that contains the communication port $v_{i,r}$ of the node i and the communication port $v_{j,l}$ of the node j, and the communication ports $v_{k,l}$ and $v_{k,r}$ of all nodes k in the range $i < k < j$ that are disabled due to pass-through, hence

$$e_{ij} = \{v_{i,r}, v_{i+1,l}, v_{i+1,r}, \ldots, v_{j-1,l}, v_{j-1,r}, v_{j,l}\}.$$

Thus, the hypergraph $H_n = (V_n, E_n)$ modeling the key exchanges for the linear chain network with pass-through has $2n-2$ vertices and a set of $\binom{n}{2}$ hyperedges $E_n = \{e_{ij} \mid 1 \leq i < j \leq n\}$.

Example 48. *The linear chain network with four nodes has six communication ports*

$$V_4 = \{v_{1,r}, v_{2,l}, v_{2,r}, v_{3,l}, v_{3,r}, v_{4,l}\}.$$

and six hyperedges e_{ij} that model the key exchange between the nodes i and j, namely

$$\begin{array}{rclrcl} e_{12} & = & \{v_{1,r}, v_{2,l}\}, & e_{13} & = & \{v_{1,r}, v_{2,l}, v_{2,r}, v_{3,l}\}, \\ e_{14} & = & \{v_{1,r}, v_{2,l}, v_{2,r}, v_{3,l}, v_{3,r}, v_{4,l}\}, & e_{23} & = & \{v_{2,r}, v_{3,l}\}, \\ e_{24} & = & \{v_{2,r}, v_{3,l}, v_{3,r}, v_{4,l}\}, & e_{34} & = & \{v_{3,r}, v_{4,l}\}. \end{array}$$

The key exchange between nodes i and j requires two ports for communication and disabling of $2(j-i+1)$ intermediate ports due to pass-through.

Our goal is to find the optimal number of rounds for key exchanges over the linear chain network with n hosts. This means that we need to find a coloring of the hyperedges of H_n with a minimal number of colors such that any two hyperedges that have a vertex in common receive different colors; in other words, we need to find the chromatic index of H_n.

Since we prefer to work with a graph rather than a hypergraph, we translate the edge coloring problem of the hypergraph H_n into an equivalent vertex coloring problem of the line graph $L(H_n)$. We can form the line graph $L(H_n) = (V_n', E_n')$ of the hypergraph H_n as follows. The set of vertices V_n' of the line graph $L(H_n)$ is given by the set of hyperedges $V_n' = E_n$ of the hypergraph H_n, and the edge set E_n' contains a pair $\{e_{i,j}, e_{k,l}\}$ of hyperedges of H_n if and only if $e_{i,j} \cap e_{k,l} \neq \emptyset$.

The smallest number of colors needed to color the vertices of $L(H_n)$ such that adjacent vertices do not share the same color is called the chromatic number of line graph $L(H_n)$. Since the line graph exchanges the role of hyperedges and vertices, the chromatic index of the hypergraph H_n is the same as the chromatic number of the line graph $L(H_n)$. Therefore, the problem of finding the minimal number of rounds in key exchanges in the line graph when pass-through is allowed is the same as finding the chromatic number of the line graph $L(H_n)$.

It might be instructive to look at an example.

Example 49. *We discussed the hypergraph H_4 in Example 48. The line graph of H_4 is shown in Figure 6.8. Every node of the line graph $L(H_4)$ represents a hyperedge. Two nodes of the line graph are adjacent if and only if the corresponding hyperedges share a vertex. Since the line graph contains a clique with four nodes, we need at least four rounds to exchange all keys. It turns out that four rounds are enough, since we can exchange keys in round 1 between nodes $\{1,4\}$, in round 2 between $\{1,3\}$ and $\{3,4\}$, in round 3 between $\{1,2\}$ and $\{2,4\}$, and in round 4 between $\{2,3\}$.*

The next theorem determines the number of key exchange rounds in the linear chain network with n nodes by deriving the chromatic number of the line graph $L(H_n)$.

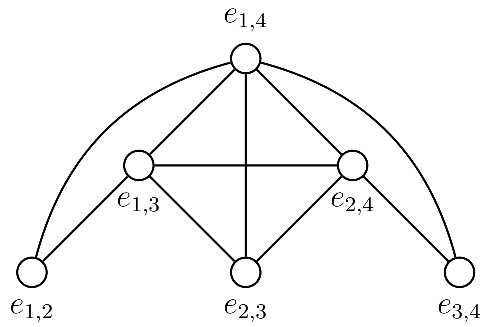

Figure 6.8: The line graph $L(H_4)$ of the hypergraph H_4 representing the linear chain network with 4 hosts. Since we have a clique with four nodes in $L(H_4)$, this means that every vertex coloring of $L(H_4)$ needs at least four colors.

Theorem 50. *The minimal number R_n of key exchange rounds in the linear chain network with n participants with key exchanges between all participants is given by*

$$R_n = \begin{cases} \frac{n^2}{4} & \text{if } n \text{ is even,} \\ \frac{n^2-1}{4} & \text{if } n \text{ is odd.} \end{cases}$$

Proof. We set $k = \lfloor n/2 \rfloor$, so that $n = 2k$ when n is even and $n = 2k + 1$ when n is odd. Let us partition the set $V'_n = E_n$ of vertices of the line graph $L(H_n)$ into the sets

$$A_n = \{e_{ij} \mid 1 \leq i \leq k, i + 1 \leq j \leq k\},$$
$$C_n = \{e_{ij} \mid 1 \leq i \leq k, k + 1 \leq j \leq n\},$$
$$B_n = \{e_{ij} \mid k + 1 \leq i \leq n - 1, i + 1 \leq j \leq n\}.$$

The sets A_n, B_n, and C_n are pairwise disjoint, and their union is the entire set $V'_n = E_n$.

Since each element e_{ij} in the set C_n contains $v_{k,r}$, we can conclude that C_n is a clique in the line graph $L(H_n)$ with m elements, where

$$m = \begin{cases} k^2 = \frac{n^2}{4} & \text{if } n \text{ is even,} \\ k(k+1) = \frac{n^2-1}{4} & \text{if } n \text{ is odd.} \end{cases}$$

Therefore, each vertex coloring of $L(H_n)$ requires at least m different colors.

We can give a vertex coloring C of $L(H_n)$ with m colors as follows.

(a) For e_{ij} in C_n, we set $C(e_{ij}) = k(j-k-1)+i$.

(b) For e_{ij} in A_n, we have $e_{j,n-i+1}$ in C_n and we set $C(e_{ij}) = C(e_{j,n-i+1})$.

(c) For e_{ij} in B_n, we have $e_{n-j+1,i}$ in C_n and we set $C(e_{ij}) = C(e_{n-j+1,i})$.

In this coloring, the m vertices in C_n have pairwise distinct colors.

We will now show that C is a proper coloring of the vertices. For $e_{ij} \in A_n$, the set $\{e_{ij}, e_{j,n-i+1}\}$ is an independent set in $L(H_n)$. Similarly, for $e_{ij} \in B_n$, the set $\{e_{n-j+1,i}, e_{ij}\}$ is an independent set in $L(H_n)$. Since vertices in an independent set can be colored with the same color, we only need to show that each element in $A_n \cup B_n$ receives a different color when assigning colors as in (b) and (c), so no coloring conflicts can arise.

The map $\tau\colon A_n \to C_n$ given by $\tau(e_{ij}) = e_{j,n-i+1}$ is an injective map. The map $\sigma\colon B_n \to C_n$ given by $\sigma(e_{ij}) = e_{n-j+1,i}$ is an injective map as well. The two maps have disjoint images in C_n, since any e_{ij} in A_n yields an image $\tau(e_{ij}) = e_{j,n-i+1}$ satisfying $j+n-i+1 \geq n+2$, whereas any e_{ij} in B_n yields an image $\sigma(e_{ij}) = e_{n-j+1,i}$ satisfying $n-j+1+i \leq n$. Therefore, the map $\rho\colon A_n \cup B_n \to C_n$ given by

$$\rho(e_{ij}) = \begin{cases} \tau(e_{ij}) = e_{j,n-i+1} & \text{if } e_{ij} \in A_n, \\ \sigma(e_{ij}) = e_{n-j+1,i} & \text{if } e_{ij} \in B_n, \end{cases}$$

is injective as well. Since all elements in $A_n \cup B_n$ get assigned different colors, no coloring conflicts will arise. The color assignments in (b) and (c) are proper, since elements in $A_n \cup B_n$ get the same color as elements in C_n only when they form an independent set. $\square$

Remark 51. *Gonzalez, Kish, Balog, and Enjeti have shown in [50] that the number of rounds for key exchanges for the linear chain network with pass-through is at most* $(N+1)^2/4$ *when* N *is odd and* $((N+1)^2-1)/4$ *when* N *is even, where* $N = n-1$. *The previous theorem proves that their key exchange protocol is optimal. Our protocol given here differs from the one given in [50].*

7. CONCLUSIONS

As nice nearrings and its special subclass, nice rings, were introduced in [16], we can construct quantum error correcting codes without the restriction that their alphabet sizes are powers of a prime. In this thesis, two quantum error correcting schemes were generalized over nice rings.

The first quantum code we showed to generalize was the subsystem code. We gave a construction of subsystem codes over nice nearrings generally. Then, by focusing our scope on nice rings, we derived a construction of subsystem codes from classical linear codes over finite Frobenius rings. Furthermore, for the much smaller class of free subsystem codes over finite chain rings, we were able to show that there exists a free subsystem code over a finite field that has the same rate and at least same minimum distance.

For the generalization of the entanglement-assisted quantum error correcting codes, we first showed that the R-module can be decomposed as an orthogonal direct sum of hyperbolic pairs since it is a finite symplectic abelian group with the anti-symmetric bicharacter. From that, it can be possible to have a symplectic basis and a isotropic basis generating a free submodule of R^{2n}. It was shown that appending appropriate entanglement qudits, the noncommuting generators are extended to the commuting generators. The entanglement-assisted quantum error correcting codes over nice rings can be constructed with the extended commuting generators.

When it comes to the generalization of the fault-tolerant quantum computation, we showed that transversal versions of Fourier transform, SUM gate, and phase gate are logical operations on CSS codes over nice rings. For non-Clifford operation, we discussed that the transversal CCZ gate can be performed fault-tolerantly on triorthogonal stabilizer codes over nice rings. Using these transversal Clifford and non-Clifford gates, quantum computation over nice rings can be fault-tolerant and universal.

Finally, the optimal key exchange protocols on three network topologies, the star network topology, the general network topology, and the pass-through network topology, were discussed for unconditionally secure key distribution. For the star network, we can compute the optimal number of rounds needed for the key exchanges between all pairs of given nodes. In order to expand our scope to the more general situation, the multigraph was investigated as the key exchange model, and based on this setting, the quite tight bounds on the minimum number of rounds required for the key exchange were shown. For the pass-through network topology, the hypergraph was introduced to model the network, and its line graph was derived and used to calculate the optimal number of rounds for the key exchange.

REFERENCES

[1] (2013) International Technology Roadmap for Semiconductors (ITRS). [Online]. Available: http://www.itrs.net/

[2] I. L. Markov, "Limits on fundamental limits to computation," *Nature*, vol. 512, no. 7513, pp. 147–154, 2014.

[3] M. M. Shulaker, G. Hills, N. Patil, H. Wei, H.-Y. Chen, H.-S. P. Wong, and S. Mitra, "Carbon nanotube computer," *Nature*, vol. 501, no. 7468, pp. 526–530, 2013.

[4] V. R. Almeida, C. A. Barrios, R. R. Panepucci, and M. Lipson, "All-optical control of light on a silicon chip," *Nature*, vol. 431, no. 7012, pp. 1081–1084, 2004.

[5] 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.

[6] S. Hallgren, "Polynomial-time quantum algorithms for pell's equation and the principal ideal problem," *Journal of the ACM (JACM)*, vol. 54, no. 1, p. 4, 2007.

[7] P. W. Shor, "Fault-tolerant quantum computation," in *37th Annual Symposium on Foundations of Computer Science (Burlington, VT, 1996).* Los Alamitos, CA: IEEE Comput. Soc. Press, 1996, pp. 56–65.

[8] D. Gottesman, "Theory of fault-tolerant quantum computation," *Phys. Rev. A*, vol. 57, no. 1, pp. 127–137, Jan 1998.

[9] D. Gottesman, "An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation," *ArXiv e-prints*, Apr. 2009.

[10] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, no. 2, pp. 1098–1105, Aug 1996.

[11] A. Steane, "Multiple Particle Interference and Quantum Error Correction," *Proc. Roy. Soc. Lond.*, vol. A452, p. 2551, 1996.

[12] D. Gottesman, "Stabilizer codes and quantum error correction," Ph.D. dissertation, California Institute of Technology, 1997.

[13] D. Aharonov and M. Ben-Or, "Fault-tolerant quantum computation with constant error," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, ser. STOC '97. New York, NY, USA: ACM, 1997, pp. 176–188.

[14] B. Eastin and E. Knill, "Restrictions on transversal encoded quantum gate sets," *Physical review letters*, vol. 102, no. 11, p. 110502, 2009.

[15] A. Paetznick and B. W. Reichardt, "Universal fault-tolerant quantum computation with only transversal gates and error correction," *Physical review letters*, vol. 111, no. 9, p. 090505, 2013.

[16] A. Klappenecker, "Nice nearrings," in *2012 IEEE International Symposium on Information Theory Proceedings (ISIT)*, july 2012, pp. 170 –173.

[17] S. Nadella and A. Klappenecker, "Stabilizer codes over frobenius rings," in *2012 IEEE International Symposium on Information Theory Proceedings (ISIT)*, july 2012, pp. 165 –169.

[18] S. Lee and A. Klappenecker, "Subsystem codes over nice nearrings," in *2013 IEEE International Symposium on Information Theory Proceedings (ISIT)*. IEEE, 2013, pp. 912–916.

[19] ——, “Entanglement-assisted quantum error correcting codes over nice rings,” in *52nd Annual Allerton Conference on Communication, Control, and Computing*, Sept 2014.

[20] ——, “Generalized fault-tolerant quantum computation over nice rings,” in *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2016, pp. 2279–2283.

[21] G. Bennett, C.H.; Brassard, “Quantum cryptography: Public key distribution and coin tossing,” *Proc. of IEEE International Conference on Computers, Systems and Signal Processing*, vol. 175, p. 8, 1984.

[22] A. K. Ekert, “Quantum cryptography based on bells theorem,” *Physical review letters*, vol. 67, no. 6, p. 661, 1991.

[23] L. B. Kish, “Totally secure classical communication utilizing johnson (-like) noise and kirchoff’s law,” *Physics Letters A*, vol. 352, no. 3, pp. 178–182, 2006.

[24] S. Lee and A. Klappenecker, “Optimal key exchange protocols for unconditionally secure key distribution schemes,” unpublished.

[25] M. A. Nielsen and I. L. Chuang, “Quantum computation and quantum information,” 2011.

[26] D. Kribs, R. Laflamme, and D. Poulin, “Unified and generalized approach to quantum error correction,” *Phys. Rev. Lett.*, vol. 94, no. 18, p. 180501, May 2005.

[27] D. W. Kribs, R. Laflamme, D. Poulin, and M. Lesosky, “Operator quantum error correction,” *Quantum Info. Comput.*, vol. 6, no. 4, pp. 382–399, Jul. 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=2012086.2012092

[28] S. A. Aly, A. Klappenecker, and P. K. Sarvepalli, "Subsystem codes," in *In 44th Annual Allerton Conference on Communication, Control, and Computing*, vol. 44, no. 1, sep 2006.

[29] A. Klappenecker and P. Sarvepalli, "Clifford code constructions of operator quantum error-correcting codes," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5760 –5765, dec. 2008.

[30] I. M. Isaacs, *Finite Group Theory*, ser. Graduate Studies in Mathematics. American Mathematical Society, 2008, vol. 92.

[31] G. H. Norton and A. Slgean, "On the structure of linear and cyclic codes over a finite chain ring," *Applicable Algebra in Engineering, Communication and Computing*, vol. 10, pp. 489–506, 2000, 10.1007/PL00012382.

[32] G. Norton and A. Salagean, "On the hamming distance of linear codes over a finite chain ring," *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 1060 –1067, may 2000.

[33] T. Brun, I. Devetak, and M. H. Hsieh, "Correcting quantum errors with entanglement," *Science*, vol. 314, no. 5798, pp. 436–439, 2006.

[34] ——, "Catalytic quantum error correction," *arXiv preprint quant-ph/0608027*, 2006.

[35] M. H. Hsieh, I. Devetak, and T. Brun, "General entanglement-assisted quantum error-correcting codes," *Physical Review A*, vol. 76, no. 6, p. 062313, 2007.

[36] E. M. Zmud, "Symplectic geometries over finite abelian groups," *Sbornik: Mathematics*, vol. 15, pp. 7–29, 1971.

[37] G. Karpilovsky, *Group representations*. Elsevier, 1994, vol. 3.

[38] D. Fattal, T. S. Cubitt, Y. Yamamoto, S. Bravyi, and I. L. Chuang, "Entanglement in the stabilizer formalism," *arXiv preprint quant-ph/0406168*, 2004.

[39] S. A. Aly, "A class of quantum ldpc codes constructed from finite geometries," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, Nov 2008, pp. 1–5.

[40] M. H. Hsieh, T. A. Brun, and I. Devetak, "Entanglement-assisted quantum quasicyclic low-density parity-check codes," *Phys. Rev. A*, vol. 79, p. 032340, Mar 2009.

[41] D. J. C. MacKay, "Optimizing sparse graph codes over GF(q)," 2003.

[42] S. Bravyi and J. Haah, "Magic-state distillation with low overhead," *Physical Review A*, vol. 86, no. 5, p. 052329, 2012.

[43] D. Gottesman, "Fault-tolerant quantum computation with higher-dimensional systems," in *Quantum Computing and Quantum Communications.* Springer, 1999, pp. 302–313.

[44] S. Nadella, "Stabilizer codes over frobenius rings," Master's thesis, Texas A&M University, 2012.

[45] S. Clark, "Valence bond solid formalism for d-level one-way quantum computation," *Journal of Physics A: Mathematical and General*, vol. 39, no. 11, p. 2701, 2006.

[46] D. Gross, "Computational power of quantum many-body states and some results on discrete phase spaces," Ph.D. dissertation, Imperial College, 2008.

[47] G. Nebe, E. M. Rains, and N. J. A. Sloane, *Self-dual codes and invariant theory.* Springer, 2006, vol. 17.

[48] E. Gonzalez, R. S. Balog, and L. B. Kish, “Resource requirements and speed versus geometry of unconditionally secure physical key exchanges,” *Entropy*, vol. 17, no. 4, pp. 2010–2024, 2015.

[49] S. Louis Hakimi and O. Kariv, “A generalization of edge-coloring in graphs,” *Journal of Graph Theory*, vol. 10, no. 2, pp. 139–154, 1986.

[50] E. Gonzalez, L. B. Kish, R. S. Balog, and P. Enjeti, “Information theoretically secure, enhanced johnson noise based key distribution over the smart grid with switched filters,” *PloS one*, vol. 8, no. 7, p. e70206, 2013.