Secure quantum key distribution and comparison over noisy channels

Vikesh Siddhu

A dissertation submitted for the partial fulfilment
of BS-MS dual degree in Science

Indian Institute of Science Education and Research Mohali
April 2013

Certificate of Examination

This is to certify that the dissertation titled **Secure quantum key distribution and comparison over noisy channels** submitted by **Mr. Vikesh Siddhu** (Reg. No. MS08053) for the partial fulfillment of BS-MS dual degree programme of the Institute, has been examined by the thesis committee duly appointed by the Institute. The committee finds the work done by the candidate satisfactory and recommends that the report be accepted.

Dr. Kavita Dorai | Dr. Sanjeev Kumar | Prof. Arvind (Supervisor)

Dated: April 26, 2013

Declaration

The work presented in this dissertation has been carried out by me under the guidance of Dr. Arvind at the Indian Institute of Science Education and Research Mohali.

This work has not been submitted in part or in full for a degree, a diploma, or a fellowship to any other university or institute. Whenever contributions of others are involved, every effort is made to indicate this clearly, with due acknowledgement of collaborative research and discussions. This thesis is a bonafide record of original work done by me and all sources listed within have been detailed in the bibliography.

Vikesh Siddhu
(Candidate)

Dated: April 26, 2013

In my capacity as the supervisor of the candidates project work, I certify that the above statements by the candidate are true to the best of my knowledge.

Prof. Arvind
(Supervisor)

Acknowledgment

My supervisor and mentor Prof. Arvind gave me space(physical and mental) and freedom to explore the subject. His valuable guidance and timely advice has allowed me to shape this thesis in its present form. I would like to thank him for this. I hold him in very high regard and have the utmost respect for him.

I would like to thank my brother, Lokesh Siddhu who has been very supportive of my educational endeavors all throughout.

I would like to thank my friends Abhilasha, Ankit, Keshav, Anshu, Zeeshan and Vanika Di who have played a very important part. They gave me a space to enjoy and interact which allowed me to think freely and made my stay at IISER memorable.

The discussions in our research group allowed me to talk about my work. I would like to thank Debmalya, Ritabrata, Shruti Di and Harpreet for patiently listening to me and pointing out lacunas in my understanding.

The project involves an understanding of elementary number theory. Dr. Amit Kulshrestha helped me with the calculations. I would like to thank him for that.

IISER Mohali provided me with a sound environment to work. The INSPIRE fellowship financially supported me, I would like to thank both of these.

I would also like to thank Nikhil for his help with LaTeX. The list of people I have thanked above is obviously not exhaustive, because every single day of study has helped build this report and there are so many people who make these days possible that one can go on and on.

List of Figures

Notation

$\oplus$	bit wise addition
$A^{\otimes n}$	n tensor product of A
$\phi(n)$	number of positive integers co-prime to n less than n
$Tr_e[A]$	Trace of A
$\mathbb{I}$	Identity Matrix(in appropriate dimensions)
$(\mathbb{Z}/n\mathbb{Z})^*$	Multiplicative group mod n
Pauli X or σ_x	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
Pauli Y or σ_y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
Pauli Z or σ_z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
Hadamard H	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$\|0\rangle$ or $\|z+\rangle$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$
$\|1\rangle$ or $\|z-\rangle$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$
$\|x+\rangle$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$
$\|x-\rangle$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$
$\mathbb{C}^n$	Hilbert space, dimension n
$\mathbb{F}_2^n$	Space of n tuple, with entry from $\{0, 1\}$

Abstract

In this report we discuss the subject of Cryptography. Cryptography is a physical or mathematical system of transforming information so that it is undecipherable by polynomial time computational adversaries. In classical cryptography we look at purely mathematical transformations. We discuss the one time pad and the RSA protocol. The former being provably secure while the latter is only conjectured to be secure and moreover the security is algorithmic in nature. The one time pad has limited usage since it requires private exchange of keys.
Secure cryptography is possible if we are able to do secure key distribution. Therefore, in Quantum Cryptography we look at the problem of secure key distribution. Once a secure key distribution is established we can use the one time pad to securely transmit data. Since fundamentally secure key distribution is possible quantum cryptography offers provably secure communication. We discuss various quantum key distribution protocols[BB84, BBM92, Eke91] that work under noiseless conditions.

Since real quantum channels are always noisy we have to consider the noise effect of noise on quantum states passing through noisy channels. In order to protect against errors we do quantum error correction. We present the CSS[CS96, Ste96] protocol for quantum error correction and derive certain general results for the fidelity of the communication protocols. We show that the BB-84 protocol is robust against noise by following the discussion in [SP00]. We then tackle the problem of Quantum Private Communication(QPC) under noise. In QPC the idea is to protect private information during its public comparison. We discuss quantum protocols that work under noiseless conditions [TLH12, WYBW12]. We analyze [TLH12] under bit-phase flip channel and depolarizing channel. We show how noise gives a bound on the length of the string that can be compared using the protocol given in [TLH12]. We then present another protocol based on CSS codes to perform Quantum Private communication under noisy channels. Here we can compare strings of arbitrary length as long as the error rate is under a given level.

Contents

Chapter 1

Classical Cryptography

Cryptography is a physical or mathematical system of transforming information so that it is undecipherable by polynomial time computational adversaries. In classical cryptography we deal with only mathematical transformations. The main idea is to come up with a transformation on a message that allows relevant parties to encrypt and decrypt the message easily but makes it *hard* for malicious parties to decrypt the transformed message.
Let us discuss the two broad categories of classical cryptography i.e. *Private Key Cryptography* and *Public Key Cryptography*.

1.1 Private Key Cryptography

Suppose Alice and Bob wish to exchange data secretly over a classical communication channel like a newspaper or telephone. In private key cryptography both Alice and Bob meet in the past to exchange a secret key. This key can be thought of as a string of 0's and 1's. They can use this secret key to do encryption and decryption of data. An encryption should allow the data to be inaccessible to a polynomial time adversary. A simple example of this scheme is the *One Time Pad*.

Example Let Alice and Bob exchange a key r which is n bit long. Let $x \in \{0,1\}^n$ be a message that Alice wants to send to Bob.
Let $\mathcal{E}_r : \{0,1\}^n \to \{0,1\}^n$ be the encryption protocol such that the message being sent to Alice is $\mathcal{E}_r(x)$. Then the decryption given by $\mathcal{D}_r : \{0,1\}^n \to \{0,1\}^n$ is such that

$$\mathcal{D}_r(\mathcal{E}_r(x)) = x$$

For a one time pad $\mathcal{E}_r(x) = x \oplus r$ and $\mathcal{D}_r(x) = x \oplus r$ where $\oplus$ is bitwise addition. It is

straightforward to see that

$$\mathcal{D}_r(\mathcal{E}_r(x)) = (x \oplus r) \oplus r$$
$$\mathcal{D}_r(\mathcal{E}_r(x)) = x(\oplus r \oplus r)$$
$$\mathcal{D}_r(\mathcal{E}_r(x)) = x$$

Example Let $r = 1000010$ and $x = 1011101$ then the encryption works as follows

$$\mathcal{E}_r(x) = 1011101 \oplus 1000010$$
$$\mathcal{E}_r(x) = 0011111$$

while the decryption works as follows

$$\mathcal{D}_r(\mathcal{E}_r(x)) = 0011111 \oplus 1000010$$
$$\mathcal{D}_r(\mathcal{E}_r(x)) = 1011101$$

Hence we recover the original message.

We now ask the question, How is this secure? It is easy to see that without knowing the key r any eavesdropper cannot faithfully decrypt the original n bit message, but with a probability of 2^{-n}.
Given $\mathcal{E}(x)$ there are 2^n different keys r which can be used to generate 2^n distinct x's since '$\oplus$' is a bijective map. If the eavesdropper chooses a key at random then with probability 2^{-n} she will recover the correct x. So the probability of deciphering the message faithfully is exponentially small.
In the above 'one time' pad protocol we use the key r only once. If we wish to use the 'one-time' pad two or more times then we compromise the security of the protocol. Specifically we can find out

- Whether two messages begin or end with the same bit values

  Example If $x = 101$ and $z = 011$ are two messages sent using the same key r then the eavesdropper can calculate $(x \oplus r) \oplus (z \oplus r) = x \oplus z$. This reveals the locations where two messages x and z have same or different values.

Consequently we have the following drawbacks

- The one time pad needs to be at least as long as the message.
- We need a new pad each time we need to communicate.

1.2 Private Key Cryptography

If Alice and Bob want to have a secure communication without having to meet first, then they want a different protocol. In private key cryptography we make the encryption key public so that Alice and Bob don't have to meet do a secure key exchange. The process of cryptography then works as follows, Bob creates an encryption key and a decryption key. The encryption key is given to Alice through an *insecure* channel. An insecure channel is one which Eve can monitor, but whose contents she cannot change, example: Newspaper. Through this insecure channel Alice returns an encrypted message. In the process Eve has access to both the encryption key and the encrypted message. The challenge is to present a protocol which makes it hard for Eve to decrypt the message using her information. The RSA protocol may do the job.

1.2.1 RSA Protocol

1. Bob chooses two large primes p, q.
2. Compute $n = pq$ and $\phi(n)$[1]
3. Find e such that $gcd(e, \phi(n)) = 1$.
4. Find d such that $ed \equiv 1(mod\ \phi(n))$
5. $P = (e, n)$ is the *public* encryption key and $S = (d, n)$ is the *secret* decryption key.

The encryption on a message x is carried out by $\mathcal{E}$ where,

$$\mathcal{E}(x) = x^e (mod\ (n))$$

while the decryption on $\mathcal{E}(x)$ is performed by $\mathcal{D}$ where,

$$\mathcal{D}(\mathcal{E}(x)) = \mathcal{E}(x)^d (mod\ (n))$$

If x is a message written as number in base 10 and has a bit representation with less than $[log(n)]$ bits then we can use the above encryption and decryption protocol.
It is possible to show that $\mathcal{D}(\mathcal{E}(x)) = x^{de} (mod\ (n)) \equiv x (mod\ (n))$. A full derivation for this is given in Appendix A along with a *python* program that implements RSA.

[1] $\phi(n)$ is the number of numbers co-prime to n less than n

Security

Here we argue why the above protocol is secure. If we look at the decryption step we see that

$$\mathcal{D}(\mathcal{E}(x)) = x(mod\ n)$$

which means that $\mathcal{D}$ is inverse of $\mathcal{E}$.
An eavesdropper with $P = (e, n)$ can decode the message $\mathcal{E}(x)$ for all x if she knows $\mathcal{D}$. In order to find inverse of $\mathcal{E}(x)$ one must be able to solve[2]

$$e.e^{-1} \equiv 1(mod\ \phi(n))$$

This equation can be solved if we know $\phi(n)$. In order to calculate the the Euler Function ϕ for a give number we must know its prime factors! Till now there is no efficient algorithm to calculate prime factors and neither is there a proof showing hardness of prime factorization. Hence it is hard to find the e^{-1} as we can't find ϕ easily. In this sense RSA is secure but its security is has not been proven.

[2]for a justification of this look for details in Appendix A

Chapter 2

Quantum Cryptographic Protocols

Quantum Cryptography deals with the secure distribution of key necessary for private key cryptography. The security in quantum cryptography is provided by the properties of quantum mechanics. In classical cryptography security is provided by the conjectured difficulty of computing certain functions. We discuss here quantum key distribution protocols that work under noiseless conditions. We look at the BB84 protocol[BB84] and the EPR based protocol [Eke91] and later show the equivalence between the two [BBM92].

2.1 BB84 Protocol

BB84 protocol is a quantum key distribution protocol invented by Bennet and Brassard in 1984 and hence the name [BB84]. Here random bits are conveyed through a quantum channel. These bits after consultation over a classical channel can be detected with high probability to have been conveyed securely.
Assume there are two people Alice and Bob who wish to share a random classical string of bits using quantum resources. This classical string of bits can serve as a key for a classical cryptography protocol such as one time pad. We schematically describe the protocol below when there is no noise in the quantum channel.

- Alice chooses a random bit string and a basis X or Z randomly for each bit.
- She sends to Bob a spin half particle through a channel in one of the eigen states of X or Z by using the following scheme

Bit value	Basis	Eigen State
0	X	$\|x+\rangle$
0	Z	$\|z+\rangle$
1	X	$\|x-\rangle$
1	Z	$\|z-\rangle$

- Bob announces the receipt of the particles. Alice and Bob communicate with each other in order to find out if certain transmissions have been unsuccessful and discard such spins.
- Bob measures his partilces in X or Z basis randomly.
- Alice and Bob publicly announce the basis directions they used for their measurements(say in a newspaper). They compare this data and find out where Bob's measurement basis and Alice's sending basis match, lets call these *matching points*.
- At the *matching points* Bob decodes the data by inverting Alice's scheme. With each measurement setting and outcome he associates a classical bit according to the table given below

Eigen Value	Basis	Bit
$+1$	X	0
$+1$	Z	0
-1	X	1
-1	Z	1

If there has been no eavesdropping then Alice and Bob will agree over the classical bits on the *matching points*. Any spin sent by Alice will be measured with $1/2$ probability in the correct basis by Bob hence *each spin on an average consists of* $1/2$ *bit of information*.
Eve can measures the qubit during transmission in a given basis and then send the spin to Bob. Alice and Bob can then detect Eve with high probability.

Alice and Bob detect Eve by releasing a sufficiently large set of data from the *matching points*. These bits must agree at all points. If there has been some eavesdropping then, there will be some disagreement in this data set. Suppose one of the matching point bits 0, is sent by Alice in the X basis but is measured by Eve in the Z basis. Such a spin would on Bob's end be decoded as 0 or 1 with $1/2$ probability each. It is possible that it is interpreted as 1 hence creating a disagreement between Alice and Bob on a value they must agree on. We can calculate the probability of such disagreement as follows.

For a disagreement to be produced Alice and Bob must use the same basis for their tasks and Eve must measure in a different basis from their common basis. If on an average Eve gets hold of $b(b \leq \frac{1}{2})$ bits of information per spin then with probability b Eve measures in the same basis as Alice and Bob. Hence with probability $1 - b$ Eve's basis doesn't match with common basis of Alice and Bob. If Eve's basis does not match with Alice's then with probability $1/2$ Bob will measure the incorrect state. E.g. Alice and Bob both choose the Z basis while Eve measures in X basis. Then Bob's Z measurement will give either $+1$ value

or -1 value each with probability $1/2$. Hence the amount of disagreement produced by Eve will be $\frac{1}{2} \times 1 - b$. For $b \leq \frac{1}{2}$ we have

$$\frac{1-b}{2} \geq \frac{b}{2}$$

Hence with probability at least $\frac{b}{2}$ Eve will produce a disagreement between Alice and Bob.

2.2 EPR Based protocols

In this section we discuss two EPR based protocols one introduced by Artur Ekert[Eke91] and the other introduced by Bennet, Brassard and Mermin[BBM92]. The later is motivated by the former but is presented with the aim to show that EPR based protocols are equivalent to BB84. The basic idea in EPR based protocols is to use EPR pairs as resources to generate random bit strings such that key distribution protocol is secure against certain general attacks.

2.2.1 E91

This protocol was proposed by Artur Ekert in 1991 and hence the name E91. Let us see how this protocol works.

- Assume there is a source that emits singlet state EPR pairs. One part is sent to Alice and the other to Bob, both of whom agree on a system of coordinates.
- Alice performs measurement on her spin randomly along any of the three directions

  $$\vec{a_1} = \hat{x} \qquad \vec{a_2} = \frac{\hat{x}+\hat{y}}{\sqrt{2}} \qquad \vec{a_3} = \hat{y}$$

  and Bob also performs a measurement on his spin randomly along any of the three directions

  $$\vec{b_1} = \frac{\hat{x}+\hat{y}}{\sqrt{2}} \qquad \vec{b_2} = \hat{y} \qquad \vec{b_3} = \frac{\hat{-x}+\hat{y}}{\sqrt{2}}$$

- Alice and Bob then announce their measurement directions in public.
- They discard the spins which they were not able to measure or the ones which only one of them could measure. They divide the rest of the spins into two groups

  1. *Group* 1: Where the measurement directions of both Alice and Bob are the same, there are two such pairs $(\vec{a_2}, \vec{b_1})$ and $(\vec{a_3}, \vec{b_2})$
  2. *Group* 2: Where the measurement directions of both Alice and Bob are the different.

- They release the results of the second group to check eavesdropping.

By using the elements in Group 1 Alice and Bob can generate a key. If Alice and Bob measure along $(\vec{a_2}, \vec{b_1})$ and Alice obtains the eigen value $+1$ then Bob would obtain the eigen value -1 since the state is a singlet state, given by

$$|\psi\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

Alice and Bob associate with the eigen value $+1$ the bit 0 and with the eigen value -1 the bit 1. Bob can then apply a NOT gate on his key. This will ensure that both share the same key. Alternatively Bob can apply an X gate before measuring his spin (this changes the state from $\frac{|01\rangle - |10\rangle}{\sqrt{2}}$ to $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$) and then associate the bit values with the measurement results.
The elements of Group 2 are used to detect eavesdropping. Alice and Bob compute the familiar *CHSH* quantity

$$S = E(\vec{a_1}, \vec{b_1}) - E(\vec{a_1}, \vec{b_3}) + E(\vec{a_3}, \vec{b_1}) + E(\vec{a_3}, \vec{b_3}) \tag{2.1}$$

Assume the eavesdropper can only perform projective measurements. If there has been no eavesdropping then for a singlet state

$$E(\vec{a_i}, \vec{b_j}) = -\vec{a_i}.\vec{b_j}$$

and by substituting the values of $\vec{a_1}$, $\vec{a_3}$, $\vec{b_1}$ and $\vec{b_3}$ in equation (2.1) we get $S = -2\sqrt{2}$. This is a violation of the *CHSH* inequality[CHSH69]. On the other hand if there has been eavesdropping such that the measurement on the spin sent to Alice was along $\vec{n_a}$ while the measurement on the spin sent to Bob was along $\vec{n_b}$. Then the state that reaches Alice and Bob is no longer an entangled state but a mixed state given by the density matrix ρ.

$$\rho = |c_{++}|^2 |n_a+\rangle\langle n_a+| \otimes |n_b+\rangle\langle n_b+| \tag{2.2}$$
$$+ |c_{+-}|^2 |n_a+\rangle\langle n_a+| \otimes |n_b-\rangle\langle n_b-| \tag{2.3}$$
$$+ |c_{-+}|^2 |n_a-\rangle\langle n_a-| \otimes |n_b+\rangle\langle n_b+| \tag{2.4}$$
$$+ |c_{--}|^2 |n_a-\rangle\langle n_a-| \otimes |n_b-\rangle\langle n_b-| \tag{2.5}$$

Where $|c_{--}|^2$ is the probability of obtaining the value -1 for both Alice and Bob's spin. Using this we can calculate

$$E(\vec{a_i}, \vec{b_j}) = Tr\{\rho(\sigma_{\vec{a_i}} \otimes \sigma_{\vec{b_j}})\}$$

by plugging this back into equation 2.1 with the additional understanding that we must integrate over the probability distribution of directions $\vec{n_a}$ and $\vec{n_b}$ chosen by the eavesdropper

we obtain

$$S = \int \rho(\vec{n_a}, \vec{n_b})\{(\vec{a_1}.\vec{n_a})(\vec{b_1}.\vec{n_b}) - (\vec{a_1}.\vec{n_a})(\vec{b_3}.\vec{n_b}) + (\vec{a_3}.\vec{n_a})(\vec{b_1}.\vec{n_b}) + (\vec{a_3}.\vec{n_a})(\vec{b_3}.\vec{n_b})d\vec{n_a}d\vec{n_b}\} \tag{2.6}$$

Where $\rho(\vec{n_a}, \vec{n_b})$ is the probability that the eavesdropper chooses the directions $\vec{n_a}$ and $\vec{n_b}$. By substituting the values of $\vec{a_1}$, $\vec{a_3}$, $\vec{b_1}$ and $\vec{b_3}$ in equation 2.6 we get

$$-\sqrt{2} \leq S \leq \sqrt{2}$$

Hence by calculating the *CHSH* inequality we can find out whether there has been eavesdropping or not. If there is eavesdropping then bound mentioned in equation 2.6 is respected. So S can be used to test the security of a cryptographic protocol. Here the security is provided by fundamental physical laws and not by the difficulty in performing any computational task.

Further in this protocol each spin has $\frac{2}{9}$ probability of being measured in the same basis by both Alice and Bob, hence on an average each spin conveys $\frac{2}{9}$ bits of information.

2.2.2 BB92

This protocol was presented by Bennet, Brassard and Mermin in 1992 [BBM92]. This protocol is similar to the E91 protocol but is motivated differently. It does not rely on the violation of the CHSH inequality for its security.[1] Rather it wants to show that the use of the CHSH inequality is not necessary for EPR based protocols. It is presented so that a connection between the EPR based protocols and the BB84 protocol can be made. The security of BB84 relies on the fact that Alice and Bob can detect an eavesdropper by looking at the disagreement produced by an eavesdropper when she or he tries to access information that is being transmitted. Let us look at the protocol

- A source emits a singlet state, it sends one spin to Albert and the other to Boris. Albert and Boris then measure their spin randomly along $\hat{x}$ or $\hat{z}$.
- They publicly disclose their measurement directions and discard the states which only one of them has measured successfully.
- They further discard the measurement results where Albert and Boris don't measure along the same axis.

[1] The violation of the CHSH inequality has very debatable connotations in Foundations of Quantum Mechanics

The measurement outcomes that are finally left with Albert and Boris are perfectly anti-correlated and can be used to generate a key. They can also be used to test the security of the scheme. We construct the quantity

$$P = E(\hat{x}, \hat{x}) + E(\hat{z}, \hat{z})$$

and observe that $P = -2$ if there has been no eavesdropping. Assume there is an eavesdropper that performs projective measurements on the states before they reach Albert and Boris. The eavesdropper measures using a normalized probability distribution along directions $\vec{n_a}$ and $\vec{n_b}$ respectively. Then we can write

$$P = \int \rho(\vec{n_a}, \vec{n_b})\{(\vec{n_a}.\hat{x})(\vec{n_b}.\hat{x}) + (\vec{n_a}.\hat{z})(\vec{n_b}.\hat{z})d\vec{n_a}d\vec{n_b}\}$$

From here it can be seen the $-1 \leq P \leq 1$. If there has been any eavesdropping then it can be detected by looking at the quantity P. Though in order to compute the value of P one has to sacrifice a few bits of information from the key.
It must be noted here that on an average each spin conveys $\frac{1}{2}$ bits of information since the probability that Boris measures in the same basis as Albert is $\frac{1}{2}$.

2.2.3 Connection between BB-84 and EPR protocols

In this section we look at the connection between the BB84 and EPR based protocols. It can be seen that both the protocols are equivalent as far as a person observing the channel between Alice and Bob is concerned.
If in the BB92 protocol we shift the source of singlet state and place it with Alice then the protocol so generated is equivalent to BB84 as far as a person monitoring the channel between Alice and Bob is concerned. Lets see how this works. Alice generates a spin singlet and measures her spin in the $\hat{x}$ or $\hat{z}$ basis and then sends the other half to Bob. The spin sent to Bob is in any of the eigen states of X or Z i.e. $|x\pm\rangle, |z\pm\rangle$. Further the uniform randomness of Alice in choosing her measurement direction ensures that the spin is being sent in the state ρ where,

$$\begin{aligned}\rho =& \frac{1}{4}|x+\rangle\langle x+| + \frac{1}{4}|x-\rangle\langle x-| \frac{1}{4}|z+\rangle\langle z+| + \frac{1}{4}|z-\rangle\langle z-| \\ =& \frac{1}{2}\mathbb{I}\end{aligned}$$

This state is the same as what is sent by Alice to Bob in BB84. There Alice generated two random string the first being a random bit string and second being a string of X and Z symbols. Alice sent any of the four eigen states $|x\pm\rangle$ and $|z\pm\rangle$ according to the table

Bit value	Basis	Eigen State
0	X	$\|x+\rangle$
0	Z	$\|z+\rangle$
1	X	$\|x-\rangle$
1	Z	$\|z-\rangle$

Since the two strings are uniformly random all eigen states are sent with probability $\frac{1}{4}$ and the resulting state sent is a completely mixed state whose density matrix is $\frac{1}{2}\mathbb{I}$. Hence a person monitoring this channel cannot make out whether an EPR based protocol is being used or BB84 protocol is being used.

In BB92 after receiving his spin Bob then measures the state randomly in the $\hat{x}$ or $\hat{z}$ direction. Alice and Bob publicly announce their orientation, then they discard the unsuccessful measurements and also the ones in which their orientations don't match. The rest of the data is used to interpret a key. Which is the same as what one does in the BB84 protocol. We also note that in this modified BB92 protocol the average number of bits conveyed per spin is $\frac{1}{2}$ which is the same as BB84.
The essential reason in proving this equivalence is that EPR based protocols can be proven to be secure without involving the CHSH inequality. Since the EPR based protocols are the same as BB84 one can check their security just the same way we test the security of BB84 i.e. by checking if there is a disagreement between Alice and Bob on bits they must agree upon.

Chapter 3

Quantum Error Correction

In this chapter we discuss the formalism needed to describe errors. Using that formalism we analyze the Bit Flip channel, the Phase Flip channel and the Depolarizing channel. We then discuss the CSS quantum error correction code. We illustrate how the code is constructed. We then show how the code is used for detecting and correcting errors. The next section discusses how the error detection and correction process can be modeled using Quantum Operations. Using these insights we prove an important theorem in the final section of the chapter.

3.1 Quantum Operations

Physical systems interacts with their environment. This interaction causes a change in the dynamics of the system. We model this interaction with the environment using Quantum Operation formalism.
Assume a system with density matrix ρ interacts with the environment in a pure state with density matrix $|e_0\rangle\langle e_0|$. They together evolve under a unitary dynamics represented by the operator U. Then the combined state of the system would be

$$U(\rho \otimes |e_0\rangle\langle e_0|)U^\dagger \tag{3.1}$$

The final state of the *system* alone is

$$\mathcal{F}(\rho) = Tr_e[U(\rho \otimes |e_0\rangle\langle e_0|)U^\dagger] \tag{3.2}$$

This can be written as follows

$$\mathcal{F}(\rho) = \sum_i \langle e_i|U(\rho \otimes |e_0\rangle\langle e_0|)U^\dagger|e_i\rangle \tag{3.3}$$

$$= \sum_{i,j,k} \langle e_i|U|e_j\rangle\langle e_j|(\rho \otimes |e_0\rangle\langle e_0|)|e_k\rangle\langle e_k|U^\dagger|e_i\rangle \tag{3.4}$$

$$= \sum_{i,j,k} \langle e_i|U|e_j\rangle(\rho \otimes \langle e_j|e_0\rangle\langle e_0|e_k\rangle)\langle e_k|U^\dagger|e_i\rangle \tag{3.5}$$

$$= \sum_{i,j,k} \langle e_i|U|e_j\rangle\rho\delta_{j0}\delta_{k0}\langle e_k|U^\dagger|e_i\rangle \tag{3.6}$$

$$= \sum_{i} \langle e_i|U|e_0\rangle\rho\langle e_0|U^\dagger|e_i\rangle \tag{3.7}$$

Let $F_i \equiv \langle e_i|U|e_0\rangle$ and put this in 3.7 to get

$$\mathcal{F}(\rho) = \sum_i F_i \rho F_i^\dagger \tag{3.8}$$

Equation 3.8 is known as the operator sum representation of Quantum Operation formalism. If $\mathcal{F}$ is trace preserving then $\sum_i F_i F_i^\dagger = \mathbb{I}$ else $\sum_i F_i F_i^\dagger \leq \mathbb{I}$. Each term in the operator sum representation admits a physical interpretation. The term $F_i \rho F_i^\dagger$ indicates the action of a Unitary on ρ with some probability.
If the system is in a pure state then the operator sum form can be seen to be arising from the action of a Unitary on the combined pure state of the system and the environment. If equation 3.8 represents the quantum operation and $\sum_i F_i F_i^\dagger = \mathbb{I}$ then the system-environment unitary has the form

$$U|\psi\rangle|e_0\rangle = \sum_i F_i|\psi\rangle|e_i\rangle \tag{3.9}$$

One can check that the above operator U is norm preserving. If $\sum_i F_i F_i^\dagger \leq \mathbb{I}$ even then, there exists a simple construct for the unitary U which we shall suppress here. Based on 3.9 we can make the following remark.

Remark 3.1. *If the system is initially in a pure state then after the quantum operation the combined system environment state can be written as a pure state.*

3.1.1 Bit Flip Channel

Using the formalism of quantum operations we model bit flip channels. A qubit is kept in an environment such that the Pauli operator X acts with probability $1-p$ and with probability p the operator $\mathbb{I}$ acts. If the initial state of the density matrix was ρ then the final one would

be given by

$$\mathcal{F}(\rho) = (1-p)X\rho X^\dagger + p\mathbb{I}\rho\mathbb{I} \tag{3.10}$$

Any ρ for a given unit vector $r = (r_x, r_y, r_z)$ can be written as

$$\rho = \frac{\mathbb{I} + \vec{r}.\rho}{2} \tag{3.11}$$

which can be rewritten as

$$\rho = \frac{1}{2}\begin{pmatrix} 1+r_z & r_x - ir_y \\ r_x + ir_y & 1-r_z \end{pmatrix} \tag{3.12}$$

If we apply equation 3.10 to 3.12 we get

$$r_z \mapsto (2p-1)r_z \quad r_x \mapsto r_x \quad r_y \mapsto (2p-1)r_y$$

3.1.2 Phase Flip Channel

Using the formalism of quantum operations we model phase flip channels. A qubit is kept in an environment such that the Pauli operator Z acts with probability $1-p$ and with probability p the operator $\mathbb{I}$ acts. If the initial state of the density matrix was ρ then the final one would be given by

$$\mathcal{F}(\rho) = (1-p)Z\rho Z^\dagger + p\mathbb{I}\rho\mathbb{I} \tag{3.13}$$

If we apply equation 3.13 to 3.12 we get

$$r_z \mapsto r_z \quad r_x \mapsto (2p-1)r_x \quad r_y \mapsto (2p-1)r_y$$

3.1.3 Depolarizing Channel

In a depolarizing channel a qubit in state ρ is replaced by a completely mixed state $\mathbb{I}/2$ with probability λ and remains untouched with probability $1-\lambda$. The state coming out of this channel can be written as

$$\mathcal{F}(\rho) = \lambda\frac{\mathbb{I}}{2} + (1-\lambda)\rho \tag{3.14}$$

We rewrite the above equation in the operator sum form by noting that

$$\frac{\mathbb{I}}{2} = \frac{\rho + X\rho X^\dagger + Y\rho Y^\dagger + Z\rho Z^\dagger}{4} \tag{3.15}$$

we put equation 3.15 in 3.14 to get

$$\mathcal{F}(\rho) = (1 - \frac{3}{4}\lambda)\rho + \frac{\lambda}{4}(X\rho X^\dagger + Y\rho Y^\dagger + Z\rho Z^\dagger) \tag{3.16}$$

we replace $\frac{1}{4}\lambda$ by p and put in the above equation to get

$$\mathcal{F}(\rho) = (1-3p)\rho + p(X\rho X^\dagger + Y\rho Y^\dagger + Z\rho Z^\dagger) \tag{3.17}$$

The above equation admits the following physical interpretation: In a depolarizing channel the state is acted upon by each pauli operator with equal probability p and remains unchanged with probability $1-3p$.

3.2 Error Correcting Codes

In order to protect information from errors we encode bit values in more that one qubit. Using this technique along with other sophisticated means we are able to transmit information in the form of qubits. Assume that we are dealing with n qubits that encode some information. We define an *error correcting code as a subspace of the Hilbert space $\mathbb{C}^{2^n}$ which is protected from errors in a small number of qubits so that any such error can be measured and subsequently corrected without disturbing the encoded state.*[BB84]

3.2.1 CSS Codes

Here we shall discuss the CSS codes[Ste96, CS96]. CSS quantum error correcting codes are motivated from classical linear codes mentioned in Appendix B
Suppose C_1 and C_2 are $[n, k_1]$ and $[n, k_2]$ classical linear codes such that $\{0\} \subset C_2 \subset C_1 \subset F_2^n$ and C_1 and C_2^T both correct t errors. Then $CSS(C_1, C_2)$ is an $[n, k_1 - k_2]$ code capable of correcting t qubit errors.

Construction 3.1. *For $x \in C_1$ we define*

$$|x + C_2\rangle \equiv \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x \oplus y\rangle \tag{3.18}$$

where $\oplus$ is summation modulo 2.

Let $x - x' = y' \in C_2$ then

$$\begin{aligned}
|x' + C_2\rangle &= \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x' \oplus y\rangle \\
&= \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x - y' \oplus y)\rangle \quad \text{let} - y' \oplus y \equiv y'' \\
&= \frac{1}{\sqrt{|C_2|}} \sum_{y'' \in C_2} |x' \oplus y''\rangle \\
&= |x + C_2\rangle
\end{aligned}$$

If x, x' belong to the same coset in C_2 i.e. $x - x' = y' \in C_2$ then they define the same code state. So the total number of distinct code states is the number of cosets of C_2 in C_1 which is $|C_1|/|C_2| = 2^{k_1 - k_2}$. We now discuss how the code defined in construction 3.1 can be used to detect and correct errors.

$e_1 \equiv$ n bit string with $1's$ in places where we have a bit flip

$e_2 \equiv$ n bit string with $1's$ in places where we have a phase flip

If $|x + C_2\rangle$ is the original state then the corrupted state is

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y).e_2} |x + y + e_1\rangle \tag{3.19}$$

We now propose two ways of detecting and correcting the errors.

Method 3.1. *Measuring on Ancillary*

In this method we use an ancillary system to record the error syndrome. We then perform measurements on the ancillary system to obtain the syndrome. With this knowledge we correct the error on the original state.

Bit Flip correction

We introduce an ancillary qubit and apply an operator such that

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y).e_2} |x + y + e_1\rangle|0\rangle \mapsto \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y).e_2} |x + y + e_1\rangle|H_1.e_1\rangle \tag{3.20}$$

where H_1 is the parity check matrix for C_1. We measure the ancillary qubit to obtain $|H_1.e_2\rangle$ and then discard it. Since C_1 can correct upto t errors. By knowing $H_1.e_1$ we can get back e_1 using classical linear coding theory as long as $wt(e_1) \leq t$. We correct the state for the error e_1 and retrieve the state

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y).e_2} |x + y\rangle \tag{3.21}$$

Phase Flip correction

We start from the state given in equation 3.26 and we apply a Hadamard to each qubit

$$H^{\otimes n}\frac{1}{\sqrt{|C_2|}}\sum_{y\in C_2}(-1)^{(x+y).e_2}|x+y\rangle = \frac{1}{\sqrt{|C_2|.2^n}}\sum_{y\in C_2}(-1)^{(x+y).e_2}\sum_{z\in\{0,1\}^n}(-1)^{z.(x+y)}|z\rangle$$

$$= \frac{1}{\sqrt{|C_2|.2^n}}\sum_{y\in C_2}\sum_{z\in\{0,1\}^n}(-1)^{(x+y).(e_2+z)}|z\rangle$$

$$z \equiv z' \oplus e_2$$

$$= \frac{1}{\sqrt{|C_2|.2^n}}\sum_{y\in C_2}\sum_{z'\in\{0,1\}^n}(-1)^{(x+y).(z')}|z'\oplus e_2\rangle$$

If $z' \in C_2^T$ then $\sum_{y\in C_2}(-1)^{y.z'} = |C_2|$
If $z' \notin C_2^T$ then $\sum_{y\in C_2}(-1)^{y.z'} = 0$
using these we get

$$H^{\otimes n}\frac{1}{\sqrt{|C_2|}}\sum_{y\in C_2}(-1)^{(x+y).e_2}|x+y\rangle = \frac{1}{\sqrt{|C_2|.2^n}}|C_2|\sum_{z'\in C_2^T}(-1)^{x.z'}|z'\oplus e_2\rangle \tag{3.22}$$

$$= \frac{1}{\sqrt{2^n/|C_2|}}\sum_{z'\in C_2^T}(-1)^{x.z'}|z'\oplus e_2\rangle \tag{3.23}$$

The state in equation 3.23 is like a state affected by bit flip error e_2. We can correct the above state using the same protocol we used to correct a bit flip error to obtain

$$\frac{1}{\sqrt{2^n/|C_2|}}\sum_{z'\in C_2^T}(-1)^{x.z'}|z'\rangle$$

we apply a Hadamard to each qubit in the above state to get back

$$\frac{1}{\sqrt{|C_2|}}\sum_{r\in C_2}|r+x\rangle$$

■

Method 3.2. *Direct measurements* [1]

Here we perform measurements on the code states directly to obtain the error syndromes. We correct for these syndromes and decode the state to get the information.

Bit Flip correction

A more physical way to look at error detection is through measurements made on the corrupted state.

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y).e_2} |x + y + e_1\rangle \tag{3.24}$$

Let $\sigma_{a(k)}$ be the pauli matrix acting on the k^{th} bit, where $a(k) \in \{x, y, z\}$ and let $\sigma_a^{[l]}$ denote the action of the pauli matrices on the n bit string l.

$$\sigma_a^{[l]} = \sigma_{a(1)}^{l_1} \otimes \sigma_{a(2)}^{l_2} \otimes ..... \otimes \sigma_{a(n)}^{l_n} \tag{3.25}$$

the superscript l_i denotes the exponent of $\sigma_{a(i)}$ which takes values from $\{0, 1\}$.
By definition $\sigma^0 = \mathbb{I}$. Let H_1 be the parity check matrix for C_1 and h_i is the i^{th} row of H_1. Suppose the state $|x + C_2\rangle$ is corrupted by the error e_1 then the state becomes $|x + e_1 + C_2\rangle$. In order to detect e_1 let us measure $\sigma_z^{[h_i]}$'s for each row h_i. It is clear that all $\sigma_z^{[h_i]}$'s commute. Eigen value of $\sigma_z^{[h_i]}$ is ± 1. The operator $\sigma_z^{[h_i]}$ has identity in those positions which have a 0 entry in the row h_i and σ_z the in others. The measurement will return the eigen value $+1(-1)$ if the $\sigma_z's$ act on an even(odd) number of $|1\rangle$ entries in the ket $|x + e_1 + C_2\rangle$. The so obtained measurement results can be mapped to the error syndrome as obtained in classical linear coding theory by using

$$+1 \mapsto 0 \quad -1 \mapsto 1$$

Example For the classical code C with parity check matrix

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

[1] mentioned in [BB84]

We measure the error syndrome for a bit string $[1011011]^T$ by calculating $H[1011011]^T$ which gives

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

For the Quantum code we measure the operators corresponding to H given by

$$H = \begin{pmatrix} \mathbb{I} & \mathbb{I} & \mathbb{I} & \sigma_z & \sigma_z & \sigma_z & \sigma_z \\ \mathbb{I} & \sigma_z & \sigma_z & \mathbb{I} & \mathbb{I} & \sigma_z & \sigma_z \\ \sigma_z & \mathbb{I} & \sigma_z & \mathbb{I} & \sigma_z & \mathbb{I} & \sigma_z \end{pmatrix}$$

on the ket $|1011011\rangle$. We obtain the following result, written in matrix form

$$H = \begin{pmatrix} \mathbb{I} & \mathbb{I} & \mathbb{I} & \sigma_z & \sigma_z & \sigma_z & \sigma_z \\ \mathbb{I} & \sigma_z & \sigma_z & \mathbb{I} & \mathbb{I} & \sigma_z & \sigma_z \\ \sigma_z & \mathbb{I} & \sigma_z & \mathbb{I} & \sigma_z & \mathbb{I} & \sigma_z \end{pmatrix} \begin{pmatrix} |1011011\rangle \\ |1011011\rangle \\ |1011011\rangle \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

we use the correspondence

$$+1 \mapsto 0 \quad -1 \mapsto 1$$

to correlate the eigen values with the classical syndrome.
We apply this scheme on the corrupted state given in equation 3.24. We obtain the syndrome and can get back e_1 using classical linear coding theory as long as $wt(e_1) \leq t$. We correct the state for the error e_1 and retrieve the state

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y).e_2} |x+y\rangle \tag{3.26}$$

Phase Flip correction

We have shown above that by applying a Hadamard we convert the phase flip error into a bit flip error. In order to correct for the phase flip we perform the same detection procedure as for bit flip error syndrome detection. We then correct the state and apply a Hadamard to retrieve the original state.
In the measurement based error detection scheme for phase flips we must measure the bit flip error for the Hadamard transformed basis. From equation 3.23 it is clear that we must measure the operator corresponding to H_2, the parity check matrix for C_2^T. The observable

we measure is

$$H^{\otimes n}\sigma_z^{[l']}H^{\otimes n\dagger} = \sigma_x^{[l']}$$

where l' is a row in H_2.
After retrieving the syndrome with the help of the above measurements and linear error correcting theory. We correct the state given in equation 3.26 to

$$\frac{1}{\sqrt{|C_2|}}\sum_{r\in C_2}|r+x\rangle$$

Notice that the bit flip correction and the phase flip correction are decoupled in the above scheme because $[\sigma_x^{[l]}, \sigma_z^{[l']}]$ for each l, l'. This follows from the fact that l and l' satisfy $l.l' = 0$
■

Example [Stean Code]
Let us given an example of a CSS code. A classical $[7, 4, 3]$ code C, defined by the parity check matrix

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

and the generator matrix

$$G = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Let $C_1 = C$ then C_1 is a $[7, 4, 3]$ code. Let $C_2 = C^T$ then C_2 is a $[7, 3, 3]$ code with generator matrix H^T and parity check matrix G^T. We notice that $C_2 \subset C_1$ hence using C_1 and C_2 we form $CSS(C_1, C_2)$ which is a $[7, (4-3)]$ quantum code.
We note that $C_1 = span(G)$ and $C_2 = span(H^T)$. Using this we generate the code words

$$|0\rangle_L = \frac{1}{2\sqrt{2}}\{|0000000\rangle + |0110011\rangle + |1011010\rangle + |1010101\rangle$$
$$+ |1100000\rangle + |0001111\rangle + |1100110\rangle + |0111100\rangle\}$$
$$|1\rangle_L = \frac{1}{2\sqrt{2}}\{|1111111\rangle + |0101010\rangle + |0010110\rangle + |0011001\rangle$$
$$+ |1110000\rangle|0100101\rangle + |1000011\rangle + |1001100\rangle\}$$

We can encode the classical bit 0 in $|0\rangle_L$ and the classical bit 1 in $|1\rangle_L$.
■

3.2.2 Generalized CSS codes

Till now we were discussing the $CSS(C_1, C_2)$ code. Where C_1 is an $[n, k_1, d]$ code and C_2 is an $[n, k_2, d]$ code. Let us generalize the discussion using two n bit string x and z. We write the code state as

$$|v + C_2\rangle \equiv \frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} (-1)^{z.w} |v \oplus x \oplus w\rangle \quad v \in C_1 \tag{3.27}$$

Let $s \equiv (x, z)$ then we call this quantum code Q_s with the understanding that if $x = 0$ and $z = 0$ then we get back the simple $CSS(C_1, C_2)$ code. We notice that if we measure $\sigma_z^{[l]}$(l is a row in H_1) and $\sigma_x^{l'}$ (l' is a row in H_2) on code state 3.27 then we will obtain syndromes corresponding to $H_1 x$ and $H_2 z$ respectively.
If there was a bit flip errors e_1 and a phase flip error e_2 on the code state 3.27. Then our syndrome measurements would be corresponding to $H_1(x + e_1)$ and $H_1(x + e_2)$. We can correct these errors using the same schemes we used earlier with the understanding that our syndrome measurements correspond to $H_1(x + e_1)$ and $H_1(x + e_2)$ and not $H_1(e_1)$ and $H_1(e_2)$. After recovering the error corresponding to a given syndrome using classical linear coding we may subtract x and z to retrieve the e_1 and e_2 respectively.
We measure measurement of σ_z^l, where l is a row in H_1 $\forall$ l and $\sigma_x^{l'}$, where l is a row in H_2 $\forall$ l on a state $|\psi\rangle \in H^{\otimes n}$. If we obtain the eigen values $+1$ for each of the observables we conclude that the $|\psi\rangle$ was $|v + C_2\rangle$ $v \in C_1$ for some v.
Hence the measurement projects the state $|\psi\rangle$ into the subspace spanned by $|v+C_2\rangle,\ v \in C_1$. If we measured the same operators and obtained some syndromes x and z for bit and phase flip respectively, then we may conclude that we have projected the state into a subspace spanned by code states of $Q_s,\ s = (x, z)$.

A mathematical way to look at this is to observe that the state $|\psi\rangle \in H^{\otimes n}$ is defined by 2^n independent complex coefficients λ's. H_1 is a $n - k_1 \times n$ matrix and H_2 is a $k_2 \times n$ matrix. Each observable $\sigma_z^{[l]}(l$ is a row in $H_1)$ commutes with $\sigma_x^{[l']}$ (l' is a row in H_2). Hence the eigen value for each of them gives independent constraints on λ's. After performing the complete measurement the number of independent constraints are $\frac{2^n}{2^{n-k_1}.2^{k_2}} = 2^{k_1-k_2}$. Hence the state $|\psi\rangle$ gets projected into a subspace with dimensions $k_1 - k_2$. All subspaces of the same dimensions are unitary equivalent. Hence the state is projected into a subspace unitary equivalent to that spanned by code states of $Q_s,\ s = (x, z)$. Further this unitary must be $\mathbb{I}$. If we perform measurements on the state $|\psi\rangle \in Q_{x=0,z=0}$ then we would project it in a space of $2^{k_1-k_2}$ dimensions. But since the state $|\psi\rangle$ is a $+1$ eigen state of all the measurement operators it would remain unaffected. So the $2^{k_1-k_2}$ dimensional space must be $Q_{x=0,z=0}$.
This completes the discussion on CSS Codes.

3.3 Correcting Errors

In general the error correction process can be seen as a quantum operation. Let our corrupted system be S whose state is given by a density matrix ρ. During error correction we perform measurements M_j and obtain their syndromes. We then apply a set of unitary gates U_i and correct for the syndrome. Let $\mathcal{B}$ be the quantum operation associated with an error correction process. Then

$$\mathcal{B}(\rho) = \sum_j Pr[\text{outcome } j]\ \frac{U_i M_i \rho M_i^\dagger U_i^\dagger}{tr(U_i M_i \rho M_i^\dagger U_i^\dagger)} \tag{3.28}$$

$$= \sum_j U_i M_i \rho M_i^\dagger U_i^\dagger \tag{3.29}$$

$$\equiv B_i \rho B_i^\dagger \tag{3.30}$$

where 3.30 represents the operator sum representation of $\mathcal{B}$. Equation 3.30 is similar to 3.8, hence we can construct a unitary U that acts on ρ and an ancillary system A. From remark 3.1 a corrupted system S in the state ρ can be described by a pure state $|\phi\rangle$ in the Hilbert space representing both the System and Environment if the state of the system alone before corruption was pure. If the ancillary system starts in the pure state $|a\rangle$ then we can define U such that

$$U|\psi\rangle|a\rangle = \sum_i (B_i|\psi\rangle)|a_i\rangle \tag{3.31}$$

Theorem 3.1. [2] *Suppose Alice sends Bob the state $|\psi\rangle\langle\psi|$ from an error correcting code Q, which supports recovery of t phase flips and t bit flips. The transmission is disturbed by some noise $\mathcal{E}$, so that Bob receives the state $\rho = \mathcal{E}(|\psi\rangle\langle\psi|)$. He corrects it to obtain ρ' . The fidelity F of the recovered state then obeys*

$$F^2 = \langle\psi|\rho'|\psi\rangle \geq tr(\prod_S \rho), \tag{3.32}$$

where $\prod_S$ is the projector onto the subspace spanned by all states which are obtained by flipping t or fewer qubits and t or fewer phases of $|\psi\rangle$.

Proof Let our system S consist of the state $|\psi\rangle$. Let E be the environment. Then the system environment action leading to an error is given by a quantum operation. In light of remark 3.1 the combined system environment state can be written as the pure state $|\psi_{SE}\rangle$. ρ is then the reduced density matrix of this pure state. Let A be the ancillary system that Bob appends for error correction.
$\prod_s$ is the projector onto the subspace from where states can be error-corrected back to $|\psi\rangle$. States lying outside this subspace will not corrected. The correction scheme will map them to a wrong state. We can divide the states in two.

$$|\psi_c\rangle = (\prod_s \otimes I_{ER})|\psi_{SE}\rangle \otimes |a_A\rangle \tag{3.33}$$

$$|\psi_{nc}\rangle = ((\mathbb{I} - \prod_s) \otimes I_{ER})|\psi_{SE}\rangle \otimes |a_A\rangle \tag{3.34}$$

Where the states $|\psi_c\rangle$ are correctable and the states $|\psi_{nc}\rangle$ are not correctable. The error correction process will map the above states such that

$$|\psi_c\rangle \mapsto |\psi'_c\rangle \quad |\psi_{nc}\rangle \mapsto |\psi'_{nc}\rangle$$

where $|\psi'_c\rangle \propto |\psi\rangle \otimes |EA\rangle$ where $|EA\rangle$ is a state in the environment-ancillary subspace. Let the proportionality constant be λ. Then

$$|\psi'_c\rangle = \lambda|\psi\rangle \otimes |EA\rangle \tag{3.35}$$

Since the correction is a Unitary process

$$\langle\psi'_c|\psi'_c\rangle = \langle\psi_c|\psi_c\rangle \tag{3.36}$$

from eq. 3.33 and 3.35 we get

[2] Understood with help from [Por05]

$$|\lambda|^2\langle\psi|\psi\rangle\langle EA|EA\rangle = \langle a_A| \otimes \langle\psi_{SE}|(\prod_s \otimes I_{ER})(\prod_s \otimes I_{ER})|\psi_{SE}\rangle \otimes |a_A\rangle \tag{3.37}$$

since $\prod_s \otimes I_{ER}$ is a projector $(\prod_s \otimes I_{ER})^2 = \prod_s \otimes I_{ER}$, eq 3.37 gives

$$|\lambda|^2 = \langle\psi_{SE}| \prod_s \otimes I_{ER}|\psi_{SE}\rangle \tag{3.38}$$

$$= tr(|\psi_{SE}\rangle\langle\psi_{SE}| \prod_s \otimes I_{ER}) \tag{3.39}$$

$$= tr(\rho \prod_s) \tag{3.40}$$

Now we analyze the fidelity of ρ' with $|\psi\rangle\langle\psi|$.

$$F^2 = tr_S(\rho'|\psi\rangle\langle\psi|) \tag{3.41}$$

$$= tr_S(tr_{EA}(|\psi'_{SEA}\rangle\langle\psi'_{SEA}|)|\psi\rangle\langle\psi|) \tag{3.42}$$

$$= tr_{SEA}((|\psi'_{SEA}\rangle\langle\psi'_{SEA}|)|\psi\rangle\langle\psi| \otimes \mathbb{I}_{ER}) \tag{3.43}$$

which gives

$$F^2 = \langle\psi_c|\psi'_{SEA}\rangle\langle\psi'_{SEA}|\psi\rangle\langle\psi| \otimes \mathbb{I}_{ER}|\psi_c\rangle + \langle\psi_{nc}|\psi'_{SEA}\rangle\langle\psi'_{SEA}|\psi\rangle\langle\psi| \otimes \mathbb{I}_{ER}|\psi_{nc}\rangle \tag{3.44}$$

if we only consider the first term then we get

$$F^2 \geq \langle\psi_c|\psi'_{SEA}\rangle\langle\psi'_{SEA}|\psi\rangle\langle\psi| \otimes \mathbb{I}_{ER}|\psi_c\rangle \tag{3.45}$$

this equation gives two results. If we put $|\psi'_{SEA}\rangle = |\psi_{nc}\rangle$. From orthogonality of $|\psi_{nc}\rangle$ and $|\psi_c\rangle$ we get $F^2 \geq 0$. If we put $|\psi'_{SEA}\rangle = |\psi_c\rangle$ then we get

$$F^2 \geq \langle\psi'_c|\psi\rangle\langle\psi| \otimes \mathbb{I}_{ER}|\psi_c\rangle \tag{3.46}$$

putting equation 3.35 in equation 3.46 we get

$$F^2 \geq |\lambda|^2(\langle\psi| \otimes \langle EA|)|\psi\rangle\langle\psi| \otimes \mathbb{I}_{ER}(|\psi\rangle \otimes |EA\rangle) \tag{3.47}$$

substituting the value of $|\lambda|^2$ from 3.40 we get

$$F^2 \geq tr(\rho \prod_s) \tag{3.48}$$

which completes the proof.

■

Chapter 4

BB84 under noise

In this chapter we analyze the BB-84 protocol under noise by following the discussion in [SP00]. We first show how the modified Lo-Chau protocol works and then prove its security. In specific we show that the Alice and Bob can share a few high fidelity EPR states even under noisy conditions by using the modified Lo-Chau protocol. These high fidelity states prohibit Eve from accessing more that exponentially small information in the event Alice and Bob agree to use the protocol. There is a deep connection between the modified Lo-Chau protocol and the CSS based quantum error correction protocol. We elucidate this connection. Finally it can be seen that by using certain parts of the CSS based quantum error correction protocol we can retrieve a modified version of the BB-84 protocol.

4.1 Lo-Chau protocol

Let us first define the four bell states

$$|\psi^{\pm}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$
$$|\phi^{\pm}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

C_1 and C_2 are $[n, k_1]$ and $[n, k_2]$ classical linear codes such that $\{0\} \subset C_2 \subset C_1 \subset F_2^n$ and C_1 and C_2^T both correct t errors. Then $CSS(C_1, C_2)$ is an $[n, k_1 - k_2]$ code capable of correcting t qubit errors. H_1 is the parity check matrix for C_1 and H_2 is the parity check matrix for C_2^T

Protocol 4.1.

1. Alice creates $2n$ EPR pairs in the state $|(\phi^+)^{\otimes 2n}\rangle$.

2. Alice selects a random $2n$ bit string b, and performs Hadamard transform on the second half of each EPR pair for which $b = 1$.

3. Alice sends the second half of each EPR pair to Bob.

4. Bob receives the qubits and publicly announces this fact.

5. Alice selects n of the $2n$ encoded EPR pairs to serve as check bits to test for Eve's interference.

6. Alice announces the bit string b and which n EPR pairs are to be check bits.

7. Bob performs Hadamards on the qubits where b is 1.

8. Alice and Bob each measure their half of the n check EPR pairs in the $|0\rangle$, $|1\rangle$ basis and share the results. It too many of these measurements disagree then they abort the protocol.

9. Alice and Bob perform entanglement purification as follows, they make the measurements on their code qubits of $\sigma_z^{[r]}$ for each row $r \in H_1$ and $\sigma_x^{[r']}$ for each row $r' \in H_2$. Alice will obtain the bit and phase syndromes x and z while Bob will obtain the bit and phase syndromes x' and z'. Assuming Alice's syndrome are considered to define the CSS code Q_s where $s = (x, z)$. Alice and Bob share the results, Bob computes his syndrome with respect to Q_s and then transform his state so as to obtain $m = k_1 - k_2$ nearly perfect EPR pairs.

10. Alice and Bob measure the EPR pairs in the $|0\rangle$, $|1\rangle$ basis to obtain the shared secret key.

Working

To see that this protocol is secure we must establish the fact that if Alice and Bob agree to use the protocol then any Eve can get at most exponentially small amount of information. This means that Eve gets virtually no information if Alice and Bob agree to use the protocol. First let us look at Eve's position. She cannot distinguish between check qubits and code qubits while monitoring the channel. The information disclosing the position of the check bit and the code bit is revealed only after Bob confirms their receipt. Hence, Eve must treat all the qubits on an equal footing.

Let us look at the measurements on the **check bits**. Alice and Bob each apply a Hadamard on their qubit before measuring in the $|0\rangle$, $|1\rangle$ basis. This means they effectively measure in the $|+\rangle$, $|-\rangle$ basis. In order to detect an error on the state $|\phi^+\rangle$ it is sufficient to measure the bit flip and the phase flip error. A general error can be described in terms of the bit flip and phase flip error. We notice $|\phi^+\rangle$ maps to $|\psi^+\rangle$ and $|\psi^-\rangle$ under a bit flip error upto an

overall phase. Further $|\phi^+\rangle$ maps to $|\phi^+\rangle$ and $|\psi^-\rangle$ under a phase flip error upto an overall phase. Hence it is sufficient to measure the projectors

$$P_1 = |\psi^+\rangle\langle\psi^+| + |\psi^-\rangle\langle\psi^-|$$
$$P_2 = \mathbb{I} - P_1$$

and

$$P_1' = |\phi^+\rangle\langle\phi^+| + |\psi^-\rangle\langle\psi^-|$$
$$P_2' = \mathbb{I} - P_1'$$

This means that we have to only measure the states in the Bell basis. If we measure a state in a fixed basis only, then the probability distribution so generated is purely classical. For example Z measurements on a density matrix ρ giving up states with half probability and down states with half probability can we written as

$$\rho = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|0\rangle\langle 0|$$

which is a purely classical mixture and its probability distribution will also be classical. Quantum probability distributions can be very different from classical ones, they violate Bell's Inequality![CHSH69] But such violations can only be done using different basis for measurement. Given that Alice and Bob will essentially use only a single basis for the code qubits, their probability distribution will be classical. From this we can say that, *the number of bit flip (phase flip) errors in the check qubits is different from the number of bit flip (phase flip) errors in the code qubits with exponentially small probability*. This fact is formally stated and proven in theorem C.1.

It seems that the measurements mentioned above seem non-local. Fortunately we can rewrite the above measurements in terms of the following local projectors

$$P_1 = \frac{\mathbb{I}^{\otimes 2} - Z^{\otimes 2}}{2}$$
$$P_2 = \mathbb{I} - P_1$$

and

$$P_1' = \frac{\mathbb{I}^{\otimes 2} - X^{\otimes 2}}{2}$$
$$P_2' = \mathbb{I} - P_1'$$

We now show how the above protocol is secure

Theorem 4.1. *After Alice and Bob use $[n, k_1 - k_2]$ CSS error correcting code, which can correct up to t bit flips and t phase flips. The state ρ' obtained after step 9 of the modified Lo-Chau protocol has a fidelity F to m EPR pairs such that*

$$F^2 \equiv \langle (\phi^+)^{\otimes k_1 - k_2} | \rho' | (\phi^+)^{\otimes k_1 - k_2} \rangle \geq tr(\prod \rho)$$

Where $\prod$ is the projection onto the space spanned by Bell pair that differ from $|(\phi^+)^{\otimes n}\rangle$ in t or fewer bit and phase flip errors.

Proof Let $m \equiv k_1 - k_2$. Then we may write

$$|(\phi^+)^{\otimes m}\rangle\langle(\phi^+)^{\otimes m}| = \frac{1}{2^m} \sum_{i,j=0}^{2^m-1} |i\rangle\langle j| \otimes |i\rangle\langle j| \tag{4.1}$$

Since the state is being sent from Alice to Bob, the noise only acts at Bob's end. The state after the action of noise can be written as

$$\rho = \frac{1}{2^n} \sum_{i,j=0}^{2^n-1} |i\rangle\langle j| \otimes \mathcal{F}(|i\rangle\langle j|) \tag{4.2}$$

Where $\mathcal{F}$ represents the action of noise on Bob's part.
They use the CSS code constructed using the linear codes C_1 with parity check matrix H_1 and C_2^T with parity check matrix H_2. Alice measure $\sigma_z^{[l]}$ for each $l \in H_1$ and $\sigma_x^{[l']}$ for each $l' \in H_2$. She obtains eigen values from which she constructs syndromes. From these she extracts x and z such that $H_1 x$ is the bit flip syndrome and $H_2 z$ is the phase flip syndrome. As discussed in section 3.2.2, these measurement projects the state onto Q_s (where $s = (x, z)$) which is spanned by

$$|q_i\rangle \equiv |v_i + C_2\rangle = \sum_{w \in C_2} \frac{1}{|C_2|^{1/2}} (-1)^{v_i . z} |w + v_i + x\rangle \tag{4.3}$$

where $v_i \in C_1$.
The projector onto the space is given by

$$P_s = \sum_{i=1}^{2^m-1} |q_i\rangle\langle q_i|$$

We note that

$$\sum_s P_s = \mathbb{I}$$

This fact follows by noting that the total possible syndromes are decided by H_1 an $n - k_1 \times n$ matrix and H_2 an $k_2 \times n$ matrix. Hence the total possible syndromes are $2^{n-k_1} \times 2^{k_2}$ in

number, each defining distinct subspace Q_s. Hence

$$\begin{aligned}\sum_{s}^{2^{n-(k_1-k_2)}} P_s &= \sum_{s}^{2^{n-m}} \sum_{i=1}^{2^m-1} |q_i\rangle\langle q_i| \\ &= \sum_{i=1}^{2^n-1} |q_i\rangle\langle q_i| \\ &= \mathbb{I}\end{aligned}$$

Let us look at the action of the projection operator on Alice's part of the state. Bob's state is perfectly correlated to Alice's and hence the effect of measurements would be identical on both halves.

$$\sum_{i,j} |i\rangle\langle j| \mapsto \sum_{i,j} P_s|i\rangle\langle j|P_s^\dagger \tag{4.4}$$

$$\propto \sum_{q,q'} |q\rangle\langle q'| \tag{4.5}$$

Equation 4.5 can be normalized to obtain the final state. Putting 4.5 into 4.2 and normalizing we obtain the combined Alice and Bob state for the given error s

$$\rho_s'' = \sum_{q,q'} \frac{1}{2^m} |q\rangle\langle q'| \otimes \mathcal{F}(|q\rangle\langle q'|) \tag{4.6}$$

Equation 4.6 admits the physical interpretation that a state $\sum_{q\in Q_s} \frac{1}{2^{m/2}}|q\rangle|q\rangle$ is being sent through a noisy channel where the noise only acts on the second half. Which means that Alice could have performed the syndrome measurements and then sent the state. We also note that

$$\sum_{q\in Q_s} \frac{1}{2^{m/2}}|q\rangle|q\rangle = \sum_{i=0}^{2^m-1} \frac{1}{2^{m/2}}|i\rangle|i\rangle \tag{4.7}$$

under a Unitary rotation. From equation 4.1 we infer that the state ρ mentioned in equation 4.2 after measurements has become the encoded $|(\phi^+)^{\otimes m}\rangle$ state. Hence the state being sent over the channel is essentially $|(\phi^+)^{\otimes m}\rangle\langle(\phi^+)^{\otimes m}|$.
For the measurements being performed by Alice,let the probability of getting a given error s be p_s then the state of the system ρ can be written as

$$\rho = \sum_s p_s \rho_s'' \tag{4.8}$$

Now Bob perform the measurements $\sigma_z^{[r]}$ for each row $r \in H_1$ and $\sigma_x^{[r']}$ for each row $r' \in H_2$. Like Alice he would obtain $s' = (x', z')$. Alice and Bob share s and s'. Bob then

uses the CSS code Q_s to correct his state. He obtains the density matrix ρ'. Using linearity of quantum error correction and the knowledge that the encoded state being corrected is $|(\phi^+)^{\otimes m}\rangle$, Theorem 3.1 gives the result

$$\langle(\phi^+)^{\otimes m}|\rho'|(\phi^+)^{\otimes m}\rangle \geq \sum_s p_s tr(\prod_s \rho''_s) \tag{4.9}$$

where $\prod_s$ is the projector onto the space spanned by all states that differ from $\sum_{q\in Q_s} \frac{1}{2^{m/2}}|q\rangle|q\rangle$ in t or fewer bit flips and t or fewer phase flips. The space defined by $\prod_s$ is a subspace of the space defined by P_s, hence $\prod_s P_s = \prod_s$ using $\rho'' = \frac{1}{p_s}P_s\rho P_s$ we get:

$$\sum_s p_s tr(\prod_s \rho''_s) = \sum_s p_s tr(\prod_s \rho''_s \prod_s) \tag{4.10}$$

$$= \sum_s p_s \frac{1}{p_s} tr(\prod_s P_s\rho P_s \prod_s) \tag{4.11}$$

$$= \sum_s tr(\prod_s \rho \prod_s) \tag{4.12}$$

$$= tr(\prod \rho) \tag{4.13}$$

where $\sum_s \prod_s = \prod$ is the projector on the subspace spanned by all stated that differ from $\frac{1}{2^{n/2}}\sum_{i=1}^{2^n}|i\rangle|i\rangle = |(\phi^+)^{\otimes n}\rangle$ by t or fewer bit flips and t or fewer phase flips.
This completes the proof. ■

From Theorem 4.1 we know that the corrected states has a fidelity to m EPR pairs greater than the probability of having no more than t bit and phase errors, which is exponentially close to 1 by Theorem C.1. Now we show that having a high fidelity implies security.

Lemma 4.1. *If* $F^2 = \langle(\phi^+)^m|\rho|(\phi^+)^m\rangle > 1 - 2^{-2}$ *then* $S(\rho) < \frac{2m+s+1}{ln2}2^{-2} + O(2^{-2s})$

Proof If $\langle(\phi^+)^m|\rho|(\phi^+)^m\rangle > 1 - 2^{-2}$ then the largest eigen value of ρ must be larger than $1 - 2^{-2}$. $S(\rho) = -tr(\rho\ log_2(\rho))$. Let ρ_{max} be the diagonal density matrix $\{1 - 2^{-2}, \frac{2^{-s}}{2^{2m}-1}, ..., \frac{2^{-s}}{2^{2m}-1}\}$. Such that

$$tr(\rho_{max}) = 1 - 2^{-2} + (2^{2m} - 1)\frac{2^{-s}}{2^{2m} - 1} \tag{4.14}$$

$$= 1 \tag{4.15}$$

Then ρ_{max} is a density matrix as close as possible to $\mathbb{I}/2m$ with the restriction that largest eigen value of ρ_{max} is $1 - 2^{-s}$. Consequently $S(\rho) \leq S(\rho_{max})$ which gives

$$S(\rho) \leq (2^{-s} - 1)log_2(1 - 2^{-s}) - 2^{-s}log_2(2^{-s}) + 2^{-s}log_2(2^{2m} - 1) \tag{4.16}$$
$$\leq (2^{-s} - 1)(-2^{-s}) + s2^{-s} + 2^{-s}log_2(2^{2m} - 1) \tag{4.17}$$
$$\leq 2^{-s}(1 + s + log_2(2^{2m} - 1)) - 2^{-2s} \tag{4.18}$$
$$\leq 2^{-s}(2m + s + 1) + O(2^{-2s}) \tag{4.19}$$

■

From Holevo's bound [Hol73] we know that the maximum information accessible to Eve is upper bound by $S(\rho)$. Hence Eve will get exponentially small amount of information about the key if Alice and Bob agree to use the protocol.

4.2 CSS based Protocol

In this section we show how the Modified Lo-Chau protocol is equivalent to a quantum error correcting protocol. We observe that nothing stops Alice from measuring her check bits before sending the EPR states to Bob. We also note from Equation 4.6 that Alice could have performed her syndrome measurements first and then sent the encoded state through the channel.
Measuring the check bits before sending EPR pairs means that Alice randomly chooses from the states $|0\rangle$ and $|1\rangle$. If she measures the syndromes first then, it means that she is sending m halves of the EPR pair, encoded in the CSS code Q_s. Here $s = (x, y)$ $x, y \in F_2^n$ and H_1x and H_2z are the bit and the phase syndromes. In step 10 of the modified Lo-Chau protocol Alice and Bob measure their EPR pairs. The order in which they measure their halves is not relevant. Alice can measure her EPR pairs before she sends them. Which means that Alice chooses a random m bit key k and encodes it using Q_s. We thus have the following equivalent protocol.

Protocol 4.2.

1. Alice creates n random check bits, a random m bit key k and a random $2n$ bit string b.
2. Alice generates $s = (x, z)$ by choosing n-bit string x and z at random .
3. Alice encodes her key $|k\rangle$ using the CSS code Q_s.
4. Alice chooses n positions(out of $2n$) and puts the check bits in these positions and the codes bits in the remaining positions.
5. Alice applies a Hadamard transform to those qubits in those positions where b is 1.

6. Alice sends the resulting state to Bob. Bob acknowledges receipt of the qubits.

7. Alice announced b, the positions of the check bits, the values of the check bits and the strings s.

8. Bob performs Hadamard on the qubits where b is 1.

9. Bob checks whether too many of the check bits have been corrupted, and aborts the protocol if so.

10. Bob decodes the key bits and uses them for the key.

Intuitively it seem that the security of the above scheme stems from the fact that as long as the error rates are acceptable the CSS codes can send states with high fidelity. From the no-cloning principle very little information can leak to Eve for such high fidelity states.

4.3 Modified BB-84

We now show how the CSS based protocol can be turned into a modified version of the original BB-84 protocol. First note that Bob only cares about the bit values that he obtains after decoding the state and not the phases. Hence, he does not need to perform phase correction, consequently Alice does not need to send z. As far as Bob is concerned, he receives a mixed state averaged over z. Let $k' \in C_1$ be the binary vector in Equation 4.3 then Bob receives the state

$$\frac{1}{2^n|C_2|}\sum_z[\sum_{w_1,w_2\in C_2}(-1)^{(w_1+w_2).z}|k'+w_1+x\rangle\langle k'+w_2+x|] \tag{4.20}$$

after some calculation Equation 4.20 can be rewritten as

$$\frac{1}{|C_2|}\sum_{w\in C_2}|k'+w+x\rangle\langle k'+w+x| \tag{4.21}$$

Equation 4.21 is equivalent to a mixture of states $|k'+w+x\rangle$ with w chosen randomly in C_2. So, Alice gives Bob the error correction information x and sends the state $|k'+w+x\rangle$ over the quantum channel. Over many iterations of the protocol, these are random variables chosen uniformly in F_2^n with the constraint that their difference $k'+w \in C_1$. Bob receives the state $k'+w+x+\epsilon$ where ϵ is the bit flip error. Bob subtracts x and corrects for the error ϵ to get a code word in C_1, which is almost certain to be $k'+w$. Since $k' \in C_1,\ w \in C_2$ and $C_2 \subset C_1$ we have $k'+w \in C_1$. Alice knows k' but not w. In order to share the same key Alice and Bob calculate $k'+w+C_2$ i.e. the coset of C_2 in C_1 and use this as the key. For a given k' both will get the same coset. Arriving at a given coset is like arriving at a given m bit string. Using an m bit string we can generate 2^m distinct keys. Similarly, the

total number of cosets of C_2 in C_1 are $2^{|C_1|-|C_1|} = 2^m$. A given coset is then like one of the 2^m possibilities.
Let us recast the above arguments in a different language to give the modified BB84 protocol. Let $v \equiv k' + x + w$, $u \equiv k' + w$ such that $u + v = x$. Then Alice sends the state $|v\rangle$ to Bob, with error correction information $u + v$. These are two random string in F_2^n with the constraint that $u \in C_1$. Bob obtains $v + \epsilon$, subtracts $u + v$ and corrects the result to a code word in C_1, which with high probability is u. The key is then the coset $u + c_2$.

Protocol 4.3.

1. Alice create $(4 + \delta)n$ random bits.
2. Alice chooses a $(4 + \delta)n$ string b. For each bit she creates a state in the $|0\rangle$,$|1\rangle$ basis (if the corresponding bit of b is 0) or the $|+\rangle$,$|-\rangle$ basis (if the corresponding bit of b is 1).
3. Alice sends the resulting qubits to Bob.
4. Bob received the $(4 + \delta)n$ qubits, measuring each in the $|0\rangle$,$|1\rangle$ or $|+\rangle$,$|-\rangle$ basis at random.
5. Alice announced b.
6. Bob discards any result where he measured in a different basis than Alice prepared. With high probability there are at least $2n$ bits left, if not they abort the protocol Alice decides randomly on a set of $2n$ bits to use for the protocol, and chooses at random n of these to be check bits.
7. Alice and Bob announce the values of their check bits. If too many of these disagree, they abort the protocol.
8. Alice announce $u + v$, where v is the string consisting of the remaining non-check bits and u is a random code word in C_1.
9. Bob subtracts $u + v$ from his code qubits, $v + \epsilon$ and corrects the result, $u + \epsilon$ to a code word in C_1
10. Alice and Bob use the coset of $u + C_2$ as the key.

This completes the discussion on BB-84 protocol under noise.

Chapter 5

Quantum Private Communication

Protection of private information during public comparison of information is the main aim in Quantum private communication(QPC). In general if Alice and Bob have information M_A and M_B respectively then Alice and Bob want to publicly calculate $f(M_A, M_B)$ without letting each other know anything more about their information but what can be inferred from the value of $f(M_A, M_B)$. Where

$$f(M_A, M_B) = \begin{cases} 0 & \text{if } M_A = M_B \\ 1 & \text{if } M_A \neq M_B \end{cases}$$

Lo[Lo97] pointed out that any function $f(M_A, M_B)$ of the above form cannot be computed securely by two parties alone. Hence a third party is needed to facilitate the process. One might think that by using a third party the problem is trivial. Both Alice and Bob can convey their information to a trusted third party and she can tell Alice and Bob the outcome of the function f. The problem here is a little different, Alice and Bob don't wish to disclose their information to anyone. So the third party in question should not know M_A or M_B. Let us assume that the third party is semi-trusted. Which means

1. The Third Part(TP) will follow all the procedures of the protocol. She may try to steal information but cannot get corrupted by an eavesdropper.

2. TP may know the position of different bit values in the compared information, but not the actual bit value of the information.

The three party QPC must have the following conditions

- $f(M_A, M_B)$ is public.
- Alice and Bob don't know M_B and M_A respectively
- The third party does not know M_A or M_B

In general it is better to compare several bits of information at once. In this report we looked at two protocols [WYBW12] and [TLH12]. We analyze the working of [WYBW12]. We then analyze [TLH12] and present an error analysis of the protocol. We analyze the robustness of [TLH12] against depolarizing noise. We then propose a protocol that is robust under noisy channels.

5.1 Bell State swapping Protocol

Recently a protocol based on Bell state swapping had been proposed in [WYBW12]. We show how this works. First we illustrate Bell state swapping. Then we will review the protocol. Finally in the analysis we will show how the protocol works.
In order to see Bell state swapping, assume there are two couples $(1,2)$ and $(3,4)$ each sharing a Bell state. If we measure the states of $(1,3)$ and $(2,4)$ in the Bell basis then we get results according to the decomposition written below

$$|\phi^+\rangle_{12} \otimes |\phi^+\rangle_{34} = \frac{1}{2}(|\phi^+\rangle_{13} \otimes |\phi^+\rangle_{24} + |\phi^-\rangle_{13} \otimes |\phi^-\rangle_{24} + |\psi^+\rangle_{13} \otimes |\psi^+\rangle_{24} + |\psi^-\rangle_{13} \otimes |\psi^-\rangle_{24}) \tag{5.1}$$

$$|\phi^-\rangle_{12} \otimes |\phi^+\rangle_{34} = \frac{1}{2}(|\phi^+\rangle_{13} \otimes |\phi^-\rangle_{24} + |\phi^-\rangle_{13} \otimes |\phi^+\rangle_{24} + |\psi^+\rangle_{13} \otimes |\psi^-\rangle_{24} + |\psi^-\rangle_{13} \otimes |\psi^+\rangle_{24}) \tag{5.2}$$

$$|\psi^+\rangle_{12} \otimes |\phi^+\rangle_{34} = \frac{1}{2}(|\phi^+\rangle_{13} \otimes |\psi^+\rangle_{24} + |\phi^-\rangle_{13} \otimes |\psi^-\rangle_{24} + |\psi^+\rangle_{13} \otimes |\phi^+\rangle_{24} + |\psi^-\rangle_{13} \otimes |\phi^-\rangle_{24}) \tag{5.3}$$

$$|\phi^+\rangle_{12} \otimes |\phi^+\rangle_{34} = \frac{1}{2}(|\phi^+\rangle_{13} \otimes |\psi^-\rangle_{24} + |\phi^-\rangle_{13} \otimes |\psi^+\rangle_{24} + |\psi^+\rangle_{13} \otimes |\phi^-\rangle_{24} + |\psi^-\rangle_{13} \otimes |\phi^+\rangle_{24}) \tag{5.4}$$

These equations show how the Bell pairs are swapped when we measure the states of alternate parties in the Bell basis.
We now present the protocol as given in [WYBW12].
Alice and Bob have L bit strings X and Y respectively. For convenience we assume that L is even. Let the semi-trusted third party by Charlie. They all agree on the following encoding scheme

State	Encoding
$\|\phi^+\rangle$	00
$\|\phi^-\rangle$	01
$\|\psi^+\rangle$	10
$\|\psi^+\rangle$	11

Protocol 5.1.

1. Alice and Bob divide their strings into $L/2$ groups. $(G_1^A, G_2^A .....G_{L/2}^A)$ and $(G_1^B, G_2^B .....G_{L/2}^B)$

2. Alice, Bob and Calvin prepare $L/2$ ordered Bell pairs in the state $|\phi^+\rangle$. They generate the sequences S_A, S_B and S_C given by

$$S_A = [(P_1^A(A_1), P_1^A(A_2)), (P_2^A(A_1), P_2^A(A_2)), ..., (P_{L/2}^A(A_1), P_{L/2}^A(A_2))]$$
$$S_B = [(P_1^B(B_1), P_1^B(B_2)), (P_2^B(B_1), P_2^B(B_2)), ..., (P_{L/2}^B(B_1), P_{L/2}^B(B_2))]$$
$$S_C = [(P_1^C(C_1), P_1^C(C_2)), (P_2^C(C_1), P_2^C(C_2)), ..., (P_{L/2}^C(C_1), P_{L/2}^C(C_2))]$$

   where $P_k^A(A_1)$ represents particle 1 of the EPR pair for the k^{th} group of *A*lice. They further separate the first and second particle from each EPR pair to generate

$$S_1^A = [P_1^A(A_1), P_2^A(A_1), ..., (P_{L/2}^A(A_1)]$$
$$S_1^B = [P_1^B(B_1), P_2^B(B_1), ..., (P_{L/2}^B(B_1)]$$
$$S_1^C = [P_1^C(C_1), P_2^C(C_1), ..., (P_{L/2}^C(C_1)]$$

   and S_2^A, S_2^B, S_2^C.

3. Alice and Calvin prepare L' ordered EPR pairs in $|\phi^+\rangle$ state. Just like S_A and S_C they prepare the sequences $T_{A'}$ and $T_{C'}$. They separate the first and the second particles from this sequence to obtain $T_1^{A'}$, $T_1^{C'}$ and $T_2^{A'}$, $T_2^{C'}$. Alice(Calvin) inserts the string $T_1^{A'}(T_1^{C'})$ in $S_1^A(S_1^C)$ according to a random string $I_A(I_C)$ and generate the string $S_1^{A'}(S_1^{C'})$. Using $I_A(I_C)$ Alice(Charlie) further generates $S_2^{A'}(S_2^{C'})$ by using $T_2^{A'}(T_2^{C'})$.

   - Alice and Calvin send the states $S_2^{A'}$ and $S_2^{C'}$ to each other.
   - Alice and Calvin then disclose I_A and I_C publicly.
   - Calvin (Alice) chooses L' photons from the sequence $S_2^{C'}(S_2^{A'})$ according to $I_C(I_A)$.
   - Calvin (Alice) chooses randomly one of the two basis, σ_z or σ_x to make single-particle measurement on the chosen photons.
   - Calvin (Alice) tells Alice (Calvin) which basis he(she) has chosen for each photon and the outcomes of his(her) measurements
   - Alice (Calvin) uses the same measuring basis as Calvin (Alice) to measure photons in the sequence $S_1^{A'}(S_1^{C'})$ according to $I_A(I_C)$ and checks with the results of Calvins (Alices).
   - If there is no eavesdropping then the results of Alice and Calvin should match.

4. Alice (Calvin) discards the measured photons in $S_1^{A'}(S_1^{C'})$ and gets a new sequence $S_1^A(S_1^C)$.

$$S_1^A = [P_1^A(A_1), P_2^A(A_1), ..., (P_{L/2}^A(A_1)]$$
$$S_1^C = [P_1^C(C_1), P_2^C(C_1), ..., (P_{L/2}^C(C_1)]$$

5. Calvin (Alice) discards the measured photons in $S_2^{A'}(S_2^{C'})$ and gets a new sequence $S_2^A(S_2^C)$.

$$S_2^A = [P_1^A(A_2), P_2^A(A_2), ..., (P_{L/2}^A(A_2)]$$
$$S_2^C = [P_1^C(C_2), P_2^C(C_2), ..., (P_{L/2}^C(C_2)]$$

6. For $j = 1, ..., L/2$, Alice performs the two-particle Bell basis measurement on corresponding two particles $(P_j^A(A_1), P_j^C(C_2))$ in S_1^A, S_2^C. Alice denotes her measurement outcome with M_j^A and makes the following correspondence.

M_j^A	R_j^A
$\|\phi^+\rangle$	00
$\|\phi^-\rangle$	01
$\|\psi^+\rangle$	10
$\|\psi^+\rangle$	11

As a result, the corresponding two particles $P_j^C(C_1), P_j^A(A_2)$ in S_1^C, S_2^A owned by Calvin are collapsed into one of the four Bell states $|\phi^\pm\rangle, |\psi^\pm\rangle$. We can write the following table using 5.1

M_j^A	Calvin's state
$\|\phi^+\rangle$	$\|\phi^+\rangle$
$\|\phi^-\rangle$	$\|\phi^-\rangle$
$\|\psi^+\rangle$	$\|\psi^+\rangle$
$\|\psi^+\rangle$	$\|\psi^+\rangle$

We denote the $L/2$ collapsed EPR pair sequence owned by Calvin with

$$[(P_1^{C''}(C_1), P_1^{C''}(C_2)), ...(P_{L/2}^{C''}(C_1), P_{L/2}^{C''}(C_2))]$$

which is called $S_{C''}$ where the subscript indicates the order of the states in the EPR pairs sequence. Calvin takes particle 1 from each state in $S_{C''}$ to form the ordered sequence

$$S_1^{C''} = [(P_1^{C''}(C_1)), ..., (P_{L/2}^{C''}(C_1))]$$

The remaining partner particles in $S_{C''}$ are labeled

$$S_2^{C''} = [(P_1^{C''}(C_2)), ..., (P_{L/2}^{C''}(C_2))]$$

7. Bob and Calvin prepare L' ordered EPR pairs in $|\phi^+\rangle$ state. Just like before they prepare the sequences $T_{B'}$ and $T_{C'''}$. They separate the first and the second particles from this sequence to obtain $T_1^{B'}$, $T_1^{C'''}$ and $T_2^{B'}$, $T_2^{C'''}$. Bob(Calvin) inserts the string $T_1^{B'}(T_1^{C'''})$ in $S_1^B(S_1^{C''})$ according to a random string $I_B(I_{C'})$ and generates the string $S_1^{B'}(S_1^{C'''})$. Using $I_B(I_{C'})$ Alice(Charlie) further generates $S_2^{B'}(S_2^{C'''})$ by using $T_2^{B'}(T_2^{C'''})$.

   - Bob and Calvin send the states $S_2^{B'}$ and $S_2^{C'''}$ to each other.
   - Bob and Calvin then disclose I_B and $I_{C'}$ publicly.
   - Calvin (Bob) chooses L' photons from the sequence $S_2^{C'''}(S_2^{B'})$ according to $I_{C'}(I_B)$.
   - Calvin (Bob) chooses randomly one of the two basis, σ_z or σ_x, to make single-particle measurement on the chosen photons.
   - Calvin (Bob) tells Alice (Bob) which basis he(she) has chosen for each photon and the outcomes of his(her) measurements
   - Bob (Calvin) uses the same measuring basis as Calvin (Bob) to measure photons in the sequence $S_1^{B'}(S_1^{C'''})$ according to $I_B(I_{C'})$ and checks with the results of Calvins (Bobs).
   - If there is no eavesdropping then the results of Alice and Bob should match.

8. Bob (Calvin) discards the measured photons in $S_1^{B'}(S_1^{C'''})$ and gets a new sequence $S_1^B(S_1^{C''})$.

$$S_1^B = [P_1^B(B_1), P_2^B(B_1), ..., (P_{L/2}^B(B_1)]$$
$$S_1^{C''} = [P_1^{C''}(C_1), P_2^{C''}(C_1), ..., (P_{L/2}^{C''}(C_1)]$$

9. Calvin (Bob) discards the measured photons in $S_2^{B'}(S_2^{C''})$ and gets a new sequence $S_2^B(S_2^{C''})$.

$$S_2^B = [P_1^B(B_2), P_2^B(B_2), ..., (P_{L/2}^B(B_2)]$$
$$S_2^{C''} = [P_1^{C''}(C_2), P_2^{C''}(C_2), ..., (P_{L/2}^{C''}(C_2)]$$

10. For $j = 1, ..., L/2$, Alice performs the two-particle Bell basis measurement on corresponding two particles $(P_j^B(B_1), P_j^{C'''}(C_2))$ in $S_1^B, S_2^{C''}$. Bob denoted the outcome of his measurement with M_j^B and makes the following correspondence.

M_j^B	R_j^B
$\|\phi^+\rangle$	00
$\|\phi^-\rangle$	01
$\|\psi^+\rangle$	10
$\|\psi^+\rangle$	11

As a result, the corresponding two particles $P_j^{C''}(C_1), P_j^B(B_2)$ in $S_1^{C''}, S_2^B$ owned by Calvin are collapsed into one of the four Bell states $|\phi^{\pm}\rangle, |\psi^{\pm}\rangle$. We denote the collapsed Bell state of Calvin with M_j^C and draw the following correspondence

M_j^C	R_j^C
$\|\phi^+\rangle$	$00 = (r_j^{C_1} r_j^{C_2})$
$\|\phi^-\rangle$	$01 = (r_j^{C_1} r_j^{C_2})$
$\|\psi^+\rangle$	$10 = (r_j^{C_1} r_j^{C_2})$
$\|\psi^+\rangle$	$11 = (r_j^{C_1} r_j^{C_2})$

11. For $j = 1, ..., L/2$ Alice and Bob calculate $R_j = (R_j^A \oplus G_j^A) \oplus (R_j^B \oplus G_j^B) = (r_j^1, r_j^2)$. Alice and Bob send $R_1, ...R_{L/2}$ to Calvin. Calvin calculates $R = \sum_{j=1}^{L/2}((r_j^1 + r_j^{C_1}) + ((r_j^2 + r_j^{C_2}))$

The conjecture is that if $R = 0$ then $X = Y$ else $X \neq Y$. We illustrate this by an example.

Let us take an explicit example to show the same. Let $G_j^A = G_j^B = 00$ hence we must get $R = 0$. Let $M_j^A = |\phi^-\rangle$ and $M_j^B = |\phi^+\rangle$, this occurs with probability 1/16. Since $M_j^A = |\phi^-\rangle$ from step 6 we realize that Charlie and Bob start with the state $|\phi^-\rangle_{12}|\phi^+\rangle_{34}$ where Charlie's particles are labeled $(1, 2)$ and Bob's particles are labeled $(3, 4)$. Since $M_j^B = |\phi^+\rangle$ Equation 5.1 gives $M_j^C = |\phi^-\rangle$. Let us look at the various calculations in step 11. We have $R_j^A = 01$, $R_j^B = 00$ this gives $R_j(r_j^1 r_j^2) = 01$. From above we see that $R_j^C(r_j^{C_1} r_j^{C_2}) = 01$ which gives $R = (r_j^{C_1} \oplus r_j^1) \oplus (r_j^{C_2} \oplus r_j^2) = 0$! But for $G_j^A = G_j^B$ we must have $R = 0$.

The above is not an isolated example. One can do the study for all states and verify that the protocol works

5.2 EPR Based Protocol

Here we discuss an EPR based protocol[TLH12] to perform QPC.
Alice and Bob can break down their information into several n bit strings and compare each of these strings. Let the broken n bit string with Alice be M_A and that of Bob be M_B. They compare their information with the help of a semi-trusted third party called Charlie. Let Alice, Bob and Charlie be connected by noiseless channels.

1. Charlie prepares a random n bit string C_T. For each value of C_T he prepares a quantum state. If the value is 0 then he prepares one of the states from $|\phi^{\pm}\rangle$. Else he prepares the one of the states from $|\psi^{\pm}\rangle$. The first particles in the states are arranged in a sequence called T_A, whereas the second particles are arranged in a sequence called T_B.

2. Charlie prepares two sets of decoy particles D_A and D_B randomly in the states: $|0\rangle$, $|1\rangle$, $|+\rangle$ and $|-\rangle$. Charlie randomly inserts D_A in T_A (D_B in T_B) to form two new sets S_A and S_B, respectively. S_A and S_B are then sent to Alice and Bob respectively.

3. Alice and Bob store their particles till they receive the complete set S_A and S_B. Upon receipt they signal Charlie to disclose the information regarding the decoys.

4. Alice and Bob measure the decoys in the appropriate basis and consult over a classical channel to check for eavesdroppers. If the errors are above an appropriate level they abort the protocol, else they proceed.

5. Alice and Bob measure the non-decoy particles in the Z basis. They eigen value $+1$ to 0, and -1 to 1 to obtain the strings R_A and R_B respectively.

6. Alice and Bob calculate $C_A = M_A \oplus R_A$ and $C_B = M_B \oplus R_B$ and then cooperate to calculate $C = C_A \oplus C_B$.

7. They return the string C to Charlie who computes $R_c = C \oplus C_T$.

8. If the string R_C has a single non-zero entry then Charlie prints 1. Else the protocol is repeated till the all the sub parts of the string are verified. Once all are verified he prints 0.

If the string R_c has a single non zero entry then we can conclude $M_A \neq M_B$. Let us show how the above protocol works. Once the protocol has been authenticated through steps 1-4 we proceed further. The authentication is just like that of a BB-84 protocol. After completion of step 5 Alice and Bob have obtained the string R_A and R_B. The strings are same at the positions where the state $|\phi^{\pm}\rangle$ was sent and different at the positions where the state $|\psi^{\pm}\rangle$ was sent. From step 1 we conclude that the strings are correlated where $C_T = 0$ and anti-correlated where $C_T = 1$. We may write

$$R_A = R_B \oplus C_T \tag{5.5}$$

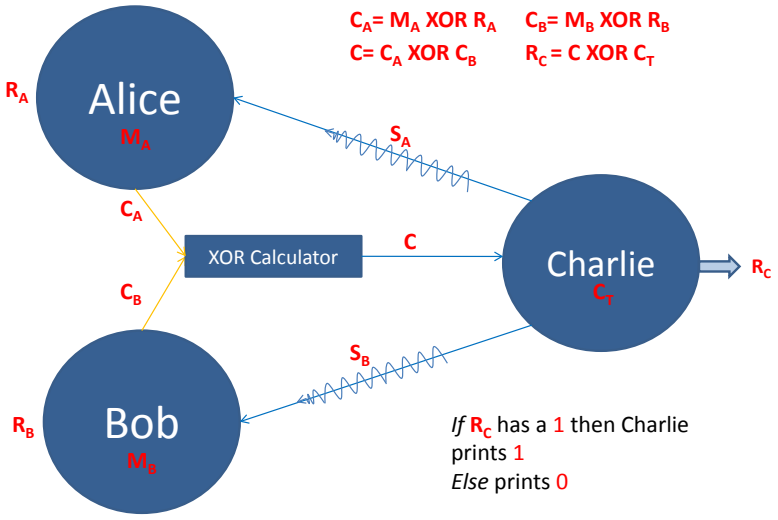

Figure 5.1: QPC Protocol

We look at step 6 and see that

$$\begin{aligned} C &= (M_A \oplus R_A) \oplus (M_B \oplus R_B) \quad \text{from 5.5} \\ C &= (M_B \oplus R_B \oplus C_T) \oplus (M_B \oplus R_B) \\ \implies C &= M_A \oplus M_B \oplus C_T \end{aligned} \tag{5.6}$$

In step 7 we calculate R_C. From 5.6 we get

$$\begin{aligned} R_C &= M_A \oplus M_B \oplus C_T \oplus C_T \\ &= M_A \oplus M_B \end{aligned} \tag{5.7}$$

From equation 5.7 it is clear that the string R_C will have all zeros iff $M_A = M_B$. In which case Charlie will print a 0. If $M_A \neq M_B$ then the string R_C will have 1 at the positions where M_A and M_B differ. In which case Charlie will print a 1. Hence at the end of the protocol Charlie would have made public the outcome of the function

$$f(M_A, M_B) = \begin{cases} 0 & \text{if } M_A = M_B \\ 1 & \text{if } M_A \neq M_B \end{cases}$$

We show how protocol satisfies all the conditions mentioned previously.

Condition 5.2.1. *Alice should not know* M_B

Alice atmost has access to M_A, R_A and $C_B = M_B \oplus R_B$. In order to know M_B Alice must know R_B. The direct way to know R_B is to eavesdrop the line between Charlie and Bob. This is not possible since Alice will get detected in the process[BB84].
We know from equation 5.5 $M_A = M_B \oplus C_T$. Alice can try to find C_T and decipher M_B. Since C_T is known only to Charlie, who won't collude with Alice or Bob or any other eavesdropper the above conditions is always satisfied.
This condition also implies that Bob does not know M_A since the status of Alice and Bob is absolutely equivalent in the protocol.
■

Condition 5.2.2. *Charlie should now know M_A or M_B*

Charlie at most has access to C_T and C. From this information he can atmost know the places at which M_A and M_B differ and nothing more.
■
Hence all the conditions are satisfied. We now analyze the protocol under noise.

5.3 QPC under Noise

5.3.1 Depolarizing Noise

Let both the channels between Alice and Charlie(AC) and between Bob and Charlie(BC) suffer from depolarizing noise. From equation 3.17 we see that under depolarizing noise the channel acts such that each pauli matrix acts on the qubit with equal probability $\frac{\lambda}{4}$. Since both the channels AC and BC are independent the errors act independently. Hence the probability for an XX error is just $\frac{\lambda}{4}.\frac{\lambda}{4}$.
If an error acts such that it takes the state $|\phi^{\pm}\rangle$ to the state $|\psi^{\pm}\rangle$ then the protocol will give an incorrect answer. Also an error that takes the state $|\psi^{\pm}\rangle$ to the state $|\phi^{\pm}\rangle$ will result in an incorrect comparison. Under the action of depolarizing noise these cases occur with probability $\lambda(1 - \frac{\lambda}{2})$.
If we compare two equal k bit strings by breaking them into several smaller strings. Then on the whole at $k.\lambda(1-\frac{\lambda}{2})$ positions the strings would seem unequal. Which means that the protocol will fail to perform a correct comparison. If at a single position a disagreement is produced then the whole calculation would be wrong. The only condition under which the protocol can work, is when over the whole run of the protocol the total error remains less that 1 bit. Which means

$$\begin{aligned} k.\lambda(1 - \frac{\lambda}{2}) &< 1 \\ \implies k &< \frac{1}{\lambda(1 - \frac{\lambda}{2})} \end{aligned} \tag{5.8}$$

Hence for a given λ we must choose an appropriate k as given above. This puts a severe restriction on the length of the strings that can be compared! If the error is 11% then the number of qubits that can be compared is 20. This limits the usage of this protocol.

5.3.2 Bit and Phase Flip Noise

Let us introduce bit flip and phase flip noise in the channels AC and BC. We focus on the AC channel. The bit flip and phase flip act independently. If the probability for bit flip is p and that for phase flip be q. We can model the channel in the following way. For the input density matrix ρ. The action of the bit flip returns the density matrix ρ' given by

$$\rho' = pX\rho X + (1-p)\rho \tag{5.9}$$

On this acts the phase flip error. The final density matrix ρ'' is

$$\begin{aligned}\rho'' &= qZ\rho' Z + (1-q)\rho' \quad \text{using eq 5.9} \\ &= (1-q)pX\rho X + (1-p)qZ\rho Z + (1-q)(1-p)\rho + pqZX\rho XZ \\ &= (1-q)pX\rho X + (1-p)qZ\rho Z + pqY\rho Y + (1-q)(1-p)\rho \end{aligned} \tag{5.10}$$

Equation 5.10 gives the total action of noise on the AC channel. The same modeling works for the BC channel.
Given this kind of noise one can calculate the amount of disagreement that the protocol will show for two equal k bit strings. After some calculation one can show that the disagreement will be in $k.2p(1-p)$ bits. Which is independent of the phase flip probability! So for a given error rate p the protocol can compare k bits of information. Where

$$\begin{aligned} k.2p(1-p) &< 1 \\ \implies k &< \frac{1}{2p(1-p)} \end{aligned}$$

This again puts a serious restriction on the number of bits that can be compared. If the error is 11% then the number of bits that can be compared is upper bounded by 5.

5.4 CSS Code based Protocol

In this section we propose a protocol that is robust under noise. In specific the protocol can work for arbitrary bits k as long as the bit(phase) error rate is under an acceptable rate. Currently the acceptable rate is 11%[SP00].
The basic idea is to use the CSS codes to transfer the key from the third party to Alice and Bob. This protocol of sending secure random bits of information is equivalent to BB84[SP00]. Using the CSS codes it is possible to send a known random classical string.

We use this property to carry out secure Quantum Private Comparison.

5.4.1 Protocol and Working

- Charlie generates a random n bit string, R_A and uses the CSS Code based quantum error correction protocol mentioned as protocol 4.2 to send it to Alice.
- Charlie generates a random n bit string C_T and computes $R_A \oplus C_T$
- Charlie uses protocol 4.2 to send R_B to Bob.
- Alice and Bob take an n bit part of their information M_A and M_B respectively. They compute $C_A = M_A \oplus R_A$ and $C_B = R_B \oplus K_B$.
- Alice and Bob collaborate together to compute $C = C_A \oplus C_B$ and send it to Charlie over a public channel.
- Charlie computes $R_c = C \oplus C_T$.
- If the string R_C has a single non-zero entry then Charlie prints 1. Else the protocol is repeated till the all the sub parts of the string are verified. Once all are verified he prints 0.

The protocol works in exactly the same manner as the one described in section 5.2. The essential difference comes in the distribution of the string R_A and R_B. These strings are conveyed through a BB-84 equivalent scheme. Protocol 4.2 allows the strings to be sent in a secure manner. Further BB-84 is robust against noise. As long as there is tolerable level of noise in the channel the protocol will give correct results for arbitrary size of the string. From literature[SP00] we know that as long as the bit(phase) flip errors are less than 11% the scheme using CSS codes will work.

Appendix A

RSA and Number Theory

Here we present some basic results in Number Theory in order to elucidate the working of RSA protocol. We then present a Python program to implement the protocol.

A.1 Number Theory

Theorem A.1. *GCD for a and b is the least positive integer that can be written in the from $ax + by$ where $x, y \in \mathbb{Z}$.*

Proof Let $s = ax + by$ be the least positive integer in the set $S = \{ax + by | x, y \in \mathbb{Z}\}$ and $gcd(a, b) = k$.
Then $k \mid s$, which means $s \geq k$. If we show that $s \mid k$, would have $k \geq s$ which would mean $s = k$.
We now show that $s \mid k$
Let $s \nmid a$ then

$$\begin{aligned} a &= \lambda s + r \quad r \in [0, s-1] \\ a &= \lambda(ax + by) + r \\ \implies r &= a(1 - \lambda x) - b(\lambda y) \end{aligned}$$

but $0 < r < s$ which contradicts the fact that s is the least positive number of the from $ax + by$. Hence $s \mid a$ and similarly $s \mid b$. Hence $s \mid gcd(a, b)$ in other words $s \mid k$.
■

Corollary A.1. *Let $n \in \mathbb{Z}$, $n > 1$. Then $\exists\ k$ s.t. $ak \equiv 1 (mod\ n)$ iff $gcd(a, n) = 1$*

Proof If a has an inverse in $(\mathbb{Z}/n\mathbb{Z})^*$ then

$$a.a^{-1} \equiv 1(mod\ n)$$
$$a.a^{-1} = 1 + \lambda n \quad \lambda \in \mathbb{Z}$$

which gives

$$a.a^{-1} + (-\lambda)n = 1$$
$$\implies gcd(a, n) = 1$$

Therefore if there exists k such that $ak \equiv 1(mod\ n)$ then $gcd(a, n) = 1$.
The converse of this statement can be shown by tracing back the steps.

Theorem A.2 (Chinese Remainder Theorem)**.** *Suppose $m_1, ...m_k$ are positive integers such that $gcd(m_i, m_j) = 1\ \forall\ i \neq j$. Then the system of equations*

$$x \equiv a_r(mod\ (m_r)) \quad \forall r = 1, ..., k$$

in variable x has a solution. Any two solutions x and x' are congruent modulo $M = \prod_i^k m_i$.

Proof Define $M_i = M/m_i$. Then $gcd(M_i, m_i) = 1$ hence $M_iN_i + m_in_i = 1$. Consequently $M_iN_i \equiv 1(mod\ m_i)$. Since $m_j \mid M_i$ for $i \neq j$ we also have $M_iN_i \equiv 0(mod\ m_j)$ for $i \neq j$. We define $x \equiv \sum_i a_iM_iN_i$. Then using the above relations we see

$$\sum_j M_jN_j \equiv 1(mod(\ m_i))$$
$$\sum_j a_jM_jN_j \equiv a_j(mod(\ m_i))$$

Hence x is a solution to the equations $x = a_r(mod\ (m_r))$. If x and x' are two solutions then

$$x - x' \equiv 0(mod\ (m_i)) \text{ for each } i$$

since $m_i's$ are mutually coprime,

$$x - x' \equiv 0(mod\ (\prod_i m_i))$$
$$x \equiv x'(mod\ (\prod_i m_i))$$
$$x \equiv x'(mod\ M)$$

■

Lemma A.1. *Given a prime p and an integer $k \in [1, p-1]$, $p \mid \binom{p}{k}$.*

Proof We know $\binom{p}{k} = \frac{p!}{(p-k)!k!}$ hence $\binom{p}{k}(p-k)! = p.(p-1).....(k+1)$.
We observe

$$p \mid p.(p-1).....(k+1)$$
$$\implies p \mid \binom{p}{k}(p-k)! \tag{A.1}$$

But $(p-k)!$ has prime factors less than p hence $p \nmid (p-k)!$ so from (A.1) we infer $p \mid \binom{p}{k}$.
■

Theorem A.3 (Fermat's Little Theorem)**.** *If p is prime and $a \in \mathbb{Z}$ then $a^p \equiv a(mod\ p)$. Further if $p \nmid a$ then $a^{p-1} \equiv 1(mod\ p)$*

Proof We prove the above theorem by induction. Let $a = 1$. Then for any prime p,

$$a^p = 1$$
$$\text{and } 1 \equiv 1(mod\ p)$$

So the result holds for $a = 1$. Let the result hold for all numbers upto a. Then

$$a^p \equiv a(mod\ p)$$

$$(a+1)^p = \sum_{k=0}^{p} \binom{p}{k} a^k$$

and

$$p \mid \binom{p}{k} \text{ for } 1 \leq k \leq p-1 \tag{A.2}$$

Therefore

$$(a+1)^p \equiv \sum_{k=0}^{p} \binom{p}{k} a^k (mod\ p)$$

using (A.2) we get

$$(a+1)^p \equiv 1 + a(mod\ p)$$

From induction first part of the theorem follows.
If $p \nmid a$ then $gcd(a, p) = 1$ so $a^{-1} \in (\mathbb{Z}/p\mathbb{Z})^*$. From above

$$a^p \equiv a(mod\ p)$$
$$a^{-1}a^p \equiv a^{-1}a(mod\ p)$$
$$a^{p-1} \equiv 1(mod\ p)$$

This completes the proof.

■

Definition $\phi(n)$ the *Euler Function* is the number of positive integers co-prime to n and are less than n.
We now calculate an explicit expression for $\phi(n)$.
If $n = p^\alpha$, then the numbers $p, 2p, 3p, ..., p^{\alpha-1}.p$ divide n and no other.The total number of positive integers less that p^α are $p^\alpha - 1$ and the total number of positive integers that divide p^α but are less than p^α are $p^{\alpha-1} - 1$. Hence,

$$\begin{aligned}\phi(n) &= (p^\alpha - 1) - (p^{\alpha-1} - 1)\\ &= p^{\alpha-1}(p-1)\end{aligned}$$

We show $\phi(ab) = \phi(a)\phi(b)$ if $gcd(a, b) = 1$. In order to see this we look for all x's s.t $1 \leq x \leq ab$ and $gcd(x, ab) = 1$. Given x_a and x_b such that $gcd(x_a, a) = 1$ and $gcd(x_b, b) = 1$, by *Chinese Remainder Theorem* we can construct x such that

$$\begin{aligned} & x \equiv x_a(mod\ a)\\ \implies\ & x = x_a + \lambda a\\ & x \equiv x_b(mod\ b)\\ \implies\ & x = x_b + \lambda b\end{aligned}$$

The solution for x has a one to one correspondence with (x_a, x_b) for $1 \leq x \leq ab$. The conditions are such that we have forced $gcd(x, ab) = 1$ by forcing $gcd(x_a, a) = 1$, $gcd(x_b, b) = 1$ and $gcd(a, b) = 1$. Hence the number of solutions x are just the number of pairs (x_a, x_b), which by construction is $\phi(a)\phi(b)$. This proves the above assertion.

We can write any $n = \prod_i p_i^{q_i}$. Hence

$$\begin{aligned}\phi(n) &= \phi(\prod_i p_i^{q_i}) \\ &= \prod_i \phi(p_i^{q_i}) \\ &= \prod_i p_i^{q_i-1}(p_i - 1)\end{aligned}$$

Theorem A.4 (Euler's Theorem)**.** *If* $gcd(a, n) = 1$ *then* $a^{\phi(n)} \equiv 1(mod\ n)$

Proof To prove $a^{\phi(n)} \equiv 1(mod\ n)$ it suffices to prove $a^{\phi(n)-1}.a \equiv 1(mod\ n)$. Let us start from $n = p^\alpha$ where p is a prime.
For $\alpha = 1$ then we have shown in A.3 that $a^{p-1} \equiv 1(mod\ p)$. Let the theorem hold for all positive integers upto α. Then $a^{\phi(p^\alpha)} = 1 + k.p^\alpha$ for some $k \in \mathbb{Z}$. For the case $\alpha + 1$ we see that

$$\begin{aligned}a^{\phi(p^{\alpha+1})} &= a^{p^\alpha(p-1)} \\ &= a^{\phi(p^\alpha).p} \\ &= (1 + k.p^\alpha)^p \\ &= \sum_{r=0}^{p} \binom{p}{r} (kp^\alpha)^r \\ &= 1 + \sum_{r=1}^{p} \binom{p}{r} (kp^\alpha)^r\end{aligned}$$

$p \mid \binom{p}{r}$ for $0 < r < p$ and $p^\alpha \mid (kp^\alpha)^r$ for $r \geq 1$. So $p^{\alpha+1} \mid \binom{p}{r}(kp^\alpha)^r$, hence

$$a^{\phi(p^{\alpha+1})} \equiv 1(mod\ p^{\alpha+1})$$

This proves the theorem for the case $n = p^\alpha$.
We now take the general case where $n = \prod_i p_i^{\alpha_i}$ where p_i are primes. Here $\phi(n) = \prod_i \phi(p_i^{\alpha_i})$ and we look at the solution for the set of equations

$$x \equiv 1(mod\ p_i^{\alpha_i})\ \forall i \tag{A.3}$$

we just showed that

$$a^{\phi(p_i^{\alpha_i})} \equiv 1(mod\ p_i^{\alpha_i})\ \forall i$$

from here it is clear that

$$a^{\prod_j \phi(p_j^{\alpha_j})} \equiv 1(mod\ p_i^{\alpha_i})\ \forall i$$

which gives that

$$a^{\phi(n)} \equiv 1(mod\ p_i^{\alpha_i})\ \forall i$$

Therefore $x = a^{\phi(n)}$ is a solution to equation A.3. Now, if we look at the construction of the solution to equation A.3 from Theorem A.2 we see that

$$x = \sum_i M_i N_i \text{ where } M_i = \prod_j p_j^{\alpha_j}/p_i^{\alpha_i},\ M_i N_i \equiv 1(mod\ p_i^{\alpha_i})$$

given this, since x satisfies equation A.3 and $gcd(p_j^{\alpha_j}, p_i^{\alpha_i})\ \forall\ i, j$ one concludes that x must satisfy

$$x \equiv 1(mod\ \prod_i p_i^{\alpha_i})$$
$$x \equiv 1(mod\ n)$$

This along with the fact that $x = a^{\phi(n)}$ completes the proof. ■

A.2 RSA Protocol

Here we explain certain fine points of the RSA protocol. For clarity we mention the protocol here again. Let Alice and Bob be the two parties where Bob distributes the public key.

1. Bob chooses two large primes p, q.
2. Let $n = pq$ then $\phi(n) = (p-1).(q-1)$.
3. Find e such that $gcd(e, \phi(n)) = 1$
4. Find a d such that $ed \equiv 1(mod\ \phi(n))$
5. $P = (e, n)$ is the *public* encryption key and $S = (d, n)$ is the *secret* decryption key.

Where the encryption decryption protocol is given by

$$\mathcal{E}(x) = x^e(mod\ (n)) \text{ and } \mathcal{D}(\mathcal{E}(x)) = \mathcal{E}(x)^d(mod\ n)$$

We first look at point 3 and 4 and from Theorem A.4 conlcude that we can find a number d such that $ed \equiv 1(mod\ \phi(n))$.
We need to show

$$\mathcal{D}(\mathcal{E}(x)) = \mathcal{E}(x)^d \equiv x(mod\ n)$$

where

$$\mathcal{D}(\mathcal{E}(x)) = x^{ed}(mod\ n) \tag{A.4}$$

We know from point 4 in the protocol that

$$ed = 1 + \phi(n) \tag{A.5}$$

Let us consider the case where $gcd(x, n) = 1$
Then from equation (A.4) we get

$$x^{1+\phi(n)} = x.x^{\phi(n)}(mod\ n) \tag{A.6}$$

But we know from Theorem A.4 that

$$x^{\phi(n)} \equiv 1(mod\ n) \tag{A.7}$$

we get put (A.7) in eqaution (A.6) to get

$$x.x^{\phi(n)}(mod\ n) \equiv x(mod\ n)$$

Next, we consider the case where $gcd(x, n) \neq 1$. Since $n = pq$ we may consider the sub cases $p \mid x$ but $q \nmid x$ and $p \nmid x$ but $q \mid x$.
But we consider the first case only since the second one is essentially the same as the first.
Since $p \mid x$,

$$x \equiv 0(mod\ p) \tag{A.8}$$

$$\implies x^{\lambda(p-1)(q-1)+1} \equiv 0(mod\ p) \tag{A.9}$$

$$\equiv x(mod\ p) \quad \forall\lambda \in \mathbb{Z}^* \tag{A.10}$$

Since the prime $q \nmid x$ hence $gcd(x, q) = 1$. From Theorem A.3

$$x^{(q-1)} \equiv 1(mod\ q)$$

Hence it is easy to see

$$(x^{q-1})^{\lambda(p-1)} \equiv 1(mod\ q)$$

$$\implies x^{\lambda\phi(n)} \equiv 1(mod\ q)$$

from above we can conclude

$$x^{\lambda\phi(n)+1} \equiv x(mod\ q) \tag{A.11}$$
$$\implies x^{\lambda\phi(n)+1} - x \equiv 0(mod\ q) \tag{A.12}$$
$$x^{\lambda\phi(n)+1} \equiv x(mod\ p) \tag{A.13}$$
$$\implies x^{\lambda\phi(n)+1} - x \equiv 0(mod\ p) \tag{A.14}$$

From the above equations and the fact that $gcd(p, q) = 1$ it follows that

$$x^{\lambda\phi(n)+1} - x \equiv 0(mod\ pq) \tag{A.15}$$

We choose λ such that

$$\lambda\phi(n) + 1 = ed$$

and hence

$$x^{\lambda\phi(n)+1} \equiv x(mod\ pq) \tag{A.16}$$

we put equation A.5 in equation A.16 to get A.4 to complete the proof.

A.3 Implementation of RSA

Here we present a python program that implements the RSA protocol.

```
#Author: Vikesh Siddhu
#The program will implement RSA in polynomial time by
#using two 20 bit pre-fed primes and generate a program
#in the file dec.py that will be able to decrypt the message

def gcd(a,b):               # GCD using Euclids Algo
    while b != 0:           # one numbers in 0 return other number
        temp = b
        b = a%b
        a = temp             # These 3 lines replace (a,b)
    return a              # by (b, a mod b) which is
    # the essence of Euclids algo



def extEuAlg(a, b) :
  """Computes a solution  to a x + b y = gcd(a,b), as well as gcd(a,b) """
  if b == 0 :
```

```
      return 1,0,a   # This will return a vector 1,0,a if b = 0
  else :
      x, y, gcd = extEuAlg(b, a % b)

# We equate the tuple x,y,gcd by a tuple formed
# if we replace (a,b) by (b, a mod b)

      return y, x - y * (a // b),gcd

# At each replacement we return x-> y,
# y -> x-y(quotient(a,b)), gcd -> gcd

#Why does this work? the basic idea is one
#we hit b=0 we will reach x = 1, y = 0
#and gcd = a'from there we build up the
#final x,y in ax+by=gcd(a,b) by observing
#that in each increase from x = 1, y = 0
# and gcd = a' to the next we must
#replace x by y and y by x-y(quotient(a,b))


def modInvEu(a,m) :
"""Computes the modular multiplicative inverse of a modulo m,
  using the extended Euclidean algorithm
  """
  x,y,gcd = extEuAlg(a,m)
  if gcd == 1 :     # The inverse exists iff a,m are coprime
      return x % m  # Euclids algo is s.t. x is the inverse
  else :
      return None
#Why do all this to find inverse mod m and gcd?
#Because this way it takes polynomial time, brute force will take
#exponential time the brainy are better than the brutes!

#The RSA protocol begins here

p = 2**20 - 3           # 20 Bit prime
q = 2**20 - 5           # 20 Bit prime
n = p*q                 # Large number, product of 2 primes
```

```
phi = (p-1)*(q-1)        # Number of numbers coprime to n
e = 3                    # Prelim. e choice
t = gcd(e,phi)
while t > 1:             # finds odd e s.t. gcd(e,phi) = 1
  # since it will halt at t=1
    e = e + 2
    t = gcd(e,phi)
d = modInvEu(e,phi)      # inverse of e mod phi in polynomial time


M = (int(raw_input("Enter an at most 40 bit number in base 10:")))
if M < 2**40:
    print "The public key is (n,e):"
    print n, e
    print "The message you must send is"
    print (M**e)%n

else:
    print "Please rerun and enter a valid number"

#What we do here is we print the secret key and a code to
#decrypt the message in the file decr.py,
fo = open("dec.py","wb+")
fo.write("#Author: Vikesh Siddhu")
fo.write("#The program decrypts a RSA encrypted message")
fo.write("#The subroutine calculates modular exponentiation in poly time\n")
fo.write("def modexp(x,y,n): \n")
fo.write("\t if y == 0: \n")
fo.write("\t \t return 1\n")
fo.write("\t z = modexp(x,y//2,n)\n")
fo.write("\t if y%2 == 0:\n")
fo.write("\t \t return (z**2)%n \n")
fo.write("\t else : \n")
fo.write("\t \t return (x*(z**2))%n \n")
x = "'Enter the encrypted key:'"
fo.write("em = (int(raw_input(")
fo.write(x)
fo.write(")))")
```

```
fo.write("\n");
fo.write("d =");
fo.write(str(d));
fo.write("\n");
fo.write("n =");
fo.write(str(n));
fo.write("\n");
fo.write("m = modexp(em,d,n)");
fo.write("\n");
fo.write("print \t");
fo.write("m")
fo.close()
```

Appendix B

Classical Linear Coding

B.1 Formalism

B.1.1 Generator Matrix Formalism

A linear code C encoding k into an n bit code space is specified by an $n \times k$ generator matrix G over $\mathbb{Z}_2$.

$$G : \{0,1\}^k \mapsto \{0,1\}^n \quad n \geq k \tag{B.1}$$

Since C encodes k bits in n bits, it is called a $[n, k]$ code.

Example [Repetition Code] Let $x \in \{0,1\}^k$ be a column vector.
For $k = 1$ and $n = 3$ we define a coding scheme as follows

$$0 \mapsto 000\ , \quad 1 \mapsto 111 \tag{B.2}$$

G for this scheme is given by

$$G = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}_{3\times 1}$$

Gx is the encoded n bit string. We see $G[0] = 000$ and $G[1] = 111$.
For $k = 2$ and $n = 6$ the coding is as follows

$$00 \mapsto 000000\ , \quad 01 \mapsto 000111$$
$$10 \mapsto 111000\ , \quad 11 \mapsto 111111$$

G for this scheme is given by

$$G = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}_{6\times 2}$$

it is easy to see that $G[00] = [000000]$ and $G[01] = [000111]$ and so on.
The span of the columns of G is defined as S the space of *code words*. For a unique coding we have to demand that the columns of G are linearly independent.
We also notice that for an $[n, k]$ code 2^k bits in $\{0, 1\}^k$ need to be encoded in the space $\{0, 1\}^n$ hence we need to specify $n.2^k$ maps of the kind B.2. But with the help of the generator matrix formalism we need only specify $n \times k$ entries of the G matrix. This reducs the storage space exponentially.

B.1.2 Parity Check Formalism

An equivalent way of defining a linear code is through the parity check matrix. Let g_i represent the i^{th} column of G. We choose $n - k$ columns g_i such that $g_i g_j^T = 0 \; \forall \; i, j$. Put them as rows in a matrix H. Then H is called the *parity check* matrix.
If $x \in S$ where S is the code space of an $[n, k]$ code C, generated by G then $Hx = 0$.
Hence, the kernel of H defines the $[n, k]$ code C.
We why $Hx = 0 \; \forall \; x \in S$. Since the code space S is the span of all columns of G and all rows of H are orthogonal to all columns of G (by construction) any product of the form

$$\begin{pmatrix} r_1 & . & . & . & . & . \\ r_2 & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ r_{n-k} & . & . & . & . & . \end{pmatrix}_{[n-k\times n]} \begin{pmatrix} c_1 \\ c_2 \\ . \\ . \\ . \\ . \\ . \\ . \\ c_n \end{pmatrix}_{n\times 1} = \begin{pmatrix} 0 \\ 0 \\ . \\ . \\ . \\ . \\ . \\ . \\ 0 \end{pmatrix}_{n\times 1}$$

returns the column vector 0. Here r_i is a row in H and $\{c_1, c_2.....c_n\}$ a column c in G.

each entry of the product column is

$$\begin{aligned} &= r_i.c \\ &= g_i^T.c \\ &= 0 \; \forall i \end{aligned}$$

Which means that every code word in S is in the Kernel of H.
In order to show that H defines C we show that the kernel of H consists of only 2^k elements. Hence $Hx = 0$ iff $x \in S$.

$$H : \mathbb{Z}_n \mapsto \mathbb{Z}_{n-k}$$

where $Ha = a',\ a \in \mathbb{Z}_n,\ a' \in \mathbb{Z}_{n-k}$. For each a there is a $b \in Kern(H)$ such that $H(a+b) = a'$. The total number of such b's are at least 2^k in number. Since all rows in H are linearly independent the image of H must have at least $2^n/2^k$ entries. Which is the maximum possible number as the image of H is in $\mathbb{Z}_{n-k}$. Hence the kernel of H can have no more than 2^k entries. Hence $Hx = 0$ iff $x \in S$.
It can be shown with the help of Gaussian elimination and bit flips that H can be rewritten in the form $[A|I_{n-k}]$. Here I_{n-k} is the $n-k \times n-k$ identity matrix and bit flip represents $0 \mapsto 1$.
If we wish to go back from the parity check matrix H to the generator matrix G we apply the following scheme

- Pick k linearly independent vectors $y_1, ..., y_k$ that span the kernel of H.
- Set columns of G to be these vectors $y_1, ..., y_k$.

From the above construction we can rewrite the G matrix in the form $[\frac{I_k}{-A}]$ such that $HG = 0$.

B.2 Error Detection

Let $x \in \{0,1\}^k$ and G be the generator matrix defining the $[n,k]$ code. Then $Gx = y$ is the encoding for x in C. If a bit flip error occurs in the j^{th} bit then it can be represented by

$$e_j = \begin{pmatrix} 0 \\ . \\ . \\ . \\ 1 \\ 0 \\ . \\ . \\ 0 \end{pmatrix}$$

where 1 occurs in the j^{th} position. The encoded string is transformed from y to y' where $y' = y + e$. The error syndrome for the error is defined as Hy' which can be computed as follows

$$Hy' = H(y + e_j)$$
$$Hy' = He_j$$

Now we can compare Hy' with $He_m \ \forall \ m \in [1, n]$ and determine e_j and then rectify the error and return y' to y.

Definition [Distance] For $x, y \in \{0, 1\}^n$ the hamming distance between x and y is number of place at which x differs from y.
Example: $d[(0010), (0101)] = 3$

Definition [Weight] For $x \in \{0, 1\}^n$ the hamming distance between x and 0 is the weight of x.
Example: $w[(0010)] = 1$
It is easy to show that $d(x, y) = w(x + y)$.

B.2.1 Error Correcting

Let x be encoded as $y = Gx$ and an error e changes y to y' where $y' = y + e$. In order to find y from y' we note that the most likely y' from which y can be recovered would differ from y in the minimum number of positions. Hence we wish to minimize $d[y, y']$ in order to arrive at y from y'. In principle we would need 2^k queries to arrive at the minimization for $d[y, y']$.

Definition [Distance of a Code] The minimum distance between any two code words $x, y \ x \neq y$ is defined as the distance of the code.

$$\begin{aligned} d(C) &= min_{x,y \in C} \ d(x, y)_{x \neq y} \\ &= min_{x,y \in C} \ wt(x + y)_{x \neq y} \\ &= min_{x' \in C} \ wt(x') \end{aligned}$$

A $[n, k]$ code C with distance d is written as an $[n, k, d]$ code. A code with distance at least $2t + 1$ for some integer t is able to correct upto t bit errors. If $y' \in (t \text{ distance of } y)$ for some y then we correct y' to y. This is a unique correction since $d(y_i, y_j) \geq 2t + 1 \ \forall \ i, j$.

A code defined by H has the distance d iff $d-1$ columns of H are linearly independent.
If $u \in C$ then $Hu = 0$

$$Hu = \sum_i h_i u_i \quad \text{where } u_i \text{ are the elements of } u \text{ and } h_i \text{ are columns of } H$$

If distance of C is d the there are at least d places in u which are non-zero hence

$$\sum_{i \text{takes } d \text{ values in total}} h_i = 0$$

Hence d columns are linearly dependent.
If $\sum_j h_j = 0$ and j takes $d-1$ values then the hamming distance for some u would be $d-1$, which is contrary to the fact that the distance of the code is d. Hence any set of $d-1$ columns are linearly independent.
If any $d-1$ columns are linearly independent and some set of d columns are linearly dependent then we look at the expression

$$Hu = \sum h_i u_i = 0$$

Here if d columns are linearly dependent then $\{u_i\}_{i=1}^n$ has at least $'d'$ $1's$.
If any $d-1$ columns are linearly independent then $\{u_i\}_{i=1}^n$ has greater than $'d-1'$,$1's$. So the distance of C is d, the minimum number of 1 entries in $\{u_i\}_{i=1}^n$

B.3 Dual Construction

C is an $[n,k]$ code with generator matrix G and parity check matrix H. Then the dual of C, C^T is defined by the generator matrix H^T and parity matrix G^T. C^T is a $[n,n-k]$ code. If the columns of G are g_i and the rows of H are h_j then $g_i.h_j = 0 \; \forall \; i,j$. The columns of H^T are h_j^T, so all columns in C^T are orthogonal to columns in C.
If C is weakly self dual then $C \subseteq C^T$. If C is strictly self dual then $C = C^T$.

Lemma B.1. *A $[n,k]$ code C is weakly self dual iff $G^T G = 0$.*

Proof Let C be weakly self dual then for $x \in C$ we have $x \in C^T$ and for some appropriate y we have

$$Gy = x$$

but G^T is the parity check matrix for C^T and hence

$$G^T x = 0 \tag{B.3}$$

$$G^T(Gy) = 0 \tag{B.4}$$

$$G^T G y = 0 \tag{B.5}$$

but for any $x \in C \; \exists \; y$ henc $G^T G y = 0 \; \forall \; y$. Hence $G^T G = 0$
If $C \subseteq C^T$ then for any $y \in \{0,1\}^k$

$$Gy = x \tag{B.6}$$

but $G^T G y = G^T x = 0$

$$\implies x \in C^T$$

but from equation B.6 $x \in C$. Hence $C \subseteq C^T$.

■

Lemma B.2. *C is a linear code and $x \in C$ then $\sum_{y \in C}(-1)^{x.y} = |C|$ and if $x \notin C^T$ then $\sum_{y \in C}(-1)^{x.y} = 0$*

Proof If $x \in C^T$ then for any $y \in C$ we have $x.y = 0$. Then the sum

$$\begin{aligned} \sum_{y \in C}(-1)^{x.y} &= \sum_{y \in C}(-1)^0 \\ &= \sum_{y \in C} 1 \\ &= |C| \end{aligned}$$

If $x \notin C^T$ then we look at the sum S given by

$$S = \sum_{y \in \{0,1\}^n} (-1)^{x.y} \quad x \in 0, 1^n, \; x \neq 0$$

For any given x the number of which $y's$ satisfying $x.y \equiv 0 (mod\ 2)$ is equal to the number of $y's$ satisfying $x.y \equiv 1 (mod\ 2)$. Hence the above some can be rewritten as

$$S = \sum_{pairs} ((-1) + 1)) \tag{B.7}$$

hence $S = 0$.
If G is the generator matrix for C, then for any $y \in \{0,1\}^k$ $Gy \in C$. Let G^T be the parity

check matrix for C^T then

$$\begin{aligned}
\sum_{Gy \in C} (-1)^{x.Gy} &= \sum_{Gy \in C} (-1)^{x^T Gy} \\
&= \sum_{Gy \in C} (-1)^{(G^T x)^T y} \\
&= \sum_{Gy \in C} (-1)^{(G^T x).y} \\
&= \sum_{Gy \in C} (-1)^{(G^T x).y} \quad \text{we know } Gy \in C \implies y \in \{0,1\}^k \\
&= \sum_{y \in \{0,1\}^k} (-1)^{(G^T x).y} \quad \text{let } G^T x \equiv a \\
&= \sum_{y \in \{0,1\}^k} (-1)^{a.y} \quad \text{using same logic as B.7} \\
&= 0
\end{aligned}$$

This proves the Lemma.

■

Appendix C

Random Sampling Test

Theorem C.1. *For a $2n$ bit string with n tested bits and n untested bits chosen randomly.The probability that there are more that $(\delta+\epsilon)n$ errors in the untested bits and there are less than δn errors in the tested bits is exponentially small in ϵ. Where $0 \leq \delta,\ \epsilon \leq 1$*

Proof Let there be a total of μn errors in the $2n$ bit string, $0 \leq \mu \leq 2$. We notice that if there are δn errors in the test bits then there will be $(\mu-\delta)n$ errors in the untested bits. We wish to show

$$p(error_{test} \leq \delta n,\ error_{untest} \geq (\mu-\delta)n) < exp\{-O(\epsilon^2 n)\} \tag{C.1}$$

Let

$$P \equiv p(error_{test} \leq \delta n,\ error_{untest} \geq (\mu-\delta)n)$$

then

$$p(error_{test} \leq \delta n) < p(error_{test} = \delta n)\delta n \tag{C.2}$$

which gives

$$P < p(error_{test} = \delta n)\delta n \tag{C.3}$$

$$< (\delta n).\frac{\binom{\mu n}{\delta n}\binom{(2-\mu)n}{(1-\delta)n}}{\binom{2n}{n}} \tag{C.4}$$

We now show

$$\frac{1}{an+1}2^{anH(\frac{b}{a})} \leq \binom{an}{bn} \leq 2^{anH(\frac{b}{a})} \tag{C.5}$$

where $H \equiv -(xlogx + (1-x)log(1-x))$, $a > b > 0$
We see that,

$$\frac{1}{an+1}2^{anH(\frac{b}{a})} = \frac{(an)^{an}}{(bn)^{bn}(a-b)^{(a-b)n}}\frac{1}{an+1} \tag{C.6}$$

$$2^{anH(\frac{b}{a})} = \frac{(an)^{an}}{(bn)^{bn}(a-b)^{(a-b)n}} \tag{C.7}$$

Also we note that

$$\sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n} \leq n! \leq en^{n+\frac{1}{2}}e^{-n} \tag{C.8}$$

Using C.8 we can show that

$$\binom{an}{bn} \leq \frac{e}{2\pi}\sqrt{\frac{a}{b(a-b)n}}\frac{(an)^{an}}{(bn)^{bn}(a-b)^{(a-b)n}} \tag{C.9}$$

$$\binom{an}{bn} \geq \frac{\sqrt{2\pi}}{e^2}\sqrt{\frac{a}{b(a-b)n}}\frac{(an)^{an}}{(bn)^{bn}(a-b)^{(a-b)n}} \tag{C.10}$$

In order to show C.5 we must prove

$$\frac{1}{an+1}2^{anH(\frac{b}{a})} \leq \frac{\sqrt{2\pi}}{e^2}\sqrt{\frac{a}{b(a-b)n}}\frac{(an)^{an}}{(bn)^{bn}(a-b)^{(a-b)n}} \tag{C.11}$$

$$2^{anH(\frac{b}{a})} \geq \frac{e}{2\pi}\sqrt{\frac{a}{b(a-b)n}}\frac{(an)^{an}}{(bn)^{bn}(a-b)^{(a-b)n}} \tag{C.12}$$

putting equation C.6 in C.11 we get

$$\frac{1}{an+1} \leq \frac{\sqrt{2\pi}}{e^2}\sqrt{\frac{a}{b(a-b)n}} \tag{C.13}$$

$$\implies 1 \leq \frac{\sqrt{2\pi}}{e^2}\{a\sqrt{n} + \frac{1}{\sqrt{n}}\}\{\frac{1}{\sqrt{(\frac{b}{a})(a-b)}}\} \tag{C.14}$$

equation C.14 is satisfied under the conditions that n is large and $a > b > 0$ where a is not large.
Similarly putting equation C.7 in C.12 we get

$$1 \geq \frac{e}{2\pi}\sqrt{\frac{a}{b(a-b)n}} \tag{C.15}$$

Which is true under the conditions that n is large and $a > b > 0$ where a is not large. Hence we have established equation C.5. We put C.5 in C.4 to get

$$P < (\delta n).\frac{2^{\mu nH(\frac{\delta}{\mu})}.2^{(2-\mu)nH(\frac{1-\delta}{2-\mu})}}{\frac{1}{2n+1}2^{2nH(\frac{1}{2})}} \tag{C.16}$$

A Taylor expansion of $H(x)$ can be used to show that

$$H(x) \leq 1 - \frac{(1-2x)^2}{2} \tag{C.17}$$

we put equation C.17 in equation C.16 and get $P < e^{-O(\epsilon^2 n)}$.

■

Bibliography

[BB84] C.H Bennet and G. Brassard, *Quantum cryptography: Public key distribution and coin tossing*, Proc. International Conference on Computers, Systems and Signal Processing, Bangalore, India (1984), 10–12.

[BBM92] Charles H. Bennett, Gilles Brassard, and N. David Mermin, *Quantum cryptography without bell's theorem*, Phys. Rev. Lett. **68** (1992), 557–559.

[CHSH69] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt, *Proposed experiment to test local hidden-variable theories*, Phys. Rev. Lett. **23** (1969), 880–884.

[CS96] A. R. Calderbank and Peter W. Shor, *Good quantum error-correcting codes exist*, Phys. Rev. A **54** (1996), 1098–1105.

[Eke91] Artur K. Ekert, *Quantum cryptography based on bell's theorem*, Phys. Rev. Lett. **67** (1991), 661–663.

[Hol73] A. S. Holevo, *Statistical problems in quantum physics*, Procedings of the second Japan-USSR synopsium on probability theory **330** (1973), Lecture Notes in Mathematics 104–119.

[Lo97] Hoi-Kwong Lo, *Insecurity of quantum secure computations*, Phys. Rev. A **56** (1997), 1154–1162.

[Por05] Christopher Portmann, *Secure quantum key distribution over noisy quantum channels*, ETH Zurich, Sept 2005.

[SP00] Peter W. Shor and John Preskill, *Simple proof of security of the bb84 quantum key distribution protocol*, Phys. Rev. Lett. **85** (2000), 441–444.

[Ste96] A Steane, *Multiple-particle interference and quantum error correction*, Proc. R. Soc. London A **452** (1996), 2551–2577.

[TLH12] Hsin-Yi Tseng, Jason Lin, and Tzonelih Hwang, *New quantum private comparison protocol using epr pairs*, Quantum Inf Process **11** (2012), 373–384.

[WYBW12] LIU Wen, WANG Yong-Bin, and CUI Wei, *Quantum private comparison protocol based on bell entangled states*, Commun. Theor. Phys **57** (2012), 583588.