

Secure Quantum Key Distribution Over Noisy Quantum Channels

Christopher Portmann

Master Thesis

September 2005

Department of Computer Science
ETH Zurich
Switzerland

Abstract

Classical key distribution is only secure if the adversary is bounded in some way, e.g., computationally. But if Alice and Bob use a quantum channel to communicate they can achieve unconditional security: eavesdropping cannot be performed on a quantum channel without modifying the states being transmitted, so Alice and Bob can detect the noise introduced, and take the necessary steps.

BB84 is one of the simplest and probably most well known QKD protocols. It is a so-called “prepare-and-measure” protocol, which doesn’t need a complete quantum computer to be performed, but can be realized with today’s technology. We present a new prepare-and-measure protocol for quantum key distribution using two-way communication, achieving a tolerable error rate of 16.9%. We combine error detection instead of correction as proposed by Gottesman and Lo [4] and classical privacy amplification, which has been proven secure against a quantum adversary by Renner and König [6, 15]. We prove the security of our protocol using EPR pairs and the result from [6, 15], and then derive our prepare-and-measure version from the EPR one as in the Shor and Preskill security proof of BB84 [16].

Our result is slightly worse than Gottesman and Lo’s, which achieves 18.9% but is much simpler and only makes use of error correcting codes for a very small number of errors, which otherwise could require exponential time, making practical realizations impossible.

Acknowledgements

I would like to thank my thesis supervisor, Louis Salvail, for having given me the opportunity to go to Denmark to write my thesis, and for all his technical help and guidance throughout these 6 months. Many thanks as well to Thomas B. Pedersen for all his help and our interesting discussions. I am also grateful to Christian Schaffner for proof reading and Saurabh Agarwal for his help with L^AT_EX.

Contents

1	Introduction	1
1.1	Quantum key distribution	1
1.2	Thesis overview	2
1.3	Basic quantum information notions	3
1.3.1	Von Neumann entropy	3
1.3.2	Fidelity	3
1.3.3	Trace distance	4
1.3.4	Commuting observables	4
2	Quantum error correcting codes	7
2.1	Quantum noise	7
2.1.1	Trace-preserving quantum operation	7
2.1.2	Non-trace-preserving quantum operations	8
2.1.3	Mocking up a quantum operation	9
2.2	Quantum error correction	10
2.2.1	Error correcting codes	10
2.2.2	Discretisation of the errors	12
2.2.3	Correcting “small” errors	14
2.3	CSS codes	16
2.3.1	Definition	16
2.3.2	Error correction	17
2.4	Stabiliser codes	18
2.4.1	Stabiliser formalism	18
2.4.2	CSS codes as stabiliser codes	21
3	BB84	23
3.1	Modified Lo-Chau	23
3.2	CSS codes	28
3.3	BB84 protocol	30
3.4	QKD error rates	32
3.4.1	Upper bound	32
3.4.2	Lower bound	32

4	Decoupling bit and phase correction	35
4.1	Privacy amplification	35
4.2	Reduction to prepare-and-measure	40
4.3	Maximum error rate	41
5	Error detection	43
5.1	The protocol	43
5.2	Residual bit flips	45
5.3	Residual phase flips	50
5.4	Maximum error rate	53
5.5	Reduction to prepare-and-measure	53
5.5.1	Prepare-and-measure protocol	53
5.5.2	Reduction	55
6	Conclusion	57
6.1	Advantages	57
6.2	Further improvements	57

Chapter 1

Introduction

1.1 Quantum key distribution

Classical key distribution is only secure against a computationally bounded adversary. In principle nothing prevents an eavesdropper with infinite computational power to passively monitor the execution of the key distribution protocol, get some information about the key and decode any encrypted communication using that key. Furthermore development of quantum computers could make some classical protocols insecure, which would have a retroactive effect on all past communication and encryption.

Quantum key distribution on the other hand is unconditionally secure. Passive monitoring is not possible in quantum mechanics. Any attempt by the adversary to gain information about the key will almost always result in disturbance being introduced in the channel. This disturbance can be detected by the legitimate users, corrected if it is small enough, if not the protocol is aborted.

To perform a quantum key distribution protocol, Alice and Bob have a classical and a quantum channel at hand. They perform the initial qubit exchange along the quantum channel, and then compare and discuss the results on the classical channel. The quantum channel is assumed to be under the total control of the adversary, Eve. The classical channel is authentic, but otherwise totally insecure, so Eve has access to all of their classical communication.

The protocols of interest are the so-called “prepare-and-measure” protocols, which do not require a complete quantum computer. In fact the only operations required are to prepare quantum states and perform a projective measurement on them. Alice prepares and sends her qubits to Bob, who measures them as soon as he receives them, without needing quantum memory to stock them or a quantum compute to perform some operation before the measurement.

In this work we concentrate on BB84-like protocols. BB84 [2] is prob-

ably the most well known and simplest prepare-and-measure quantum key distribution protocol. Alice prepares single qubits in one of two possible basis chosen at random. She then sends them to Bob who measures them immediately upon reception in one of the two basis. They now each have a string of bits and can apply classical error correction and privacy amplification by communicating over the classical channel. All BB84-like protocols follow the same principle of sending and measuring single qubits in one of two possible basis. What differs is the classical post-processing applied to the resulting bit strings. We see that the only quantum operations involved in BB84 are preparing states and measuring them upon reception. All the post-processing is done classically, which makes it a prepare-and-measure protocol.

Other prepare-and-measure quantum key distribution protocols exist, e.g., the six-state scheme or B92, but they are less practical to use: the six-state scheme uses one extra basis so needs a more sophisticated physical apparatus and is harder to implement, and B92 is much less resistant to noise, so only works over shorter distances. Thus BB84 is the most interesting protocol to work on and improve for practical implementation purposes.

Any BB84-like protocol has an upper bound on the tolerable error rate of 25%, which is due to a simple intercept and resend eavesdropping strategy (see Section 3.4.1). The Shor and Preskill proof of BB84 [16], which relies on entanglement purification and quantum error correction has a tolerable error rate lower bound of 11%. Several methods have been proposed to improve this lower bound. Gottesman and Lo propose to detect the errors and drop blocks with errors instead of correcting them. They thus achieve a lower bound of 18.9%.

We combine the error detection idea of [4] with a proof of security of privacy amplification against a quantum adversary from [6, 15], and propose a new simpler BB84-like protocol for QKD. Our protocol only tolerates an error rate of 16.9%, but the bound is obtained analytically and we only use error correcting codes for very small errors.

1.2 Thesis overview

After a brief introduction to the basic notions of quantum information given in the last section of this chapter, we will explain how noise can affect quantum transmission and how to correct it in Chapter 2. In that same chapter we will introduce the error correcting codes and stabiliser formalism, which we will use throughout the thesis.

In Chapter 3 we will explain BB84 in detail, and go through the proof of security as proposed by Shor and Preskill [16], which relies on entanglement purification and achieves the 11% lower bound on the tolerable error rate.

In Chapter 4 we will then replace the privacy amplification step derived

from quantum error correction of phase flips which appears in BB84 by classical privacy amplification using two-universal functions. The result from [6, 15] proves that this is secure against a quantum adversary.

Finally in Chapter 5 we will add an error detection step as proposed in [4]. We divide our string in substrings and perform random parity checks. If Alice and Bob's results differ, they drop the substring. This final protocol tolerates an error rate of 16.9%.

1.3 Basic quantum information notions

In this section we will give a few definitions and properties of some quantum information theoretic concepts, which we will need throughout the thesis. We refer to e.g., [13], for proofs and more formal derivations of these properties.

1.3.1 Von Neumann entropy

Similarly to classical information theory, in quantum information theory there exists a notion of entropy of a quantum state. Let a state be given by its density operator ρ , then the entropy, also known as *Von Neumann entropy*, is defined as

$$S(\rho) = -\text{tr}(\rho \log \rho),$$

where the log is taken in base 2.

If ρ has the eigenvalues λ_x , then the entropy can be rewritten as

$$S(\rho) = -\sum_x \lambda_x \log \lambda_x.$$

Here are a few properties of the Von Neumann entropy which we will use in this thesis.

1. $\forall \rho, S(\rho) \geq 0$. $S(\rho) = 0$ iff ρ is a pure state.
2. In a d -dimensional Hilbert space $S(\rho) \leq \log d$. $S(\rho) = \log d$ iff ρ is in the completely mixed state I/d .
3. If a composite system AB is in a pure state, then $S(A) = S(B)$.
4. For a composite system AB , $S(AB) \leq S(A) + S(B)$ with equality iff the two systems are uncorrelated, i.e., $\rho^{AB} = \rho^A \otimes \rho^B$.

1.3.2 Fidelity

The *fidelity* is a distance measure between quantum states. For density operators ρ and σ it is defined as

$$F(\rho, \sigma) = \text{tr} \left(\sqrt{\rho^{1/2} \sigma \rho^{1/2}} \right).$$

There are two particular cases in which this formula can be simplified. The first is when ρ and σ commute, i.e., diagonalisable in the same basis. If $\rho = \sum_i r_i |i\rangle\langle i|$ and $\sigma = \sum_i s_i |i\rangle\langle i|$, then the fidelity becomes

$$F(\rho, \sigma) = \sum_i \sqrt{r_i s_i}.$$

The second case is when one of the states is in a pure state $|\psi\rangle$ and the other is an arbitrary state ρ , then

$$F(|\psi\rangle, \rho) = \sqrt{\langle \psi | \rho | \psi \rangle}.$$

An important property is that when two states undergo a trace-preserving quantum operation (which is defined in Definition 2.1), their fidelity cannot decrease

$$F(\mathcal{E}(\rho), \mathcal{E}(\sigma)) \geq F(\rho, \sigma).$$

1.3.3 Trace distance

Another distance measure which we will use is the *trace distance*. For two quantum states given by their density operators ρ and σ , it is defined as

$$\delta(\rho, \sigma) = \frac{1}{2} \text{tr} (|\rho - \sigma|).$$

For two states which commute, i.e., are diagonalisable in the same basis, the trace distance reduces to the classical variational distance of two probability distributions given by the eigenvalues of the density operators. Let $\rho = \sum_i r_i |i\rangle\langle i|$ and $\sigma = \sum_i s_i |i\rangle\langle i|$, then the trace distance becomes

$$\delta(\rho, \sigma) = \frac{1}{2} \sum_i |r_i - s_i|.$$

As with the fidelity, a trace-preserving quantum operation (which is defined in Definition 2.1) can only make two states “closer”, so the trace distance does not increase:

$$\delta(\mathcal{E}(\rho), \mathcal{E}(\sigma)) \leq \delta(\rho, \sigma).$$

1.3.4 Commuting observables

Lemma 1.1. *Let M_1 and M_2 be two observables which commute, i.e., are diagonal in the same basis. Then if a state ρ is measured with M_1 and M_2 , the probability of obtaining the outcome (m_1, m_2) does not depend on the order in which the measurements are performed.*

Proof. Let $M_1 = \sum_{m_1} m_1 P_{m_1}$ and $M_2 = \sum_{m_2} m_2 P_{m_2}$ be diagonal in a the basis $\{|i\rangle\}$. Then $P_{m_1} = \sum_i \alpha_{m_1}^i |i\rangle\langle i|$ and $P_{m_2} = \sum_i \beta_{m_2}^i |i\rangle\langle i|$ with $\alpha_{m_1}^i, \beta_{m_2}^i \in \{0, 1\}$ for all m_1, m_2 . If we first apply the measurement of M_1 to ρ , followed the measurement of M_2 , then we will end up in the state $P_{m_2} P_{m_1} \rho P_{m_1} P_{m_2}$ with probability $\text{tr}(P_{m_2} P_{m_1} \rho)$. But

$$\begin{aligned} P_{m_2} P_{m_1} &= \sum_i \alpha_{m_1}^i |i\rangle\langle i| \sum_j \beta_{m_2}^j |j\rangle\langle j| \\ &= \sum_i \alpha_{m_1}^i \beta_{m_2}^i |i\rangle\langle i| \\ &= P_{m_1} P_{m_2}. \end{aligned}$$

So the result of the measurements will be the same states with the same probabilities, no matter in which order the measurements are performed. \square

The importance of these commuting observables is that the outcome of such measurements can be reduced to a classical problem, which can be described by classical random variables X_1 and X_2 with a joint probability distribution $P_{X_1 X_2}$. If only the measurement M_1 is preformed, then X_1 with probability distribution $P_{X_1}(\cdot) = \sum_{x_2} P_{X_1 X_2}(\cdot, x_2)$ describes the outcome. So whether the measurement M_2 was performed or not, or whether the outcome is simply unknown, the probability distribution of X_1 is the same. This naturally also applies for X_2 .

In the following chapters we will very often use this property and describe the outcome of measurements of commuting observables with random variables.

Chapter 2

Quantum error correcting codes

Alice would like to send some qubits to Bob. But the quantum channel they have access to is noisy. So to make sure Bob receives what she sends, Alice will encode her qubits into some Hilbert space of higher dimension. The idea behind quantum error correction is the same as for classical error correction. If the noise level is not too high, Bob can recover Alice's original state by taking the "closest" code-word to the one he receives.

In the Section 2.1 we will explain what kind of noise can act on quantum channels and how it affects qubits. In Section 2.2 we will define the concept of quantum error correcting and see how quantum error correcting codes work. In Section 2.3 we will then have a look at a class of very useful codes, the CSS codes. And finally in Section 2.4 we will explain the stabiliser formalism, which is used to describe a family of codes, including CSS codes.

2.1 Quantum noise

2.1.1 Trace-preserving quantum operation

Noise can be seen as the interaction of an environment on the quantum system of interest. It can be described as a unitary transform applied to the composition of the quantum system of interest and its environment. The state of our system after it has been affected is obtained by tracing out the environment. If our system started in the state ρ then its state after being affected by the noise is

$$\mathcal{E}(\rho) = \text{tr}_E \left(U(\rho \otimes \rho_E)U^\dagger \right), \quad (2.1)$$

where ρ_E is the initial state of the environment and U is the unitary transformation, the "noise", that the whole system undergoes.

Let $|e_k\rangle$ be an orthonormal basis for the environment, and $\rho_E = |e_0\rangle\langle e_0|$ the initial state of the environment. There is no loss of generality in assuming the environment starts in a pure state, as if this was not the case, we could append an extra system to purify it. Equation (2.1) can be rewritten as

$$\mathcal{E}(\rho) = \sum_k \langle e_k | \left(U(\rho \otimes |e_0\rangle\langle e_0|)U^\dagger \right) | e_k \rangle \quad (2.2)$$

$$= \sum_k E_k \rho E_k^\dagger, \quad (2.3)$$

where $E_k := \langle e_k | U | e_0 \rangle$.

Definition 2.1. Let \mathcal{E} be a trace-preserving quantum operation modelling some noise affecting a system. Equation (2.1) gives the *system-environment model* of the quantum operation. Equation (2.3) is known as the *operator-sum* representation of \mathcal{E} . The operators E_k are the *operation elements* for the quantum operation \mathcal{E} . The operation is trace preserving if $\text{tr}(\mathcal{E}(\rho)) = 1$ for all ρ , so the operation elements must respect $\sum_k E_k^\dagger E_k = I$.

Note 2.2. The exact operation elements $\{E_k\}$ depend on the basis $\{|e_k\rangle\}$ chosen for the environment. A different basis leads to different $\{E_k\}$, but resulting in the same operation \mathcal{E} .

The advantage of the operator-sum representation is that we do not need to take the environment into account. The error is given as a set of operations acting only on our system.

2.1.2 Non-trace-preserving quantum operations

It is often useful to consider non-trace-preserving quantum operations, for which $\sum_k E_k^\dagger E_k \leq I$. Such an operation can be seen as resulting from an extra measurement with a known outcome. Let the measurement $\{M_m\}$ be applied to our system and its environment, giving the result m . Then, if our system started in state ρ , the environment in state $|e_0\rangle\langle e_0|$, and a unitary operation U affects our system and its environment, the final state of our system after tracing out the environment is

$$\frac{\text{tr}_E \left(M_m U(\rho \otimes |e_0\rangle\langle e_0|)U^\dagger M_m^\dagger \right)}{\text{tr} \left(M_m U(\rho \otimes |e_0\rangle\langle e_0|)U^\dagger M_m^\dagger \right)}.$$

We will define the operation \mathcal{E}_m as

$$\mathcal{E}_m(\rho) = \text{tr}_E \left(M_m U(\rho \otimes |e_0\rangle\langle e_0|)U^\dagger M_m^\dagger \right) \quad (2.4)$$

$$= \sum_k \langle e_k | \left(M_m U(\rho \otimes |e_0\rangle\langle e_0|)U^\dagger M_m^\dagger \right) | e_k \rangle \quad (2.5)$$

$$= \sum_k E_k \rho E_k^\dagger, \quad (2.6)$$

where $E_k := \langle e_k | M_m U | e_0 \rangle$.

Definition 2.3. Similarly to Definition 2.1 Equations (2.4) and (2.6) give us the system-environment and the operator-sum representations of the non-trace-preserving quantum operation \mathcal{E}_m .

Modelling a quantum operation in terms of its interaction with the environment is not the only possible way to define it. An axiomatic approach is also possible, see e.g., [13].

2.1.3 Mocking up a quantum operation

A quantum operation \mathcal{E} , which is given by its operator-sum representation $\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger$, it can easily be expressed in a system-environment form. If the operation is trace-preserving, then by labelling the initial environment state $|e_0\rangle$, we can define the unitary operation acting on both systems as

$$U|\psi\rangle|e_0\rangle := \sum_k (E_k|\psi\rangle)|e_k\rangle. \quad (2.7)$$

We can see that U is unitary by applying it to states $|\phi\rangle|e_0\rangle$ and $|\psi\rangle|e_0\rangle$ and checking that the scalar product is preserved:

$$\begin{aligned} (U|\phi\rangle|e_0\rangle)^\dagger U|\psi\rangle|e_0\rangle &= \sum_k \left(\langle \phi | E_k^\dagger \right) \langle e_k | \sum_\ell (E_\ell |\psi\rangle) |e_\ell\rangle \\ &= \sum_k \langle \phi | E_k^\dagger E_k |\psi\rangle \\ &= \langle \phi | \psi \rangle, \end{aligned}$$

where the last equality was obtained because $\sum_k E_k^\dagger E_k = I$.

If the operation is non-trace-preserving, we first need to complete it with an additional operation element E_{n+1} , such that $E_{n+1}^\dagger E_{n+1} = I - \sum_{k=1}^n E_k^\dagger E_k$. We can now define the unitary operation similarly to Equation (2.7):

$$U|\psi\rangle|e_0\rangle := \sum_{k=1}^{n+1} (E_k|\psi\rangle)|e_k\rangle, \quad (2.8)$$

and the projector onto the subspace which is affected by \mathcal{E} :

$$P := I \otimes \sum_{k=1}^n |e_k\rangle\langle e_k|. \quad (2.9)$$

By putting Equations (2.8) and (2.9) together we get an equation in the form of (2.4)

$$\mathcal{E}(\rho) = \text{tr}_E \left(P U (\rho \otimes |e_0\rangle\langle e_0|) U^\dagger P^\dagger \right). \quad (2.10)$$

This expansion of a quantum operation to an ancilla system can be seen as a purely mathematical construct. The unitary operator U obtained is not necessarily the one which was applied, but is equivalent in all aspects for the subsystem on which the state ρ lies. We will use it pretty frequently in future chapters to expand some quantum operator \mathcal{E} to a unitary operator acting on the main system and an ancilla.

2.2 Quantum error correction

2.2.1 Error correcting codes

An error correcting code should provide both a way to encode a message and correct it once it is received. The error correction is often done in two stages. First a measurement operator is applied to identify the error that occurred. The outcome of that measurement is called the *error syndrome*. The second stage is a *recovery* step, in which an operator is applied to correct the error corresponding to the syndrome measured in the first stage.

Definition 2.4. A quantum error correcting code is a subspace C of a Hilbert space \mathcal{H}_d . To use such a code we need to define a mapping between an orthonormal basis of our message space and the code space. But this mapping has no influence on the error correcting properties of C so it will often be omitted.

Definition 2.5. An error correction strategy \mathcal{R} consists of a measurement operator and set of unitary operations $(\{M_i\}, \{U_i\})$, where $\{M_i\}$ is the measurement done to obtain the syndrome i , and $\{U_i\}$ are the unitary operations used to correct the error with syndrome i .

Let \mathcal{R} be the quantum operation associated with the error correction strategy $\mathcal{R} = (\{M_i\}, \{U_i\})$. The state after error correction is

$$\begin{aligned} \mathcal{R}(\sigma) &= \sum_i \Pr[\text{measurement outcome } i] \frac{U_i M_i \sigma M_i^\dagger U_i^\dagger}{\text{tr}(U_i M_i \sigma M_i^\dagger U_i^\dagger)} \\ &= \sum_i U_i M_i \sigma M_i^\dagger U_i^\dagger = \sum_i R_i \sigma R_i^\dagger, \end{aligned}$$

where $R_i := U_i M_i$. Very often we will simply consider the operation \mathcal{R} and its elements $\{R_i\}$ and not mention \mathcal{R} and the decomposition of $\{R_i\}$ in $\{M_i\}$ and $\{U_i\}$.

Note 2.6. The error correction operation \mathcal{R} may be expanded into one unitary operation U by adjoining an ancilla system, which “contains” the result of the measurement. This procedure is very similar to what was done in Section 2.1.3 to expand a quantum operation to include the environment.

Let the ancilla system start in the pure state $|0\rangle$. U is then defined as follows:

$$U|\psi\rangle|0\rangle := \sum_i (U_i M_i |\psi\rangle) |i\rangle. \quad (2.11)$$

A code and correction strategy protect against the error operation \mathcal{E} if for all ρ in C

$$(\mathcal{R} \circ \mathcal{E})(\rho) \propto \rho. \quad (2.12)$$

We do not require equality in Equation (2.12), because the error is not necessarily trace preserving. But due to linearity the proportionality factor must be a constant c independent of ρ .

In the Lemma 2.7 and Note 2.8 we will show that if a set of errors $\{E_k\}$ is correctable, then it can be expressed in a different basis, in which every error operator brings a code-word to an orthogonal subspace. It can then be corrected by identifying the subspace and rotating the code-word back.

Lemma 2.7. *Given a quantum operation \mathcal{E} with elements $\{E_k\}$ and a recovery operation \mathcal{R} with elements $\{R_k\}$ which successfully corrects all code words in C subject to \mathcal{E} , there exist equivalent operation elements $\{F_k\}$ and $\{Q_k\}$ such that $\mathcal{E} = \mathcal{F}$, $\mathcal{R} = \mathcal{Q}$ and $Q_i F_j \sqrt{\rho} = \delta_{ij} d_i \sqrt{\rho}$ for some real number d_i and all ρ in C .*

Proof. \mathcal{R} recovers errors from \mathcal{E} so according to Equation (2.12) for all ρ in C we have

$$(\mathcal{R} \circ \mathcal{E})(\rho) = c\rho.$$

By expanding this system to include the environment and an ancilla system for the recovery measurement, as described in Section 2.1.3 and Note 2.6, we get

$$(R \otimes I_E)(E \otimes I_R)(\rho \otimes |e_0\rangle\langle e_0| \otimes |r_0\rangle\langle r_0|)(E^\dagger \otimes I_R)(R^\dagger \otimes I_E) = c\rho \otimes \sigma_{ER}, \quad (2.13)$$

where E is the unitary noise operation acting on the main system and the environment, R is the recovery operation acting on the main system and the recovery system and σ_{ER} is the state of the environment and recovery system after the noise and recovery operations.

Note that σ_{ER} is a pure state: before the operations \mathcal{E} and \mathcal{R} were applied, the entropy of the system was

$$S(\rho \otimes |e_0\rangle\langle e_0| \otimes |r_0\rangle\langle r_0|) = S(\rho).$$

After the noise and recovery operations the entropy of the normalised state is

$$S(\rho \otimes \sigma_{ER}) = S(\rho) + S(\sigma_{ER}).$$

But as the operations R and E are unitary, they did not increase the entropy, and we must have $S(\sigma_{ER}) = 0$.

So we can take the Schmidt decomposition of σ_{ER} and get

$$\sigma_{ER} = \sum_{ik} \lambda_i \lambda_k |i\rangle_E \langle i|_R \langle k|_E \langle k|_R$$

We now recall Note 2.2, which says that the operation elements depend on the basis chosen for the environment, and rewrite Equation (2.13) with the noise and recovery operations expressed in the Schmidt basis:

$$\sum_{ijkl} Q_i F_j \rho F_k^\dagger Q_l^\dagger \otimes |j\rangle_E \langle k|_E \otimes |i\rangle_R \langle l|_R = c\rho \otimes \sum_{ik} \lambda_i \lambda_k |i\rangle_E \langle k|_E \otimes |i\rangle_R \langle k|_R, \quad (2.14)$$

where $Q_i = \langle i|_R R|r_0\rangle_R$ and $F_j = \langle j|_E E|e_0\rangle_E$. Equation (2.14) compares two matrices. For them to be equal we need every ‘‘cell’’ to be equal, in particular

$$Q_i F_j \rho F_j^\dagger Q_i^\dagger \otimes |j\rangle_E \langle j|_E \otimes |i\rangle_R \langle i|_R = c\rho \otimes \delta_{ij} \lambda_i^2 |j\rangle_E \langle j|_E \otimes |i\rangle_R \langle i|_R.$$

Therefore $Q_i F_j \sqrt{\rho} = \delta_{ij} \lambda_i \sqrt{c} \sqrt{\rho}$. \square

Note 2.8. The operation elements $\{Q_i\}$ obtained in Lemma 2.7 can be seen as a measurement which collapses the error onto the error operator F_i with syndrome i and corrects it. As these errors are distinguished with probability 1, the elements $\{F_i\}$ must rotate the code space to orthogonal subspaces. So the $\{Q_i\}$ could be decomposed in a projective measurement $\{M_i\}$ which identifies the subspace to which the errors $\{F_i\}$ take elements of the code C and a unitary operator $\{U_i\}$ which rotates them back to the original code space, such as defined in Definition 2.5. In other words Lemma 2.7 says that an error which is correctable can be expressed in a basis which takes the code space to orthogonal subspaces, and to correct it we need to identify those subspaces and rotate them back.

2.2.2 Discretisation of the errors

So far we have defined a recovery procedure as being able to correct a particular error operation \mathcal{E} . But in reality we do not necessarily know what error is affecting our system. In this section we will show that it is in fact sufficient to correct a discrete set of errors on some qubits to correct arbitrary errors on these qubits.

Theorem 2.9. *If \mathcal{R} can recover from noise process \mathcal{E} with operation elements $\{E_i\}$ for all states on a code space C , then \mathcal{R} can recover from any quantum operation \mathcal{F} whose operation elements $\{F_j\}$ are linear combinations of $\{E_i\}$, i.e., $F_j = \sum_i m_{ji} E_i$ for some complex numbers m_{ji} .*

Proof. Without loss of generality we can assume the operation elements to be given in the Schmidt bases calculated in Lemma 2.7, so

$$R_k E_i \sqrt{\rho} = \delta_{ki} d_k \sqrt{\rho}.$$

For \mathcal{F} we now have

$$\begin{aligned} R_k F_j \sqrt{\rho} &= \sum_i m_{ji} R_k E_i \sqrt{\rho} \\ &= \sum_i m_{ji} \delta_{ki} d_k \sqrt{\rho} \\ &= m_{jk} d_k \sqrt{\rho}. \end{aligned}$$

And the noise and error correction procedures together produce

$$\begin{aligned} (\mathcal{R} \circ \mathcal{F})(\rho) &= \sum_{kj} R_k F_j \rho F_j^\dagger R_k^\dagger \\ &= \sum_{kj} |m_{jk}|^2 d_k^2 \rho \\ &\propto \rho. \end{aligned}$$

So the operation \mathcal{R} corrects the error \mathcal{F} according to Equation (2.12). \square

We now have the tools necessary to correct arbitrary errors on a limited number of qubits. Any error acting on one qubit can be written as a linear combination of the Pauli matrices (X , Y , Z and I) acting on that qubit. So it is sufficient to design a code and recovery strategy, which can correct these errors, to correct arbitrary errors on one qubit. Similarly if we can correct Pauli errors on t qubits, we can correct arbitrary errors on t qubits.

In the next section we will illustrate these error correcting properties with an example, the Shor code, which we have taken from [13].

Example: Shor code

The Shor code protects against an arbitrary error on a single qubit. But according to Theorem 2.9 to show this it is sufficient to make sure we can correct bit flips, phase flips and combined bit and phase flips.

The encoding is defined the following way:

$$\begin{aligned} |0\rangle \rightarrow |0_L\rangle &:= \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} \\ |1\rangle \rightarrow |1_L\rangle &:= \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} \end{aligned} \quad (2.15)$$

So the code space is spanned by $|0_L\rangle$ and $|1_L\rangle$

Let's first look at how this error protects against a bit flip on a single qubit. Suppose Alice sends Bob the state $|\psi\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$. If a bit flip occurred on one of the two first qubits, we can detect it by measuring the observable $Z_1Z_2 = (|00\rangle\langle 00| + |11\rangle\langle 11|) - (|01\rangle\langle 01| + |10\rangle\langle 10|)$. $|\psi\rangle$ is in the $+1$ eigenspace of Z_1Z_2 and $X_1|\psi\rangle$ and $X_2|\psi\rangle$ are in the -1 eigenspace. So the measurement will give the syndrome $+1$ if no bit flip occurred and -1 if it did, without modifying the state of the code word measured.

Similarly by measuring the observable Z_2Z_3 it is possible to determine whether a bit flip occurred on the second or third qubit. Given that not more than one bit flip occurred on the whole code word and the results of the two measurements Z_1Z_2 and Z_2Z_3 , it is possible to determine exactly which qubit was flipped. It can then be corrected by being flipped once more.

We will use the same strategy to correct phase flips. A measurement of $X_1X_2X_3X_4X_5X_6$ and $X_4X_5X_6X_7X_8X_9$ will tell us whether there was a phase flip on one of the first 6 respectively last 6 qubits. We can then correct the error by flipping it again, e.g., by applying the operator $Z_1Z_2Z_3$ to flip the phase of the first 3 qubits.

The observables used to detect a bit flip and a phase flip commute. So they are diagonalisable in the same basis, and the measurement with one will not modify the outcome of the measurement with the other¹. So if both a bit and a phase flip occurred on the same qubit, say the operator X_1Z_1 was applied, then our error correcting strategy would correct them both.

Let's now suppose we want to correct an arbitrary error with operations elements $\{E_i\}$ which affects the first qubit. Each E_i can be written as

$$E_i = e_{i0}I + e_{i1}X_1 + e_{i2}Z_1 + e_{i3}X_1Z_1.$$

So the state $E_i|\psi\rangle$ can be written as a superposition of $|\psi\rangle$, $X_1|\psi\rangle$, $Z_1|\psi\rangle$ and $X_1Z_1|\psi\rangle$ and measuring the syndrome will collapse it into one of those four states, then correct it.

2.2.3 Correcting “small” errors

So far we have seen that if we can correct bit and phase flips on some qubits, then we can correct arbitrary errors on these qubits. But that does not seem to be of much use against errors which would act on every qubit, because it is clearly impossible to correct arbitrary errors on every qubit. But if the error is “small” we hope that only correcting a few qubits will be enough. We can see the intuition for this by looking at the Shor code example. Measuring the syndrome which identifies bit or phase flips collapses the error to a bit flip, phase flip, both or neither. So if it gets projected often enough on the identity error, only correcting a few qubits should be enough.

¹see Section 1.3.4

A measure for how close a state is to containing no more than t bit or phase flips is its probability to get projected on the subspace of all states differing from the original state by no more than t bit or phase flips. The following theorem gives a bound on the fidelity of error correction using this distance measure.

Theorem 2.10. (Adapted from [10]) *Suppose Alice sends Bob the state $|\psi\rangle\langle\psi|$ from an error correcting code C , 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 then corrects it to obtain ρ' . For the fidelity² F of the recovered state we have*

$$F^2 = \langle\psi|\rho'|\psi\rangle \geq \text{tr}(\Pi_S\rho),$$

where Π_S is the projector onto the subspace spanned by all states which are obtained by flipping no more than t bits and no more than t phases of $|\psi\rangle$.

Proof. Let S be our main quantum system containing $|\psi\rangle$, E be the environment and R an ancilla system we append for error correction, as described in Section 2.1.3 and Note 2.6. ρ can be seen as the reduced density matrix of some pure state $|\psi_{SE}\rangle$ which is the result of the noise acting on S and E .

Π_S is the projector onto the subspace which after error correction will be brought back to $|\psi\rangle$. Any other state will probably be corrected to the wrong code word. So we can decompose the state before error correction into a “good” and a “bad” component

$$\begin{aligned} |\psi_{\text{good}}\rangle &= (\Pi_S \otimes I_{ER})|\psi_{SE}\rangle \otimes |0_R\rangle, \\ |\psi_{\text{bad}}\rangle &= ((I_S - \Pi_S) \otimes I_{ER})|\psi_{SE}\rangle \otimes |0_R\rangle. \end{aligned}$$

The unitary recovery operation defined on S and R will take these states to $|\psi'_{\text{good}}\rangle$ and $|\psi'_{\text{bad}}\rangle$. Since the recovery procedure works perfectly in the subspace defined by Π_S we have

$$|\psi'_{\text{good}}\rangle = |\psi\rangle \otimes |\text{junk}_{ER}\rangle.$$

And its norm is:

$$\begin{aligned} \langle\psi'_{\text{good}}|\psi'_{\text{good}}\rangle &= \langle\psi_{\text{good}}|\psi_{\text{good}}\rangle \\ &= \langle\psi_{SE}|(\Pi_S \otimes I_E)|\psi_{SE}\rangle \\ &= \text{tr}((\Pi_S \otimes I_E)|\psi_{SE}\rangle\langle\psi_{SE}|) \\ &= \text{tr}(\Pi_S\rho). \end{aligned}$$

²see Section 1.3.2

Finally the fidelity squared of the state after error correction is

$$F^2 = \langle \psi | \rho' | \psi \rangle \quad (2.16)$$

$$\geq \langle \psi'_{SER} | (|\psi\rangle\langle\psi| \otimes I_{ER}) | \psi'_{SER} \rangle \quad (2.17)$$

$$= \langle \psi'_{good} | (|\psi\rangle\langle\psi| \otimes I_{ER}) | \psi'_{good} \rangle + \langle \psi'_{good} | (|\psi\rangle\langle\psi| \otimes I_{ER}) | \psi'_{bad} \rangle \\ + \langle \psi'_{bad} | (|\psi\rangle\langle\psi| \otimes I_{ER}) | \psi'_{good} \rangle + \langle \psi'_{bad} | (|\psi\rangle\langle\psi| \otimes I_{ER}) | \psi'_{bad} \rangle \quad (2.18)$$

$$= \text{tr}(\Pi_S \rho) + \langle \psi'_{good} | \psi'_{bad} \rangle \\ + \langle \psi'_{bad} | \psi'_{good} \rangle + \langle \psi'_{bad} | (|\psi\rangle\langle\psi| \otimes I_{ER}) | \psi'_{bad} \rangle \quad (2.19)$$

$$= \text{tr}(\Pi_S \rho) + \langle \psi'_{bad} | (|\psi\rangle\langle\psi| \otimes I_{ER}) | \psi'_{bad} \rangle \quad (2.20)$$

$$\geq \text{tr}(\Pi_S \rho),$$

where in the step from (2.18) to (2.19) the equality $|\psi'_{good}\rangle = |\psi\rangle \otimes |\text{junk}_{ER}\rangle$ was used, and from (2.19) to (2.20) the orthogonality of $|\psi'_{good}\rangle$ and $|\psi'_{bad}\rangle$. \square

In Chapter 3 this theorem will be used in a practical way: we will sample the errors on some qubits and measure the number of flips to get an estimate of the fidelity of the state after error correction.

2.3 CSS codes

2.3.1 Definition

We will now have a look at a very useful class of error correcting codes, the so-called Calderbank-Shor-Steane codes.

Definition 2.11. Let C_1 and C_2 be $[n, k_1]$ and $[n, k_2]$ classical linear codes with $C_2 \subset C_1$, such that both C_1 and C_2^\perp correct t errors. We define the $[n, k_1 - k_2]$ quantum error correcting code $\text{CSS}(C_1, C_2)$ capable of correcting errors on t qubits as the space spanned by the following states

$$\forall x \in C_1, \quad |x + C_2\rangle := \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle.$$

It is easy to see that for $x, x' \in C_1$, such that $x - x' \in C_2$, we have $|x + C_2\rangle = |x' + C_2\rangle$. So the state $|x + C_2\rangle$ only depends on the coset³ of C_1/C_2 in which x is. And if x and x' belong to different cosets, $\nexists y, y' \in C_2$, s.t. $x + y = x' + y'$, so $|x + C_2\rangle$ and $|x' + C_2\rangle$ are orthogonal. We now see that the dimension of $\text{CSS}(C_1, C_2)$ is the number of cosets of C_1/C_2 , which is $|C_1|/|C_2| = 2^{k_1 - k_2}$.

³Let G be a group and H a subgroup of G . Let $g \in G$, then $gH = \{gh | h \in H\}$ is a left coset of H in G .

2.3.2 Error correction

To correct against bit and phase flips we will use the error correcting properties of the codes C_1 and C_2 . Let the bit flips be described by a n vector e_1 with 1 at the positions where bit flips occurred, and the phase flips be similarly described by a vector e_2 . If the original state was $|x + C_2\rangle$ the corrupted one is

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y) \cdot e_2} |x + y + e_1\rangle. \quad (2.21)$$

We will now measure the syndrome for the error e_1 by using the parity matrix H_1 of the code C_1 . The easiest way to see how the operation is done is by introducing an ancilla system and applying an operator which will take $|x + y + e_1\rangle|0\rangle$ to $|x + y + e_1\rangle|H_1(x + y + e_1)\rangle = |x + y + e_1\rangle|H_1 e_1\rangle$. By measuring the ancilla we can get the syndrome $H_1 e_1$, compute the error e_1 (provided it has no more than t 1s) and correct the state by applying bit flips where appropriate, taking the state to

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

To correct phase flips we apply Hadamards to every qubit to turn phase flips into bit flips:

$$|\psi\rangle = H^{\otimes n} \left(\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y) \cdot e_2} |x + y\rangle \right) \quad (2.22)$$

$$= \frac{1}{\sqrt{|C_2|2^n}} \sum_z \sum_{y \in C_2} (-1)^{(x+y) \cdot (e_2+z)} |z\rangle \quad (2.23)$$

$$= \frac{1}{\sqrt{|C_2|2^n}} \sum_{z'} \sum_{y \in C_2} (-1)^{(x+y) \cdot z'} |z' + e_2\rangle, \text{ where } z' := z + e_2 \quad (2.24)$$

$$= \frac{1}{\sqrt{2^n/|C_2|}} \sum_{z' \in C_2^\perp} (-1)^{x \cdot z'} |z' + e_2\rangle. \quad (2.25)$$

In Equation (2.24) for $z' \in C_2^\perp$ we have $y \cdot z' = 0$, so $(x + y) \cdot z' = x \cdot z'$. For $z' \notin C_2^\perp$ we have $|\{y \in C_2 : y \cdot z' = 1\}| = |\{y \in C_2 : y \cdot z' = 0\}|$, so $\sum_{y \in C_2} (-1)^{(x+y) \cdot z'} = 0$.

Equation (2.25) is similar to Equation (2.21), the error e_2 corresponds to bit flips of words in the code C_2^\perp . And we can correct it the same way, by using the parity check matrix H_2 for the code C_2^\perp . We then correct the bit flips and apply Hadamards again to get back to the original state.

2.4 Stabiliser codes

In this section we will give a few simple definitions needed to understand the stabiliser formalism and codes, which were introduced by Gottesman [3]. For a more detailed insight into stabilisers we refer to e.g., [3, 13].

2.4.1 Stabiliser formalism

Definition 2.12. The *Pauli group* G_n consists of all n -fold tensor products of the Pauli matrices with multiplicative factors ± 1 and $\pm i$. Or in other words:

$$\begin{aligned} G_1 &:= \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}, \\ G_n &:= \{g_1 \otimes g_{n-1} : g_1 \in G_1, g_{n-1} \in G_{n-1}\}. \end{aligned}$$

Definition 2.13. Let S be a subgroup of G_n and V_S the set of n qubit states which are fixed by every element of S :

$$V_S := \{|\psi\rangle : g|\psi\rangle = |\psi\rangle, \forall g \in S\} \quad \text{for } S \subset G_n,$$

S is said to be the stabiliser of the space V_S .

Without proof we will state Lemma 2.14 about the dimension of the space stabilised by a stabiliser S .

Lemma 2.14. *If the stabiliser $S = \langle g_1, \dots, g_{n-k} \rangle$ is generated by $n - k$ independent commuting generators, then it stabilises a space of dimension 2^k .*

Intuitively each extra generator divides the space by 2. For a proof of Lemma 2.14 we refer to e.g., [3, 13].

Definition 2.15. Stabiliser codes are a class of error correcting codes, which consist of all code spaces that can be described using a stabiliser. Let the code space C be defined by the stabiliser $S = \langle g_1, \dots, g_{n-k} \rangle$ given by $n - k$ independent generators. By Lemma 2.14 this defines a space of dimension 2^k . Any orthonormal basis of the resulting code space C can be chosen as the code words. But it is more practical to choose $\bar{Z}_1, \dots, \bar{Z}_k \in G_n$ such that $\{g_1, \dots, g_{n-k}, \bar{Z}_1, \dots, \bar{Z}_k\}$ forms an independent and commuting set. The code word $|x_1 \dots x_k\rangle$ in C is then defined by the stabiliser

$$\langle g_1, \dots, g_{n-k}, (-1)^{x_1} \bar{Z}_1, \dots, (-1)^{x_k} \bar{Z}_k \rangle.$$

Notice that $\forall x_1, \dots, x_k \in \{0, 1\}$, $\{g_1, \dots, g_{n-k}, (-1)^{x_1} \bar{Z}_1, \dots, (-1)^{x_k} \bar{Z}_k\}$ forms a commuting set, because $\{g_1, \dots, g_{n-k}, \bar{Z}_1, \dots, \bar{Z}_k\}$ commute, so the code words are well defined.

The advantages of using the stabiliser formalism becomes more apparent when we explain how to do syndrome measurement and error correction on a stabiliser code in the following lemmas.

Lemma 2.16. *Let $S = \langle g_1, \dots, g_{n-k} \rangle$ be the stabiliser of a code C given by its generators and the operators $\{\bar{Z}_1, \dots, \bar{Z}_k\}$ which define the basis as in Definition 2.15. If there exists an error correction strategy to correct a set of Pauli errors $\{E_i\}$, then the combination of the observables $\{g_1, \dots, g_{n-k}\}$ can be used as measurement operator for that strategy. The corresponding recovery operation is the Pauli operator in $\{E_i\}$, which is unique for every syndrome measured in the first step.*

Proof. Every Pauli operator either commutes or anti-commutes with every other Pauli operator. For a Pauli operator E_i , define

$$\beta_\ell^i := \begin{cases} 0 & \text{if } E_i \text{ and } g_\ell \text{ commute,} \\ 1 & \text{if } E_i \text{ and } g_\ell \text{ anti-commute,} \end{cases}$$

$$\alpha_\ell^i := \begin{cases} 0 & \text{if } E_i \text{ and } \bar{Z}_\ell \text{ commute,} \\ 1 & \text{if } E_i \text{ and } \bar{Z}_\ell \text{ anti-commute.} \end{cases}$$

Let $|x_1 \dots x_k\rangle$ be a code word of the stabiliser code C defined by the stabiliser

$$\langle g_1, \dots, g_{n-k}, (-1)^{x_1} \bar{Z}_1, \dots, (-1)^{x_k} \bar{Z}_k \rangle.$$

Then $E_i|x_1 \dots x_k\rangle$ is stabilised by

$$\langle (-1)^{\beta_1^i} g_1, \dots, (-1)^{\beta_{n-k}^i} g_{n-k}, (-1)^{x_1 + \alpha_1^i} \bar{Z}_1, \dots, (-1)^{x_k + \alpha_k^i} \bar{Z}_k \rangle,$$

and the measurement with the observable g_ℓ will have outcome $+1$ iff $\beta_\ell^i = 0$ and -1 iff $\beta_\ell^i = 1$.

Suppose there exist two Pauli operators E_i and E_j in the correctable set $\{E_i\}$, such that $\forall \ell \beta_\ell^i = \beta_\ell^j$, but $\exists \ell$ s.t. $\alpha_\ell^i \neq \alpha_\ell^j$. Let $|x_1 \dots x_k\rangle$ be defined as above and $|y_1 \dots y_k\rangle$ be defined by the stabiliser

$$\langle g_1, \dots, g_{n-k}, (-1)^{x_1 + \alpha_1^i + \alpha_1^j} \bar{Z}_1, \dots, (-1)^{x_k + \alpha_k^i + \alpha_k^j} \bar{Z}_k \rangle.$$

We now have $|x_1 \dots x_k\rangle, |y_1 \dots y_k\rangle \in C$, such that $|x_1 \dots x_k\rangle \neq |y_1 \dots y_k\rangle$ and $E_i|x_1 \dots x_k\rangle = E_j|y_1 \dots y_k\rangle$. So there can not exist a correction strategy to recover from both E_i and E_j . Therefore any two elements of the correctable Pauli errors $\{E_i\}$ will have a different syndrome $\{\beta_\ell^i\}$ when measured with the generators of the code $\{g_\ell\}$. So as described in Note 2.8 our recovery strategy identifies which Pauli operator was applied (or projects the error onto one Pauli operator) and recovers from it. \square

Lemma 2.16 shows how to perform correction of Pauli errors. But by Theorem 2.9 if we can correct a set of errors $\{E_i\}$ we can correct any linear combination of these errors. And as Pauli errors on t qubits from a basis for arbitrary errors on t qubits, this lemma also tells us how to correct against arbitrary errors on t qubits, provided our code supports correction of all Pauli errors affecting no more than t qubits.

Lemma 2.17. *Let C be the code space defined by the stabiliser*

$$S = \langle g_1, \dots, g_{n-k} \rangle.$$

The subspaces defined by the stabilisers

$$\langle (-1)^{\beta_1} g_1, \dots, (-1)^{\beta_{n-k}} g_{n-k} \rangle \text{ for } \beta_\ell \in \{0, 1\}$$

are mutually orthogonal, span the whole space, any error which can be corrected on C can be corrected on one of these other subspaces C_β , and to perform this correction it is sufficient to measure the syndrome with the $\{g_\ell\}$ and recover the original state by applying the operator to correct the error with syndrome $\beta \oplus \beta^e$, where β is the bit string of $\{\beta_\ell\}$ for the subspace used and β^e is the bit string of the syndrome measured.

Proof. The orthogonality of the subspaces and the fact that they span the whole space are direct consequences of stabiliser properties.

Let β be the bit string of all β_ℓ . If we can correct a state encoded in the code space C , then to correct a state encoded in the subspaces C_β we can apply a unitary operator U_β taking C_β to C , correct the state as if it was encoded in C then apply the reverse unitary operator U_β^\dagger to take the state back to the original one in C_β . Let β^e be the result of the syndrome measurement of the error correction in C , the operations applied to correct a state ρ encoded in C_β and the unnormalised result ρ' are then

$$\rho' = U_\beta^\dagger U_{\beta^e} P_{\beta^e} U_\beta \rho U_\beta^\dagger P_{\beta^e} U_{\beta^e} U_\beta,$$

where P_{β^e} is the projection which results from the syndrome measurement and U_{β^e} is the operator applied to correct the error with syndrome β^e .

The effect of U_β on the syndrome measurement is to flip the results for which $\beta_\ell = 1$. So if the syndrome measurement had been done before the rotation U_β , the syndrome would have been $\beta^{e'} = \beta^e \oplus \beta$, therefore

$$\rho' = U_\beta^\dagger U_{\beta^e \oplus \beta} U_\beta P_{\beta^e} \rho P_{\beta^e} U_\beta^\dagger U_{\beta^e \oplus \beta} U_\beta.$$

The U operators are all Pauli operators, so they either commute or anti-commute and the result can be rewritten

$$\rho' = U_{\beta^e \oplus \beta} P_{\beta^e} \rho P_{\beta^e} U_{\beta^e \oplus \beta}^\dagger.$$

So to perform error correction on a code word encoded using C_β it is sufficient to measure the syndrome $\beta^{e'}$ and apply the operator $U_{\beta^e \oplus \beta}$. \square

2.4.2 CSS codes as stabiliser codes

CSS codes are a nice example of a class of stabiliser codes. Any code word in C_1 , when multiplied by the parity check matrix H_1 , gives 0. Similarly if for each line in H_1 we construct a Pauli operator g_i composed of tensor products of I and Z , with an I for each 0 in the line of H_1 and a Z for each 1, then any code word of $\text{CSS}(C_1, C_2)$ is in the +1 eigenspace of g_i and is stabilised by g_i . In the same way we can find the other stabilisers of the code space $\text{CSS}(C_1, C_2)$ by looking at the parity check matrix H_2 of the code C_2^\perp . All code words in $\text{CSS}(C_1, C_2)$ are stabilised by Pauli operators formed of tensor products of I and X , with an I for each 0 and an X for each 1 in a line of H_2 . H_1 is a $(n - k_1) \times n$ matrix and H_2 is a $k_2 \times n$ matrix. So we now have $n - (k_1 - k_2)$ commuting independent Pauli operators in our stabiliser, which fully defines the $2^{k_1 - k_2}$ code space.

In Lemma 2.17 we introduced the code spaces C_β which are orthogonal to C and have a syndrome β when measured with the stabilisers of C . For a CSS code, from each syndrome we can compute the bit and phase flip strings x and z , so CSS_β can be defined as $\text{CSS}_{x,z}$.

Definition 2.18. Let C_1 and C_2 be two classical linear codes with $C_2 \subset C_1$, such that both C_1 and C_2^\perp correct t errors, as in Definition 2.11. We define the quantum error correcting code $\text{CSS}_{x,z}(C_1, C_2)$ as the space spanned by the following states

$$\forall v \in C_1, \quad |v + C_2\rangle := \frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} (-1)^{(v+w) \cdot z} |v + w + x\rangle.$$

Chapter 3

BB84

BB84 is a prepare-and-measure quantum key distribution scheme proposed by Bennett and Brassard in 1984 [2]. The first general but pretty complex proof of security was given more than ten years later by Mayers [12]. Lo and Chau [9] proposed a much simpler proof of security for quantum key distribution, but their protocol unfortunately required a quantum computer. Shor and Preskill [16] unified the ideas from [12] and [9], and gave a simple and elegant proof of security of standard BB84. In this chapter we follow the proof structure of Shor and Preskill, and take a few elements from [13] and [10].

To prove the security of BB84 we will start with a protocol which relies on entanglement and heavy quantum operations. We will prove this protocol is secure, then reduce it step by step to BB84.

3.1 Modified Lo-Chau

The first protocol is based on entanglement purification[1]. Alice prepares EPR pairs $|\Phi^+\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ and sends half of each pair to Bob. These will get disturbed by noise or Eve's interference, but Bob will extract a smaller number of pure EPR pairs out of them using error correcting codes. Alice and Bob can then measure their EPR pairs in the computational basis to obtain the same key. This protocol is given in Figure 3.1.

To prove this protocol is secure we need to prove that with high probability Eve has exponentially little information about the secret key produced in Step 10. To prove this we will show that our error sampling in Step 8 guarantees that with high probability we will not have more than t phase or bit flips in the qubits used to generate the key, that the fidelity to m EPR pairs of the state produced by entanglement purification in Step 9 is higher than this probability, and given this fidelity is high, the information Eve has is about the key is small.

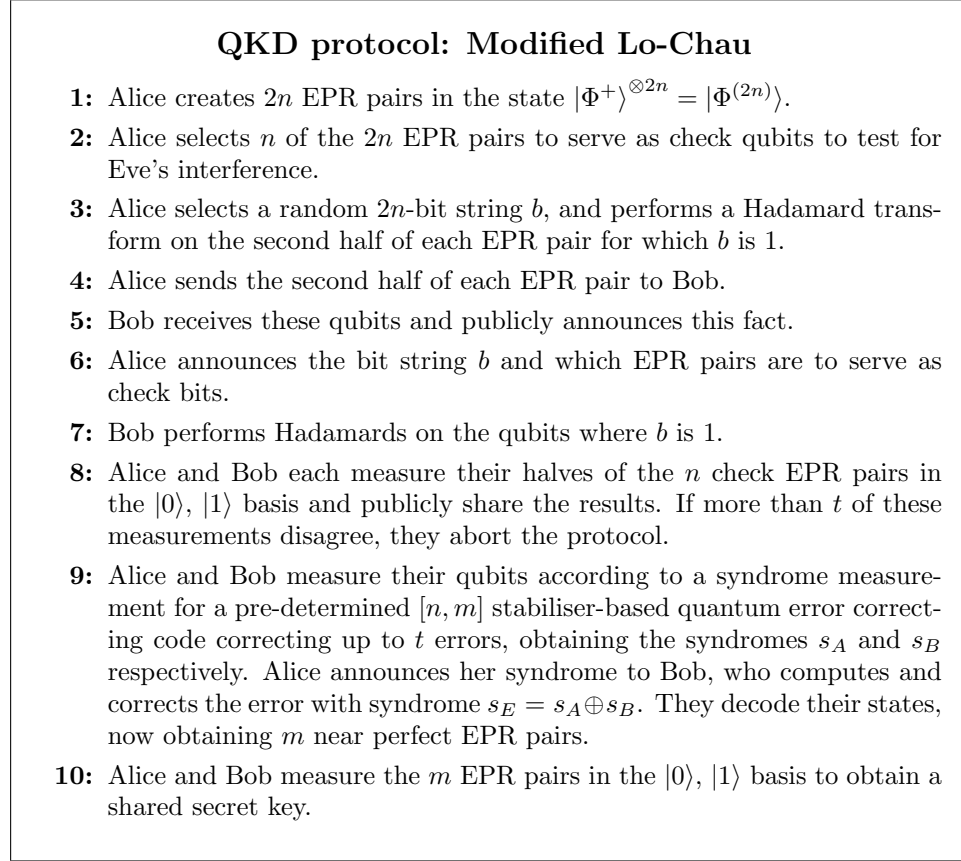


Figure 3.1: A QKD protocol which relies on entanglement purification

Lemma 3.1. *If Alice and Bob measure half of their qubits to sample the error as described in Step 8 of the modified Lo-Chau protocol (Figure 3.1) and find t errors, then the probability that a complete measurement of the remaining qubits in the Bell basis¹ would yield more than $(1 + \epsilon)t$ bit or phase flips is less or equal to $e^{-\theta(\epsilon^2 n)}$.*

Proof. From their measurements on the sample qubits Alice and Bob can detect if a bit flip occurred, or phase flip when they applied a Hadamard. This information corresponds to the result of measuring the sample EPR pairs with the observables $Z \otimes Z$ and $X \otimes X$ respectively. Note that

$$\begin{aligned} Z \otimes Z &= |00\rangle\langle 00| + |11\rangle\langle 11| - |01\rangle\langle 01| - |10\rangle\langle 10| \\ &= |\Phi^+\rangle\langle \Phi^+| + |\Phi^-\rangle\langle \Phi^-| - |\Psi^+\rangle\langle \Psi^+| - |\Psi^-\rangle\langle \Psi^-|, \end{aligned}$$

¹the Bell basis consists of the four Bell states

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad |\Phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \quad |\Psi^+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \quad |\Psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

$$\begin{aligned} X \otimes X &= |00\rangle\langle 11| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 00| \\ &= |\Phi^+\rangle\langle\Phi^+| + |\Psi^+\rangle\langle\Psi^+| - |\Phi^-\rangle\langle\Phi^-| - |\Psi^-\rangle\langle\Psi^-|. \end{aligned}$$

So these observables are diagonal in the Bell basis and commute with any measurement in that basis. Therefore the probabilities of the outcomes of $Z \otimes Z$ and $X \otimes X$ are unmodified by any other measurement in the Bell basis, so we might as well measure all EPR pairs in the Bell basis to reduce the problem to a classical probability one.

Let X_{eve} and Z_{eve} be two random variables representing the number of bit respectively phase flips introduced by Eve, which would be detected by a complete measurement in the Bell basis of all $2n$ EPR pairs. The probability distribution of X_{eve} and Z_{eve} are totally under Eve's control. Let T be a random variable representing the number of errors detected by the sampling of n qubits, and R_{bf} and R_{pf} be two random variables representing the number of bit respectively phase flips in the rest of the qubits. Due to the Hadamards applied randomly R_{bf} and R_{pf} have the same distribution, so we only need to find a bound on one of them.

We are interested in finding a bound on the probability that we have more errors in the qubits used to generate the key than we found by sampling the other half, i.e., a bound for $\Pr[R_{bf} > (1 + \epsilon)T]$. As the measurements with outcome T and R_{bf} commute with the measurements producing X_{eve} and Z_{eve} , we may write

$$\Pr[R_{bf} > (1 + \epsilon)T] = \sum_{x,z} \Pr[R_{bf} > (1 + \epsilon)T | X_{\text{eve}} = x, Z_{\text{eve}} = z] p_{x,z}, \quad (3.1)$$

where $p_{x,z} := \Pr[X_{\text{eve}} = x, Z_{\text{eve}} = z]$. The task of estimating

$$\Pr[R_{bf} > (1 + \epsilon)T | X_{\text{eve}} = x, Z_{\text{eve}} = z] \quad (3.2)$$

is now one of classical probability.

It is known that by randomly sampling elements of a set, the result will be exponentially close to the expected value. More precisely we have

$$\Pr\left[T < \left(1 - \frac{\epsilon}{2}\right) \frac{x+z}{4} \mid X_{\text{eve}} = x, Z_{\text{eve}} = z\right] \leq e^{-\theta(\epsilon^2 n)}, \quad (3.3)$$

$$\Pr\left[R_{bf} > \left(1 + \frac{\epsilon}{2}\right) \frac{x+z}{4} \mid X_{\text{eve}} = x, Z_{\text{eve}} = z\right] \leq e^{-\theta(\epsilon^2 n)}, \quad (3.4)$$

where an exact derivation can be found e.g., in [13]. We can now use (3.3) and (3.4) to find a bound for (3.2). To have $R_{bf} - T > \epsilon T$ we certainly need either R_{bf} or T to be out of the interval $[(1 - \frac{\epsilon}{2}) \frac{x+z}{4}, (1 + \frac{\epsilon}{2}) \frac{x+z}{4}]$, because if not they would be too close to each other to have a distance greater than

ϵT . So

$$\begin{aligned} & \Pr [R_{bf} > (1 + \epsilon)T | X_{\text{eve}} = x, Z_{\text{eve}} = z] \\ & \leq \Pr \left[T < \left(1 - \frac{\epsilon}{2}\right) \frac{x+z}{4} \middle| X_{\text{eve}} = x, Z_{\text{eve}} = z \right] \\ & \quad + \Pr \left[R_{bf} > \left(1 + \frac{\epsilon}{2}\right) \frac{x+z}{4} \middle| X_{\text{eve}} = x, Z_{\text{eve}} = z \right] \\ & \leq e^{-\theta(\epsilon^2 n)}. \end{aligned}$$

By putting this in (3.1) we get the desired probability. \square

We have shown that the probability of having more errors than measured is exponentially small. So if we measure $(1 - \epsilon)t$ errors in the sample qubits, we may assume that we have no more than t errors in the remaining qubits.

Theorem 3.2. *If Alice and Bob use a $[n, m]$ stabiliser-based error correcting code, which can correct up to t bit flips and t phase flips, then the state ρ'' resulting from the entanglement purification performed in Step 9 of the modified Lo-Chau protocol (Figure 3.1) has a fidelity F to m EPR pairs such that*

$$F^2 = \langle \Phi^{(m)} | \rho'' | \Phi^{(m)} \rangle \geq \text{tr}(\Pi\rho),$$

where ρ is the state of the initial n EPR pairs after the noise affected them, and Π is the projector on the subspace spanned by all states which differ from $|\Phi^{(n)}\rangle$ by no more than t bit and t phase flips.

Proof. Let the error which affects Bob's qubits be denoted by \mathcal{E} . Because of the linearity of quantum operations, the state of the system when Bob receives his qubits is

$$\rho = \frac{1}{2^n} \sum_{i,j=0}^{2^n-1} |i\rangle\langle j| \otimes \mathcal{E}(|i\rangle\langle j|). \quad (3.5)$$

Let Alice measure the syndrome of her qubits and obtain s_A with probability p_{s_A} . According to Lemma 2.17 this measurement projects the state on the code space C_{s_A} . The state of the system can be written as $\rho = \sum_{s_A} p_{s_A} \rho'_{s_A}$ where

$$\rho'_{s_A} = \frac{1}{2^m} \sum_{c_i, c_j \in C_{s_A}} |c_i\rangle\langle c_j| \otimes \mathcal{E}(|c_i\rangle\langle c_j|).$$

As this error only affects Bob's system, it can be corrected using only operations on Bob's system. And by Lemma 2.17 again, if Bob measures the syndrome s_B he should correct $s_A \oplus s_B$. Theorem 2.10 holds when the noise affects the whole state, so it holds in particular when only half the qubits are sent. By a simple change of basis $\frac{1}{2^{\frac{m}{2}}} \sum_{c_i \in C_{s_A}} |c_i\rangle|c_i\rangle$ becomes

$\frac{1}{2^{\frac{m}{2}}} \sum_{i=0}^{2^m-1} |i\rangle|i\rangle = |\Phi^{(m)}\rangle$ so by Theorem 2.10 and by linearity of error correction, for the fidelity F of the state ρ'' after error correction we have

$$F^2 = \langle \Phi^{(m)} | \rho'' | \Phi^{(m)} \rangle \geq \sum_{s_A} p_{s_A} \text{tr} (\Pi_{s_A} \rho'_{s_A}),$$

where Π_{s_A} is the projector onto the space spanned by all states which differ from $\frac{1}{2^{\frac{k}{2}}} \sum_{c_i \in C_{s_A}} |c_i\rangle|c_i\rangle$ in no more than t bit flip or t phase flip errors on Bob's system. Let P_{s_A} be the projector onto Alice's subspace C_{s_A} which was applied as a result of her syndrome measurement. The space defined by Π_{s_A} is a subspace of the space defined by P_{s_A} , so $\Pi_{s_A} P_{s_A} = \Pi_{s_A}$ and using $\rho' = \frac{1}{p_{s_A}} P_{s_A} \rho P_{s_A}$ we get:

$$\begin{aligned} \sum_{s_A} p_{s_A} \text{tr} (\Pi_{s_A} \rho'_{s_A}) &= \sum_{s_A} p_{s_A} \text{tr} (\Pi_{s_A} \rho'_{s_A} \Pi_{s_A}) \\ &= \sum_{s_A} p_{s_A} \text{tr} \left(\frac{1}{p_{s_A}} \Pi_{s_A} P_{s_A} \rho P_{s_A} \Pi_{s_A} \right) \\ &= \sum_{s_A} \text{tr} (\Pi_{s_A} \rho) \\ &= \text{tr} (\Pi \rho), \end{aligned}$$

where $\Pi = \sum_{s_A} \Pi_{s_A}$ is the projector on the subspace spanned by all states which differ from $\frac{1}{2^{\frac{n}{2}}} \sum_{i=1}^{2^n} |i\rangle|i\rangle = |\Phi^{(n)}\rangle$ by no more than t bit and t phase flips. \square

We now know that our corrected states have 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 Lemma 3.1. So we still have to show that having a high fidelity implies security, which is done in the two next lemmas.

Lemma 3.3. *If the fidelity squared between ρ and $|\Phi^{(m)}\rangle$ is such that $F^2 = \langle \Phi^{(m)} | \rho | \Phi^{(m)} \rangle \geq 1 - 2^{-s}$ then $S(\rho) < 2^{-s}(s + 2m + \frac{1}{\ln 2}) + O(2^{-2s})$.*

Proof. $\langle \Phi^{(m)} | \rho | \Phi^{(m)} \rangle \geq 1 - 2^{-s}$ means that the largest eigenvalue of ρ is larger than $1 - 2^{-s}$. The entropy is maximised when the other $2^{2m} - 1$ eigenvalues are equal, so

$$\begin{aligned} S(\rho) &\leq -(1 - 2^{-s}) \log(1 - 2^{-s}) - 2^{-s} \log \frac{2^{-s}}{2^{2m} - 1} \\ &< -(1 - 2^{-s}) \left(\frac{-2^{-s}}{\ln 2} + O(2^{-2s}) \right) + 2^{-s}(s + 2m) \\ &= 2^{-s} \left(s + 2m + \frac{1}{\ln 2} \right) + O(2^{-2s}). \end{aligned}$$

\square

The following lemma is in fact a corollary of the Holevo bound[5].

Lemma 3.4. *Let $|\psi\rangle$ be a pure state shared by Alice, Bob and Eve, and ρ be Alice's and Bob's part $\rho = \text{tr}_E(|\psi\rangle\langle\psi|)$. Let $\{M_x\}$ be a measurement performed by Alice and Bob with the outcome given by the random variable X , then $I(X;Y) \leq S(\rho)$ for any measurement $\{M_y\}$ Eve can perform on her system.*

Proof. Let $\sigma = \text{tr}_{AB}(|\psi\rangle\langle\psi|)$ be the state of Eve's system before the measurement of $\{M_x\}$ and

$$\sigma_x = \frac{\text{tr}_{AB} \left((M_x \otimes I_E) |\psi\rangle\langle\psi| (M_x^\dagger \otimes I_E) \right)}{\text{tr} \left((M_x \otimes I_E) |\psi\rangle\langle\psi| (M_x^\dagger \otimes I_E) \right)}$$

the state after the measurement, which occurs with probability

$$p_x = \text{tr} \left((M_x \otimes I_E) |\psi\rangle\langle\psi| (M_x^\dagger \otimes I_E) \right).$$

$\{M_x\}$ is a general measurement, not a projective one, but we still have $\sigma = \sum_x p_x \sigma_x$, because the measurement is performed on Alice and Bob's system, and σ is in Eve's.

Eve's task is to find a measurement $\{M_y\}$ with outcome Y to distinguish between the different σ_x . A bound is known on $I(X;Y)$, the Holevo bound[5], which states that

$$I(X;Y) \leq S(\sigma) - \sum_x p_x S(\sigma_x).$$

So $I(X;Y) \leq S(\sigma) = S(\rho)$. □

In Lemma 3.3 we have shown that if Alice and Bob's state has a fidelity exponentially close to m EPR pairs, then the entropy of Eve's state is exponentially close to 0. In Lemma 3.4 we then showed that any information Eve could get from her state about Alice and Bob's state is smaller than her entropy, which concludes the proof of security for the modified Lo-Chau protocol. We can now reduce it a prepare-and-measure protocol.

3.2 CSS codes

Now that we have a key distribution protocol, which we have proven secure, we would like to modify it to obtain a prepare-and-measure protocol. The first step will be to remove the need for EPR pairs. To do this we need to notice that the measurements Alice does at the end of the modified Lo-Chau protocol could have been done at the start without any change to the rest

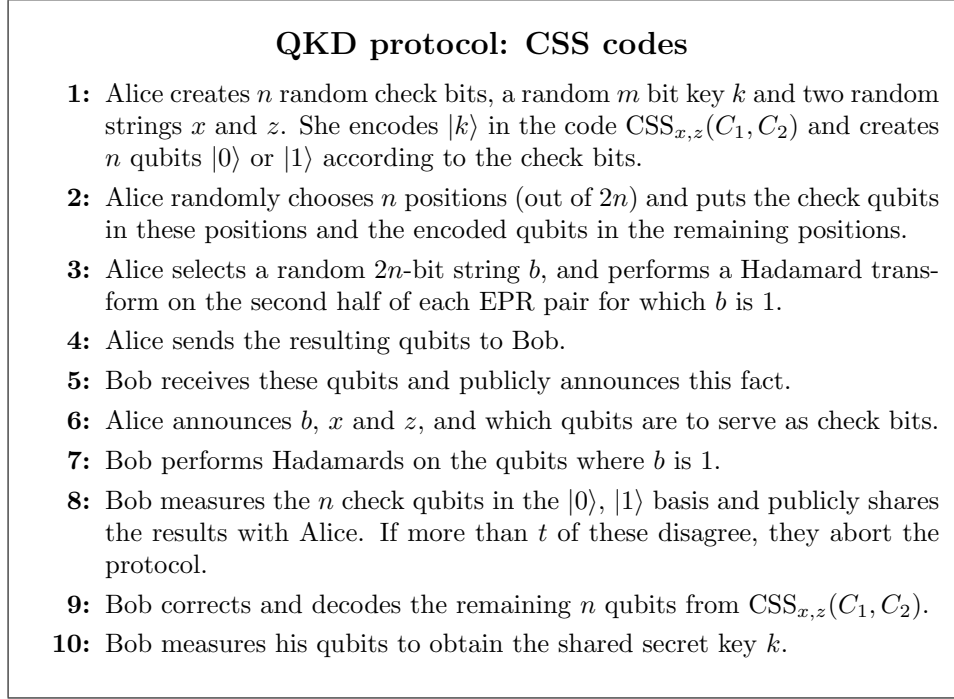


Figure 3.2: A QKD protocol which uses CSS codes

of the states held by the other players. The result is the CSS codes protocol given in Figure 3.2.

When Alice measures the check qubits in Step 8 of the modified Lo-Chau protocol in Figure 3.1 on Page 24, she collapses the EPR pairs randomly into $|0\rangle$ or $|1\rangle$. Instead of doing this, she might as well have randomly chosen 0 or 1 and sent the corresponding states to Bob, which is now done in Steps 1 and 2 of the CSS codes protocol.

When Alice measures her syndrome s_A she projects the EPR pairs onto the code space C_{s_A} . Let \mathcal{E} be the error operator affecting Bob's qubits. The state of the system before Alice's measurement is then as given in Equation (3.5). The probability of obtaining the outcome s_A is

$$\begin{aligned}
p_{s_A} &= \text{tr} \left((P_{s_A} \otimes I) \frac{1}{2^n} \sum_{i,j=0}^{2^n-1} |i\rangle\langle j| \otimes \mathcal{E}(|i\rangle\langle j|) (P_{s_A} \otimes I) \right) \\
&= \text{tr} \left(\frac{1}{2^n} \sum_{c_i, c_j \in C_{s_A}} |c_i\rangle\langle c_j| \otimes \mathcal{E}(|c_i\rangle\langle c_j|) \right) \\
&= \frac{1}{2^n} \sum_{c_i, c_j \in C_{s_A}} \text{tr}(|c_i\rangle\langle c_j|) \text{tr}(\mathcal{E}(|c_i\rangle\langle c_j|))
\end{aligned}$$

$$= \frac{1}{2^{n-m}},$$

where P_{s_A} is the projector on the code space C_{s_A} . In the last step, the trace-preserving property of \mathcal{E} is used.

The probabilities $\{p_{s_A}\}$ are equal and independent of the subspaces $\{s_A\}$. So instead of letting the measurement randomly choose which code space will be used, Alice could have chosen it herself and sent Bob a superposition of all codes in that space (the B half of $\frac{1}{2^{\frac{m}{2}}} \sum_{c_i \in C_{s_A}} |c_i\rangle_A |c_i\rangle_B$), and then announced which code space she used instead of measuring the syndrome, which she does in the new protocol at Step 6.

When they measure the m EPR pairs in Step 10 of the Lo-Chau protocol, each possible key is produced with equal probability. So here again Alice could have chosen the key randomly herself and encoded it in whatever code they use, then let Bob decode it, which is now done at Steps 1 and 9.

In the new protocol Alice and Bob use the code $\text{CSS}_{x,z}(C_1, C_2)$. When Alice announces which code she has used, she just needs to specify which bit and phase flips x and z she has introduced. Different bit and phase flips can lead to the same code, e.g., any bit flip vector in C_2 will leave the code word unchanged. But as all cosets are of the same size, the number of bit and phase flip vectors which lead to the same code space are equal, so by choosing them uniformly at randomly Alice chooses a code space uniformly at randomly.

3.3 BB84 protocol

The CSS codes protocol does not make use of EPR pairs anymore. But we still need a full quantum computer to perform the encoding and decoding. And Bob needs to store the quantum states he receives, while waiting for Alice to send him the information he needs for decoding. To get around these two problems we need to notice that Bob can first measure and then decode classically. The result is the BB84 protocol given in Figure 3.3.

Let the state Bob receives encoded in $\text{CSS}_{x,z}(C_1, C_2)$ be

$$|v + C_2\rangle' = \frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} (-1)^{(v+w) \cdot (z+e_z)} |v + w + x + e_x\rangle, \quad (3.6)$$

where e_x and e_z are the errors introduced by Eve or noise. In the CSS codes protocol in Figure 3.2 on Page 29 in Step 9, when he decodes his state, he will first correct the flips x , z , e_x and e_z , obtaining the state

$$|v + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} |v + w\rangle.$$

Then he will identify the coset of $v + C_2$ and use this as the key. But instead Bob could start by measuring in the Z basis, then decode classical. If he

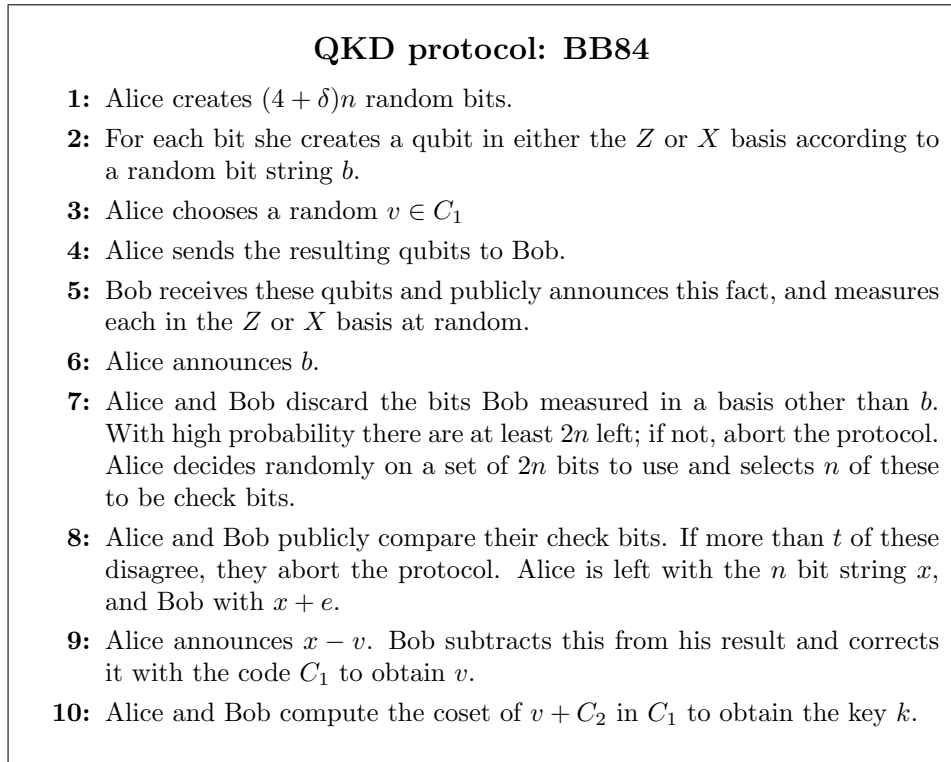


Figure 3.3: A prepare-and-measure QKD protocol

measures the state (3.6) in the Z basis, he will get $v + w + x + e_x$ for some $w \in C_2$ uniformly at random. Alice can then send him x , which he will subtract. He can correct e_x by using the error correcting properties of C_1 (because $v + w \in C_1$), then calculate the coset of $v + w$ obtaining the key k . Notice that Bob does not need z to perform this decoding.

But this can still be simplified further. Instead of letting the measurement choose w randomly, Alice can choose it. $v + w$ is a random string in C_1 , so Alice just needs to randomly choose $v \in C_1$. The string $v + x$ is totally random, so Alice could simply send Bob $|x\rangle$, let him measure it to obtain $x + e_x$, then send him $x - v$ and let him subtract it to get $v + e_x$ which he corrects and decodes as above.

We still have one problem, Bob needs to know which qubits have had a Hadamard applied to them and has to store his qubits while waiting for this information. The solution is to send Bob $(4 + \delta)n$ qubits instead of $2n$, let him randomly choose which he applies a Hadamard to, and hope that he got it right in at least $2n$ of the cases (which will happen with very high probability). As Bob's next step is now to measure in the Z basis, instead of applying Hadamards to some qubits he can measure them in the X basis. All these modifications finally give the BB84 protocol.

3.4 QKD error rates

We have given a prepare-and-measure quantum key distribution protocol and proven it is secure. But our protocol and proof uses tools such as error correcting codes, which can only work if the number of errors is limited. So we need to find bounds on what error rates our protocol and QKD in general can tolerate. In Section 3.4.1 we show that no matter what method we use to correct the errors Eve introduced, we can not make the protocol secure if there are more than 25% errors. In Section 3.4.2 we show that CSS codes can correct up to 11% errors, which gives us a lower bound on what is achievable by BB84.

3.4.1 Upper bound

The upper bound on the tolerable error rate of BB84 is due to an intercept and resend eavesdropping strategy. Eve can intercept every qubit sent by Alice. She measures it randomly in the X or Z basis, and prepares a new qubit in the measured state, which she sends to Bob. Whatever Bob does from this point on can be simulated by a classical random variable prepared by Eve, making secure QKD impossible.

Let Alice send the qubit $|0\rangle$. Eve will choose the wrong basis with probability $\frac{1}{2}$, in which case the state will either become $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ or $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$. If Bob chooses the correct basis, the state will get projected back to $|0\rangle$ or to $|1\rangle$, each with probability $\frac{1}{2}$. So the measured error rate will be $\frac{1}{4} = 25\%$, and is the same whatever state Alice sends.

3.4.2 Lower bound

The error rate a BB84 protocol can tolerate is limited by the number of errors the error correcting code used can correct. The Gilbert-Varshamov bound [11] tells us if n is sufficiently large, classical linear $[n, k]$ error correcting codes protecting against t errors exist for which²

$$\frac{k}{n} \geq 1 - h\left(\frac{2t}{n}\right). \quad (3.7)$$

If we use two such codes for C_1 and C_2^\perp in our CSS code, then (3.7) becomes

$$\frac{k_1}{n} \geq 1 - h\left(\frac{2t}{n}\right), \quad (3.8)$$

$$\frac{n - k_2}{n} \geq 1 - h\left(\frac{2t}{n}\right). \quad (3.9)$$

² h is the binary entropy function: $h(x) = -x \log x - (1-x) \log(1-x)$

By adding (3.8) and (3.9) we get

$$\begin{aligned} \frac{n + k_1 - k_2}{n} &\geq 2 - 2h\left(\frac{2t}{n}\right) \\ \Rightarrow \frac{k_1 - k_2}{n} &\geq 1 - 2h\left(\frac{2t}{n}\right), \end{aligned} \quad (3.10)$$

which gives a lower bound on the achievable rate of CSS codes.

But if we do not want a code which corrects all errors consisting of no more than t flips, but only corrects the errors with high probability, then it is possible to achieve higher rates. Shannon's theorem on the existence of good codes [11] tells us that for any $\epsilon > 0$ if n is sufficiently large there exists a code with rate

$$\frac{k}{n} = 1 - h\left(\frac{t}{n}\right), \quad (3.11)$$

with error probability $P_{\text{err}} < \epsilon$ when each bit is flipped with probability t/n .

To use such a code, the errors need to be random. So an extra step must be introduced in the protocol to randomise them. Alice applies a random permutation to her bits. Once Bob has received and measured the qubits, she announces the permutation and Bob applies the inverse operation.

By using the new rate (3.11) instead of (3.7), instead of the bound in (3.10) we get

$$\frac{k_1 - k_2}{n} = 1 - 2h\left(\frac{t}{n}\right). \quad (3.12)$$

So to have a positive rate we need $1 - 2h(t/n) > 0$, so

$$t < h^{-1}\left(\frac{1}{2}\right)n \approx 0.11n. \quad (3.13)$$

Equation (3.13) shows that CSS codes can only correct up to error rates of 11%, so that is the maximum rate to which BB84 is secure.

Chapter 4

Decoupling bit and phase correction

BB84 uses an error correcting code to correct both bit and phase flips. But these two steps can also be regarded as error correction and privacy amplification: correcting bit flips is all that is necessary for Alice and Bob to share the same key, but Eve still gains information about it through her entanglement with phase flips, so by correcting these we perform privacy amplification. In BB84 these two steps are strongly correlated, because they correspond to operations for one error correcting code and must commute. [8] proposes a method to decouple these two processes, if Alice and Bob share an initial secret key. But Renner, Gisin and Kraus [14] using a result from [6, 15] have completely decoupled these two processes, and given a new information theoretic security proof for QKD protocols, which does not rely on entanglement purification anymore. They achieve this by applying classical privacy amplification and proving that it is still secure against a quantum adversary.

An advantage of decoupling these two processes is that we can consider different procedures to treat bit and phase flips. The tolerable error rate of BB84 is limited by the tolerable error rate of the error correcting codes used. So by decoupling the two, we can hopefully find methods with higher tolerable error rates. In this chapter we will use [6, 15] to achieve this separation.

4.1 Privacy amplification

The recent result by Koenig and Renner [6, 15] gives a bound on the security of privacy amplification when the adversary has some quantum knowledge about our secret key. More precisely, they prove the following theorem

Theorem 4.1. *Let X be a random variable with range \mathcal{X} and Rényi en-*

tropy¹ $R(X) = m$, and let G be a two-universal random function² from \mathcal{X} to $\{0, 1\}^s$. Then for any family of states³ $\{\rho_x\}_{x \in \mathcal{X}}$ in a Hilbert space \mathcal{H}_{2^r} of dimension 2^r ,

$$d(G(X)|\rho_X; G) \leq \frac{1}{2} 2^{-\frac{m-r-s}{2}}, \quad (4.1)$$

where d is the non-uniformity⁴ of $G(X)$ given ρ_X and G , defined as

$$d(Z|\rho_X; G) = \delta \left(\sum_{x,g,z} p_{xgz} |z\rangle\langle z| \otimes \rho_x \otimes |g\rangle\langle g|, \sum_{x,g,u} p_{xgu} |u\rangle\langle u| \otimes \rho_x \otimes |g\rangle\langle g| \right), \quad (4.2)$$

with $p_{xgz} = \Pr[X = x, G = g, Z = z]$, U is a uniform random variable with the same range as Z , $\{|g\rangle\}$, $\{|z\rangle\}$ and $\{|u\rangle\}$ are orthogonal basis, and $\delta(\cdot, \cdot)$ is the trace distance as defined in Section 1.3.3.

It might not be obvious at first why the definition of non-uniformity given in Theorem 4.1 implies security. So in the following corollary we will show that it implies that Eve cannot extract any information from ρ_X by a measurement which would depend on her knowledge of G .

Corollary 4.2. *The definition of non-uniformity given in (4.2) implies that for the outcome Y of any measurement of ρ_X depending on G we would have*

$$d(Z|Y) \leq d(Z|\rho_X; G),$$

where $d(Z|Y)$ is the classical non-uniformity of random variable Z given Y , which is the expected value over Y of the variational distance between the probability distribution of Z given Y and a uniform distribution U : $d(Z|Y) = \frac{1}{2} \sum_y P_Y(y) \sum_z |P_{Z|Y}(z, y) - P_U(z)|$.

Proof. Let Eve perform a measurement $\{M_y^g\}_y$ which depends on the value of G and has outcome Y . If we represent the classical information G quantumly by saying Eve has the state $\sum_{x,g} p_{xg} \rho_x \otimes |g\rangle\langle g|$, as has been done in (4.2), applying this measurement corresponds to measuring with $\{M_y^g \otimes |g\rangle\langle g|\}_{y,g}$. Eve can write the outcome in an ancilla system and discard the X and G systems. Applying this to Equation (4.2) we get

$$\begin{aligned} d(Z|\rho_X; G) &= \delta \left(\sum_{x,g,z} p_{xgz} |z\rangle\langle z| \otimes \rho_x \otimes |g\rangle\langle g|, \sum_{x,g,u} p_{xgu} |u\rangle\langle u| \otimes \rho_x \otimes |g\rangle\langle g| \right) \\ &\geq \delta \left(\sum_{y,z} p_{yz} |z\rangle\langle z| \otimes |y\rangle\langle y|, \sum_{u,y} p_{uy} |u\rangle\langle u| \otimes |y\rangle\langle y| \right). \end{aligned} \quad (4.3)$$

¹Then Rényi entropy is defined as $R(X) := -\log \sum_{x \in \mathcal{X}} p_x^2$, where p_x is the probability of outcome x

²A random function G from \mathcal{X} to \mathcal{Y} is called *two-universal* if $\Pr[G(x) = G(x')] \leq 1/|\mathcal{Y}|$ holds for any distinct $x, x' \in \mathcal{X}$

³These states represent the adversaries knowledge of X . The size of the Hilbert space corresponds to the adversaries memory size, namely a Hilbert space of dimension 2^r would mean a memory of r qubits.

⁴We refer to [15] for a definition of non-uniformity using a more adapted notation

The inequality in (4.3) comes from the fact that $\delta(\cdot, \cdot)$ cannot increase when the same quantum operation is applied to both arguments (see Section 1.3.3).

Both arguments of $\delta(\cdot, \cdot)$ in Equation 4.3 are diagonal in the same basis, because $|z\rangle = |u\rangle$ as Z and U have the same range, only their probabilities differ. So by a property of the trace distance given in Section 1.3.3, it is equal to the variational distance of their probability distributions. We then get:

$$\begin{aligned} d(Z|\rho_X; G) &\geq \delta\left(\sum_{y,z} p_{yz}|z\rangle\langle z| \otimes |y\rangle\langle y|, \sum_{u,y} p_{uy}|u\rangle\langle u| \otimes |y\rangle\langle y|\right) \\ &= \frac{1}{2} \sum_{y,z} |P_{ZY}(z, y) - P_{UY}(z, y)| \\ &= \frac{1}{2} \sum_y P_Y(y) \sum_z |P_{Z|Y}(z, y) - P_U(z)|, \end{aligned}$$

where the last line comes from the independence of U from the other random variables. \square

It is clear that if no measurement with outcome Y can distinguish the probability distribution of Z (respectively $G(X)$ in the particular case of Theorem 4.1) from a uniform distribution, then Y does not give us any information about Z and our key as is secure as with Lemma 3.4.

In order to share the same key it is sufficient for Alice and Bob to correct bit flips and measure their EPR pairs in the computational basis. This key will not be secure, because it will still be strongly entangled with Eve's state through the phase flips. But Alice and Bob can then do privacy amplification and by Theorem 4.1 make the key secure if they sacrifice enough bits.

So instead of using an error correcting code, which corrects both bit and phase flips, we propose to only correct bit flips, then measure the resulting states and apply privacy amplification to get a secret key. The first protocol, EPR-based, is given in Figure 4.1. Similarly to the proof of security for BB84, we will prove this protocol is secure, then modify it to obtain an equivalent prepare-and-measure protocol.

The error sampling in Step 8 of the EPR-based protocol in Figure 4.1 is identical to Step 8 of the modified Lo-Chau protocol in Figure 3.1 on Page 24. So Lemma 3.1 still applies and we can assume we do not have more than t bit or phase flips in the qubits remaining after error sampling.

Step 9 differs from the modified Lo-Chau protocol only in that we use another code, one which corrects only bit flips. But Lemmas 2.16 and 2.17 apply to any stabiliser code, thus ensuring that Alice and Bob will share the same key after error correction and measurement, if the syndrome measurement projects the state on one which is correctable (which happens with high probability according to Lemma 3.1).

QKD Protocol: EPR-based with 2-universal privacy amplification

- 1:** Alice creates $2n$ EPR pairs in the state $|\Phi^+\rangle^{\otimes 2n} = |\Phi^{(2n)}\rangle$.
- 2:** Alice selects n of the $2n$ EPR pairs to serve as check bits to test for Eve's interference.
- 3:** Alice selects a random $2n$ -bit string b , and performs a Hadamard transform on the second half of each EPR pair for which b is 1.
- 4:** Alice sends the second half of each EPR pair to Bob.
- 5:** Bob receives these qubits and publicly announces this fact.
- 6:** Alice announces the bit string b and which EPR pairs are to serve as check bits.
- 7:** Bob performs Hadamards on the qubits where b is 1.
- 8:** Alice and Bob each measure their halves of the n check EPR pairs in the $|0\rangle, |1\rangle$ basis and publicly share the results. If more than t of these measurements disagree, they abort the protocol.
- 9:** Alice and Bob measure their qubits according to a syndrome measurement for a pre-determined $[n, m]$ stabiliser-based quantum error correcting code correcting up to t bit flips, obtaining the syndromes s_A and s_B respectively. Alice announces her syndrome to Bob, who computes and corrects the error with syndrome $s_E = s_A \oplus s_B$. They decode their states, now obtaining a superposition of the 2^m computational basis states.
- 10:** Alice and Bob measure their states in the computational basis obtaining a shared (insecure) key x .
- 11:** Alice fixes the randomness of a two-universal random function $G: \{0, 1\}^m \rightarrow \{0, 1\}^s$, obtaining g , and communicates this to Bob.
- 12:** Alice and Bob compute the final key as $k = g(x)$.

Figure 4.1: A QKD protocol which relies on EPR pairs and the security of 2-universal privacy amplification

The security of this protocol doesn't rely on the fidelity to EPR pairs and Theorem 3.2 anymore, but on the security of privacy amplification against an adversary with quantum information about our key and Theorem 4.1.

Theorem 4.3. *Let X be the random variable taking the value of the key obtained at Step 9 of the EPR-based protocol (Figure 4.1) with range $\{0, 1\}^m$, ρ_X the state of Eve's system and G the two-universal random function used for privacy amplification with range $\{0, 1\}^s$. If $s = m - nh(t/n) - \zeta$, then*

$$d(G(X)|\rho_X; G) \leq \frac{1}{2}2^{-\frac{\zeta}{2}}.$$

Proof. Let us choose the Pauli operators as a basis to express the noise or tampering affecting our system. Let $E_{x,z}$ be the Pauli operator with X

operators at the positions indicated by the vector x , Z indicated by z (and Y when both x_i and z_i are 1). As our error sampling showed us that we do not have more than t errors, we must have $w(x) \leq t$ and $w(z) \leq t$, where w is the hamming weight function. Let Eve's system start in the state $|0\rangle$. After her tampering the combined system is in the state

$$\begin{aligned} |\Psi_{ABE}\rangle &= \left(\sum_{x,z} \lambda_{x,z} E_{x,z} \otimes |x,z\rangle_E |0\rangle_E \right) \left| \Phi^{(n)} \right\rangle_{AB} |0\rangle_E \\ &= \frac{1}{2^{\frac{n}{2}}} \sum_{i,x,z} \lambda_{x,z} (-1)^{i \cdot z} |i\rangle_A |i+x\rangle_B |x,z\rangle_E. \end{aligned} \quad (4.4)$$

Let Alice and Bob perform the bit flip error correction as described in Step 9. The state in (4.4) becomes (before decoding)

$$|\Psi'_{ABE}\rangle = \frac{1}{2^{\frac{m}{2}}} \sum_{c,z} \lambda_z (-1)^{c \cdot z} |c\rangle_A |c\rangle_B |z\rangle_E, \quad (4.5)$$

where $c \in C_{s_A}$ and $w(z) \leq t$.

The measurement in the computational basis performed in Step 10 produces each key with equal probability. Therefore the Rényi entropy of X is

$$R(X) = -\log \sum_{x \in \mathcal{X}} p_x^2 = m. \quad (4.6)$$

From (4.5) we can see that the size of Eve's quantum memory is maximised when the states $|z\rangle$ are orthonormal, and the λ_z uniform. z can take any value as long as $w(z) \leq t$, i.e., $\sum_{j=0}^t \binom{n}{j}$ values. So the entropy of Eve's system is

$$S(E) \leq -\log \sum_{j=0}^t \binom{n}{j} \leq nh \left(\frac{t}{n} \right). \quad (4.7)$$

By placing the bound on Eve's memory size obtained in Equation (4.7) and the Rényi entropy of X from Equation (4.6) in the Equation (4.1) of Theorem 4.1, we get

$$d(G(X)|\rho_X; G) \leq \frac{1}{2} 2^{-\frac{m-nh(t/n)-s}{2}}.$$

We then set

$$s := m - nh \left(\frac{t}{n} \right) - \zeta. \quad (4.8)$$

in the above equation to conclude the proof. \square

If $\zeta = \theta(n)$, e.g., $\zeta = \delta n$ for a small δ , then the non-uniformity will go to 0 for n going to infinity.

4.2 Reduction to prepare-and-measure

We would now like to reduce our EPR-based protocol to a prepare-and-measure one similar to BB84. The only differences between this EPR-based protocol using privacy amplification and the modified Lo-Chau protocol from Chapter 3 are in Step 9 and the new Steps 11 and 12. So only for these 3 steps will the reduction be different.

QKD protocol: BB84-like with 2-universal privacy amplification

- 1:** Alice creates $(4 + \delta)n$ random bits.
- 2:** For each bit she creates a qubit in either the Z or X basis according to a random bit string b .
- 3:** Alice chooses a random $v \in C$
- 4:** Alice sends the resulting qubits to Bob.
- 5:** Bob receives these qubits and publicly announces this fact, and measures each in the Z or X basis at random.
- 6:** Alice announces b .
- 7:** Alice and Bob discard the bits Bob measured in a basis other than b . With high probability there are at least $2n$ left; if not, abort the protocol. Alice decides randomly on a set of $2n$ bits to use and selects n of these to be check bits.
- 8:** Alice and Bob publicly compare their check bits. If more than t of these disagree, they abort the protocol. Alice is left with the n bit string x , and Bob with $x \oplus e$.
- 9:** Alice announces $x \oplus v$. Bob adds this to his result, and corrects it with the code C to obtain v .
- 10:** Bob then decodes v obtaining y such that $v = Gy$, where G is the generator of C .
- 11:** Alice fixes the randomness of a two-universal random function $G: \{0,1\}^m \rightarrow \{0,1\}^s$, obtaining g , and communicates this to Bob.
- 12:** Alice and Bob compute the final key as $k = g(y)$.

Figure 4.2: A prepare-and-measure QKD protocol which uses privacy amplification

The privacy amplification steps are in fact already classical steps, so we do not need to modify them. The difference in Step 9 between the two protocols is that a different code is used. Instead of using a CSS code, composed of two classical codes, C_1 and C_2 , used to correct bit and phase flips, we only make use of one classical linear code C , which corrects only bit flips. The reduction is nearly identical. Instead of letting a measurement

randomly choose the code used, Alice might as well choose it herself. Bob can also do the measurement before the error correcting, and then do this classically. And instead of announcing the code space and letting another measurement choose randomly a word in the code space, Alice might choose the code-word herself, $v \in C$, and send it to Bob as $x \oplus v$, which is a totally random bit string. This is exactly identical to what was done in Chapter 3, so we refer to that for more details.

The only slight difference is in the decoding, in Step 10, where instead of finding the coset of $v + C_2$ in C_1 , Alice and Bob simply decode v from C , finding the y with $Gy = v$, where G is the generator of C .

4.3 Maximum error rate

As in Section 3.4.2 we would like to calculate what error rate our protocol can tolerate. For the linear code part we can reuse the Shannon code rate given in Equation (3.11) if we randomly permute the qubits again. For the privacy amplification, part, the rate is given by Equation (4.8). By putting the two together we get

$$s = n \left(1 - 2h \left(\frac{t}{n} \right) \right) - \zeta.$$

In order to have a positive rate we need $1 - 2h(t/n) > 0$, so

$$t < h^{-1} \left(\frac{1}{2} \right) n = 0.11n.$$

Exactly the same maximum tolerable error rate as in (3.13) for BB84.

Chapter 5

Error detection

BB84 corrects the errors introduced by Eve to un-entangle her states from Alice and Bob's and make the result of their final measurement secret. In this chapter we will use an idea proposed by Gottesman and Lo [4] and perform error detection. Instead of correcting every error, we try to detect them and drop the qubits with them.

The error detection is done by blocks, so we first randomly permute the qubits and form blocks. The error detection is done by checking the parities of random samples for each block. If we find an error, we drop the block, if not we keep it. The blocks left are not totally bit flip free, because some errors might survive the random sample parity check step. So we use an error correcting code to correct them. When we have bit flip free blocks, we can measure them and apply classical privacy amplification as in the previous chapter.

5.1 The protocol

The protocol which we will discuss throughout this chapter is given in Figure 5.1. The difference between this protocol and the one from Chapter 4 given in Figure 4.1 is that the new protocol has two extra steps, Steps 9 and 10, in which we try to detect bit flips. For each block we measure parities of random qubits in the Z basis, and if we do not detect differences between Alice's and Bob's block, we keep these blocks, else we discard them.

This confirmation step is very similar a the syndrome measurement, as we have done in previous protocols for error correction (and in this protocol in Step 11). Let us look at the Conf_ℓ protocol given in Figure 5.2 in a bit more detail. The measurements done in Step 2 are equivalent to a syndrome measurement for a random code. To see this let us define β_i as

$$\beta_i := \begin{cases} 0 & \text{if the outcome of the measurement } g_i \text{ gives } +1, \\ 1 & \text{if the outcome of the measurement } g_i \text{ gives } -1. \end{cases}$$

QKD Protocol: EPR-based with block confirmation and 2-universal privacy amplification

- 1:** Alice creates $2n$ EPR pairs in the state $|\Phi^+\rangle^{\otimes 2n} = |\Phi^{(2n)}\rangle$.
- 2:** Alice selects n of the $2n$ EPR pairs to serve as check bits to test for Eve's interference.
- 3:** Alice selects a random $2n$ -bit string b , and performs a Hadamard transform on the second half of each EPR pair for which b is 1.
- 4:** Alice sends the second half of each EPR pair to Bob.
- 5:** Bob receives these qubits and publicly announces this fact.
- 6:** Alice announces the bit string b and which EPR pairs are to serve as check bits.
- 7:** Bob performs Hadamards on the qubits where b is 1.
- 8:** Alice and Bob each measure their halves of the n check EPR pairs in the $|0\rangle, |1\rangle$ basis and publicly share the results. If more than t of these measurements disagree, they abort the protocol.
- 9:** Alice announces a random permutation of the remaining n qubits, and Alice and Bob permute the qubits accordingly. They form $\frac{n}{\log n}$ blocks B_j of size $\log n$, containing their qubits in positions $(j-1)\log n + 1$ to $j\log n$.
- 10:** For each block B_j Alice and Bob run^a $\text{Conf}_\ell(B_j)$ and either drop it if the confirmation fails, or obtain a block encoded in a random code space C_{β_j} of dimension $2^{\log n - \ell}$, where ℓ is the number of iterations (of tests) of the Conf_ℓ protocol. Let μ be the number of blocks left. If it is too small to produce a key of the desired length, they abort the protocol.
- 11:** Alice and Bob measure their qubits according to a syndrome measurement for a $[\mu \log n, \nu]$ stabiliser-based quantum error correcting code correcting up to t' bit flips, for some small t' , obtaining the syndromes s_A and s_B respectively. Alice announces her syndrome to Bob, who computes and corrects the error with syndrome $s_E = s_A \oplus s_B$. But they do not decode it yet.
- 12:** Each of Alice and Bob's systems are now encoded in a code space stabilised by the set of all the generators of the block code spaces $\{C_{\beta_j}\}$ and the error correcting code space C_{s_A} . So Alice and Bob decode from this space and measure their states in the computational basis obtaining a shared (insecure) key x of length $\nu - \mu\ell$.
- 13:** Alice fixes the randomness of a two-universal random function $G: \{0, 1\}^{\nu - \mu\ell} \rightarrow \{0, 1\}^s$, obtaining g , and communicates this to Bob.
- 14:** Alice and Bob compute the final key as $k = g(x)$.

^aThis protocol is given in Figure 5.2

Figure 5.1: A QKD protocol which relies on EPR pairs, error detection and the security of 2-universal privacy amplification

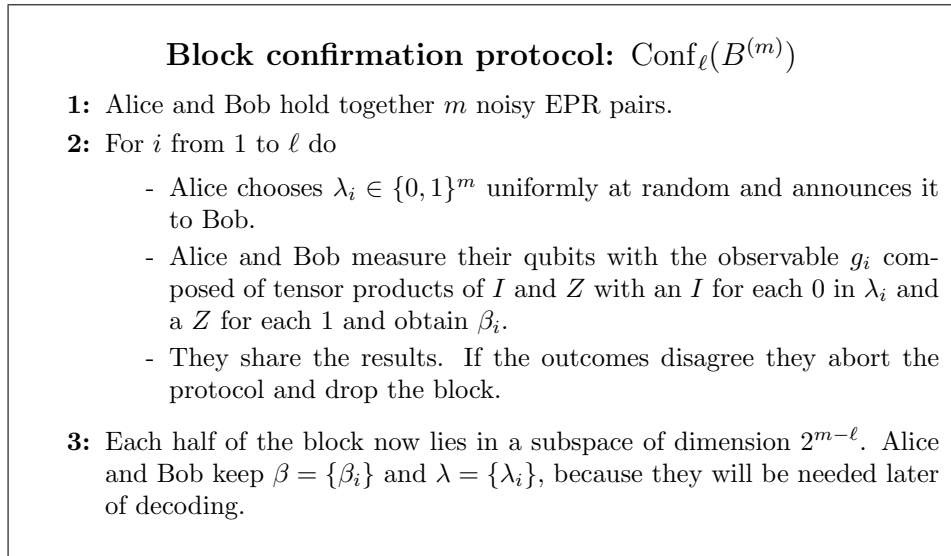


Figure 5.2: A protocol to test a block for bit flips

The states measured with the observables $\{g_i\}$ then get projected onto the subspace stabilised by $\langle(-1)^{\beta_1}g_1, \dots, (-1)^{\beta_\ell}g_\ell\rangle$. This is a code space of dimension $2^{m-\ell}$ of the code C_β , which has syndrome β and is built from a classical linear code with parity check matrix λ , where the i^{th} line of λ is the λ_i chosen randomly in the protocol. As Alice and Bob only keep blocks for which they measured the same syndrome, there is no error correction step, so they don't need to know anything about the error correcting properties of this code.

The next step of the EPR-based protocol in Figure 5.1, the error correcting (Step 11), works identically to the error correction seen in the previous chapter. But the bit flip error rate is hopefully lower after the block confirmation step, and we can use an error correcting code which only corrects a few errors t' . But when we decode in Step 12 we should decode from both the error correcting code space and the random block code spaces used in the block confirmation protocol. We finally apply a random two-universal function in the last steps to make the key secure.

5.2 Residual bit flips

We are now interested in calculating exactly how many bit flips are expected to get through the confirmation phase, and how many qubits are expected to remain after. Let us first have a look at the number of blocks which don't contain any errors. By Lemma 3.1 with high probability we do not have more than t bit or phase errors, so to take the worst case, we may assume

that we have exactly t bit and phase flips. Lemma 5.1 gives us a bound on the expected number of blocks with 0 bit flips. And Lemma 5.3 gives a bound on the probability of being close to the expected value.

Lemma 5.1. *Let N be a random variable representing the number of blocks which do not contain any bit flips. Then*

$$\frac{1}{\log n} n^{1-\log(\frac{1}{1-\epsilon}+o(1))} \leq \mathbf{E}[N] \leq \frac{1}{\log n} n^{1-\log(\frac{1}{1-\epsilon})},$$

where $\epsilon := \frac{t}{n}$.

Proof. Let us define random variables X_i as

$$X_i := \begin{cases} 0 & \text{if block } i \text{ contains a bit flip,} \\ 1 & \text{otherwise.} \end{cases}$$

Then $N = \sum_i X_i$ and

$$\mathbf{E}[N] = \sum_i \mathbf{E}[X_i] = \frac{n}{m} \Pr[X_i = 1], \quad (5.1)$$

where m is the size of a block, and $\frac{n}{m}$ is the number of blocks.

There are t bit flips in the n qubits. So the probability of a block containing no errors is the number of ways to choose m qubits out of the $n - t$ error free qubits over the total number of ways to construct a block:

$$\Pr[X_i = 1] = \frac{\binom{n-t}{m}}{\binom{n}{m}},$$

which can be upper bounded by

$$\begin{aligned} \frac{\binom{n-t}{m}}{\binom{n}{m}} &= \frac{(n-t)!(n-m)!}{(n-t-m)!n!} \\ &= \frac{(n-t) \cdots (n-t-m+1)}{n \cdots (n-m+1)} \\ &\leq \left(\frac{n-t}{n}\right)^m = (1-\epsilon)^m, \end{aligned} \quad (5.2)$$

and lower bounded by

$$\begin{aligned} \frac{\binom{n-t}{m}}{\binom{n}{m}} &= \frac{(n-t)!(n-m)!}{(n-t-m)!n!} \\ &= \frac{(n-t) \cdots (n-t-m+1)}{n \cdots (n-m+1)} \\ &\geq \left(\frac{n-t-m}{n-m}\right)^m = \left(1 - \frac{\epsilon}{1 - \frac{m}{n}}\right)^m. \end{aligned} \quad (5.3)$$

The upper bound of this lemma follows immediately by setting $m := \log n$ in (5.2) and putting this in (5.1). To derive the lower bound we need to notice that $\frac{m}{n} = \frac{\log n}{n} = o(1)$. So

$$\begin{aligned} \mathbf{E}[N] &\geq \frac{n}{\log n} \left(1 - \frac{\epsilon}{1 - o(1)}\right)^{\log n} \\ &= \frac{n}{\log n} n^{\log\left(1 - \frac{\epsilon}{1 - o(1)}\right)} \\ &= \frac{1}{\log n} n^{1 - \log\left(\frac{1}{1 - \epsilon} + o(1)\right)}. \end{aligned}$$

□

Note 5.2. The Equations (5.2) and (5.3) bound the probability of there being no errors in a block very tightly. If we take the limit when $\frac{m}{n} \rightarrow 0$ we get $\Pr[X_i = 1] = (1 - \epsilon)^m$. This is exactly the probability we would have had, if we had assumed a slightly different (and simpler) model in which the errors are independent: each qubit is bit-flipped with probability ϵ , independently of the other qubits.

We now have a bound on the expected number of blocks with no bit flips which is sublinear in n , but still polynomial. For this to be useful we still have to show that with large probability the number of bit flip free blocks will be close to the expected value, which we do in the following lemma.

Lemma 5.3. *Let N be a random variable representing the number of blocks which contain not bit flips. Then*

$$\Pr[N < (1 - \delta)\mathbf{E}[N]] \leq \frac{1}{\delta^2 \mathbf{E}[N]} \leq \frac{\log n}{\delta^2} n^{-(1 - \log(\frac{1}{1 - \epsilon}))}.$$

Proof. Let us again define random variables X_i which indicate whether block i has a bit flip or not, so $N = \sum_i X_i$. The X_i are unfortunately not independent, so we can not apply a bound like Chernoff. But we can use Chebyshev, which states that

$$\Pr[|N - \mathbf{E}[N]| > \delta \mathbf{E}[N]] \leq \frac{\text{Var}(N)}{\delta^2 (\mathbf{E}[N])^2}. \quad (5.4)$$

Let us calculate the variance of N :

$$\begin{aligned} \text{Var}(N) &= \mathbf{E}[N^2] - \mathbf{E}[N]^2 = \sum_{i,j} \mathbf{E}[X_i X_j] - \mathbf{E}[N]^2 \\ &= \sum_i \mathbf{E}[X_i^2] + \sum_{\substack{i,j \\ i \neq j}} \mathbf{E}[X_i X_j] - \mathbf{E}[N]^2 \\ &= \mathbf{E}[N] + \sum_{\substack{i,j \\ i \neq j}} \mathbf{E}[X_i X_j] - \mathbf{E}[N]^2. \end{aligned} \quad (5.5)$$

$$X_i X_j = \begin{cases} 0 & \text{if either block } i \text{ or block } j \text{ contains a bit flip,} \\ 1 & \text{otherwise.} \end{cases}$$

$$\text{So } \mathbf{E}[X_i X_j] = \Pr[X_i X_j = 1] = \frac{\binom{n-t}{2m}}{\binom{n}{2m}}.$$

$$\begin{aligned} \frac{\sum_{\substack{i,j \\ i \neq j}} \mathbf{E}[X_i X_j]}{\mathbf{E}[N]^2} &= \frac{\frac{n}{m} \left(\frac{n}{m} - 1\right) \binom{n-t}{2m} \binom{n}{m}^2}{\left(\frac{n}{m}\right)^2 \binom{n}{2m} \binom{n-t}{m}^2} \\ &= \left(1 - \frac{m}{n}\right) \frac{n!(n-2m)!(n-t-m)!^2}{(n-t)!(n-t-2m)!(n-m)!^2} \\ &= \left(1 - \frac{m}{n}\right) \frac{n \cdots (n-m+1)(n-t-m) \cdots (n-t-2m+1)}{(n-t) \cdots (n-t-m+1)(n-m) \cdots (n-2m+1)} \\ &\leq \left(1 - \frac{m}{n}\right) \left(\frac{n-m}{n-t-m}\right)^m \left(\frac{n-t-m}{n-m}\right)^m \leq 1. \end{aligned}$$

So $\sum_{\substack{i,j \\ i \neq j}} \mathbf{E}[X_i X_j] \leq \mathbf{E}[N]^2$. By putting this in Equation (5.5) we get $\text{Var}(N) \leq \mathbf{E}[N]$. Which in turn goes in the Chebyshev bound (5.4) and together with the bound for $\mathbf{E}[N]$ from Lemma 5.1 proves this lemma. \square

We have found a bound on the number of bit flip free blocks which will be respected with high probability. So we can make sure that enough qubits will survive the block confirmation step to generate a key of the desired length. But not only bit flip free blocks will pass the test, some errors might slip past as well. So let us now look at the probability of failing to detect bit flips in our block confirmation protocol.

Lemma 5.4. *Let X be the random variable for the vector of differences between Alice's and Bob's measurement when they each measure the block B in the Z basis. Then*

$$\Pr[\text{Conf}_\ell(B) = \text{ok} | X = x] = \begin{cases} 1 & \text{if } x = 0, \\ 2^{-\ell} & \text{otherwise.} \end{cases}$$

Proof. A measurement of $Z \otimes Z$ on every EPR pair would give us the value of X . As $\text{Conf}_\ell(\cdot)$ only uses measurements in the Z basis, these operations commute. The measurement of one does not influence the outcome of the other, and we may do them in the order we want or not do them at all. The lemma can be read as a statement on the probability of a block being accepted, given that a measurement of $Z \otimes Z$ on every EPR pair would have shown a non-zero bit flip vector. Actually any measurement in the Z basis would commute with these operations, so we might as well do a

complete measurement in the Z basis and reduce the problem to classical probability. Let M_A and M_B denote random variables for the outcome of complete measurements in the Z basis of Alice respectively Bob's part of the block B . We have

$$\begin{aligned}
& \Pr [\text{Conf}_\ell(B) = \text{ok} | X = x] \\
&= \frac{\Pr [\text{Conf}_\ell(B) = \text{ok}, X = x]}{\Pr [X = x]} \\
&= \frac{\sum_i \Pr [\text{Conf}_\ell(B) = \text{ok}, M_A = i, M_B = i + x]}{\Pr [X = x]} \\
&= \frac{\sum_i \Pr [\text{Conf}_\ell(B) = \text{ok} | M_A = i, M_B = i + x] \Pr [M_A = i, M_B = i + x]}{\Pr [X = x]}.
\end{aligned} \tag{5.6}$$

Let us express Eve's tampering in terms of bit and phase flips as we have up to now. Alice and Bob's system is in the state

$$\rho_{AB} = \frac{1}{2^n} \sum_{i,j,x,z} p_{x,z} (-1)^{(i+j) \cdot z} |i\rangle_A |i+x\rangle_B \langle j|_A \langle j+x|_B.$$

which is equivalent to the state in (4.4) with Eve's system traced out. Let us perform a complete measurement of the system in the Z basis and obtain $M_A = i$ and $M_B = i + x$, for some $x \neq 0$. The system is then reduced to the pure state

$$|\psi\rangle = |i\rangle_A |i+x\rangle_B.$$

Let us now apply the confirmation protocol, and write the result of the measurements from Step 2 in ancilla systems A' and B'

$$|\psi'\rangle = |i\rangle_A |\lambda i^T\rangle_{A'} |i+x\rangle_B |\lambda(i+x)^T\rangle_{B'}.$$

The confirmation succeeds if $\lambda i^T = \lambda(i+x)^T$, i.e., $\lambda x^T = 0$. But as each λ_i is chosen randomly we have

$$\Pr [\lambda_i x^T = 0 | x \neq 0] = \frac{1}{2} \text{ and } \Pr [\lambda_i x^T = 0 | x = 0] = 1,$$

so

$$\Pr [\lambda x^T = 0 | x \neq 0] = 2^{-\ell} \text{ and } \Pr [\lambda x^T = 0 | x = 0] = 1.$$

In other words, for $x \neq 0$,

$$\Pr [\text{Conf}_\ell(B) = \text{ok} | M_A = i, M_B = i + x] = 2^{-\ell}.$$

By putting this in Equation (5.6) we get for $x \neq 0$,

$$\Pr [\text{Conf}_\ell(B) = \text{ok} | X = x] = \frac{2^{-\ell} \sum_i \Pr [M_A = i, M_B = i + x]}{\Pr [X = x]} = 2^{-\ell}.$$

And the probability for $x = 0$ is obtained similarly. \square

Now that we know the probability of errors getting through the test, let's use Baye's rule to find the expected number of errors in the blocks which pass the test.

Lemma 5.5. *Let the number of the tests performed in the block confirmation protocol be $\ell := \alpha \log n$ for some $0 \leq \alpha \leq 1$, and the blocks have size $m := \log n$. For a random variable X_i representing the number of bit flips in block B_i , the expected ratio of bit flip errors if it passes the confirmation is*

$$\frac{1}{m} \mathbf{E}[X_i | \text{Conf}_\ell(B_i) = \text{ok}] \leq \epsilon n^{-\alpha + \log(\frac{1}{1-\epsilon} + o(1))}.$$

Proof.

$$\begin{aligned} \mathbf{E}[X_i | \text{Conf}_\ell(B_i) = \text{ok}] &= \sum_{x=0}^m x \Pr[X_i = x | \text{Conf}_\ell(B_i) = \text{ok}] \\ &= \frac{\sum_{x=0}^m x \Pr[X_i = x, \text{Conf}_\ell(B_i) = \text{ok}]}{\Pr[\text{Conf}_\ell(B_i) = \text{ok}]} \\ &= \frac{\sum_{x=1}^m x \Pr[\text{Conf}_\ell(B_i) = \text{ok} | X_i = x] \Pr[X_i = x]}{\sum_{x=0}^m \Pr[\text{Conf}_\ell(B_i) = \text{ok} | X_i = x] \Pr[X_i = x]} \\ &= \frac{2^{-\ell} \sum_{x=1}^m x \Pr[X_i = x]}{2^{-\ell} \Pr[X_i \neq 0] + \Pr[X_i = 0]} \\ &\leq \frac{2^{-\ell} \mathbf{E}[X_i]}{\Pr[X_i = 0]} \\ &\leq \frac{\epsilon m 2^{-\ell}}{\left(1 - \frac{\epsilon}{1-o(1)}\right)^m} \\ &= \epsilon m 2^{-\ell + m \log(\frac{1}{1-\epsilon} + o(1))}. \end{aligned}$$

To finish the proof we set $m = \log n$ and $\ell = \alpha \log n$ in the result above. \square

Note 5.6. If we choose $\alpha > \log\left(\frac{1}{1-\epsilon}\right)$ then the ratio of bit flips left after the block confirmation will go to 0 for n going to infinity. So by choosing a larger n we can use a code in Step 11 correcting as little errors as we want, making its effect on the decoding rate negligible.

5.3 Residual phase flips

The block confirmation step reduces drastically the number of bit flips in the qubits kept. But depending on how Eve places the errors, such a strategy could increase the concentration of phase flips. In this section we will calculate this new concentration of phase flips to know what parameter we need for the privacy amplification step.

Lemma 5.7. *Let $\eta := \mu \log n$ be the number of qubits left after the block confirmation step. Let X be the random variable for the number of bit flips in the remaining qubits and Z the random variable for the number of phase flips. Assuming there were exactly t bit and t phase flips in the qubits to start with, then if Eve never flipper both the bit and the phase of the same qubit, the probability of the phase flips given that no bit flips got through the block confirmation protocol has a hypergeometric distribution:*

$$\Pr[Z = z | X = 0] = \frac{\binom{t}{z} \binom{n-2t}{\eta-z}}{\binom{n-t}{\eta}}. \quad (5.7)$$

Proof. $\binom{n-t}{\eta}$ is the number of ways to keep η qubits from the $n-t$ qubits without any bit flips. There are z phase flips in the remaining qubits, which can be chosen in $\binom{t}{z}$ ways. The remaining $\eta-z$ qubits must come from the $n-2t$ qubits with no errors. Put together this gives us the sought distribution. \square

Assuming that Eve never flipped both the bit and phase of a qubit is the worst case: if she had, those phase flips would have been dropped with high probability during the block confirmation step and the concentration of phase flips left would be less.

For convenience let us rename the random variable $Z|X=0$ with Z_0 . The expected value of a hypergeometric distribution is known to be

$$\mathbf{E}[Z_0] = \left(\frac{t}{n-t}\right) \eta = \left(\frac{\epsilon}{1-\epsilon}\right) \eta. \quad (5.8)$$

In reality t is an upper bound on the number of errors, and some bit flips might have got through the block confirmation, so (5.8) is an upper bound on the expected number of phase flips. But for it to be useful, we still have to show that with high probability the number of phase flips is close to the expected value.

Lemma 5.8. *The probability of having more phase flips than the expected value is bounded by*

$$\Pr[Z_0 > (1+\delta)\mathbf{E}[Z_0]] \leq e^{-O(\delta^2\eta)}.$$

Proof. Finding a bound for the hypergeometric distribution given in (5.7) is not simple, because the errors on the qubits are not independent. Instead we will consider the following model: each qubit contains an error with probability 2ϵ . With probability $\frac{1}{2}$ that error is a bit flip respectively a phase flip. We then have the following distribution:

$$\Pr[Z' = z, X' = x] = \binom{\eta}{x+z} (2\epsilon)^{x+z} (1-2\epsilon)^{\eta-x-z} \binom{x+z}{x} \left(\frac{1}{2}\right)^x \left(\frac{1}{2}\right)^z. \quad (5.9)$$

We are interested in the case $X' = 0$, so let us calculate the probability distribution for $Z'_0 := Z|X = 0$ from (5.9):

$$\Pr[Z' = z, X' = 0] = \binom{\eta}{z} (2\epsilon)^z (1 - 2\epsilon)^{\eta-z} \left(\frac{1}{2}\right)^z.$$

$$\begin{aligned} \Pr[X' = 0] &= \sum_z \Pr[Z' = z, X' = 0] \\ &= \sum_z \binom{\eta}{z} \epsilon^z (1 - 2\epsilon)^{\eta-z} \\ &= (\epsilon + 1 - 2\epsilon)^\eta = (1 - \epsilon)^\eta. \end{aligned}$$

$$\begin{aligned} \Pr[Z'_0 = z] &= \frac{\Pr[Z' = z, X' = 0]}{\Pr[X' = 0]} \\ &= (1 - \epsilon)^{-\eta} \binom{\eta}{z} \epsilon^z (1 - 2\epsilon)^{\eta-z} \\ &= \binom{\eta}{z} \left(\frac{\epsilon}{1 - \epsilon}\right)^z \left(1 - \frac{\epsilon}{1 - \epsilon}\right)^{\eta-z}. \end{aligned} \quad (5.10)$$

Equation (5.10) is in fact a binomial distribution with parameters η and $\frac{\epsilon}{1 - \epsilon}$.

Let us now compare the binomial distribution of Z'_0 and the hypergeometric distribution of Z_0 . They have the same expected value, namely $\left(\frac{\epsilon}{1 - \epsilon}\right)\eta$. But the variance of Z'_0 is greater. If $\eta = n - t$ then the Z_0 model will have exactly t phase flips and the variance is 0, whereas in the independent model every qubit could be an error with probability $\frac{t}{n-t}$, so the expected value is t but the variance is much larger. If we reduce the number of qubits kept, the variance of the dependent model will slowly increase. And when we only choose only one qubit out of the n , both models will have the same distribution. So Z_0 varies less from the expected value and

$$\Pr\left[Z_0 > (1 + \delta) \left(\frac{\epsilon}{1 - \epsilon}\right)\eta\right] \leq \Pr\left[Z'_0 > (1 + \delta) \left(\frac{\epsilon}{1 - \epsilon}\right)\eta\right]. \quad (5.11)$$

For a binomial distribution we can use Chernoff's inequality which says that

$$\Pr\left[Z'_0 > (1 + \delta)\mathbf{E}[Z'_0]\right] \leq e^{-\frac{\delta^2\mathbf{E}[Z'_0]}{4}}. \quad (5.12)$$

By putting Equations (5.11) and (5.12) together we prove the lemma. \square

5.4 Maximum error rate

We can now put everything together to calculate the maximum tolerable error rate of this protocol.

Theorem 5.9. *For very large n the EPR-based protocol using block confirmation and privacy amplification given in Figure 5.1 can tolerate errors up to 16.9%.*

Proof. Let μ be the number of blocks which pass the confirmation, so $\eta := \mu \log n$ is the number of qubits left. From Lemma 5.5 and Note 5.6 we know that if we choose $\ell > \log\left(\frac{1}{1-\epsilon}\right) \log n$ the residual bit flip error rate will go to 0. The decoding in Step 12 then reduces the number of qubits from η to $\eta - \mu \log\left(\frac{1}{1-\epsilon}\right) \log n = \eta \left(1 - \log\left(\frac{1}{1-\epsilon}\right)\right)$. From Lemma 5.8 and Theorem 4.3 we know that to be secure the privacy amplification should further reduce the number of qubits by the size of the adversary's memory, which is $\eta h\left(\frac{\epsilon}{1-\epsilon}\right)$. The final number of qubits is

$$\eta \left(1 - \log\left(\frac{1}{1-\epsilon}\right) - h\left(\frac{\epsilon}{1-\epsilon}\right)\right),$$

which is greater than 0 if $\epsilon < 0.169$. □

5.5 Reduction to prepare-and-measure

5.5.1 Prepare-and-measure protocol

The reduction to a prepare-and-measure protocol is similar to what we have done in the previous chapters. The final protocol given in Figure 5.3 resembles BB84 (Figure 3.3) in its structure, with extra steps for error detection and privacy amplification.

Steps 1 to 8 are nearly identical to the previous sections: Alice creates $(4 + \delta)n$ random bits and encodes them in the Z or X basis at random, Bob measures them upon reception in one of the two basis at random, they keep the ones measured in the same basis and compare the values for half of them getting an estimation on the error rate. They now each have a string x and $x \oplus e$.

In Step 3 Alice randomly chooses the parity check matrices which will be used for the block-wise error detection. Each block is of size $\log n$ and ℓ checks are performed, so the random parity check matrices λ_j are of size $(\ell \times \log n)$. The effect on the complete string of performing these block-wise parity checks is equivalent to multiplying the whole string with the parity check matrix H_λ , which contains the λ_j matrices in its diagonal, as illustrated in Figure 5.4.

**QKD protocol: BB84-like with block confirmation and
2-universal privacy amplification**

- 1:** Alice creates $(4 + \delta)n$ random bits.
- 2:** For each bit she creates a qubit in either the Z or X basis according to a random bit string b .
- 3:** Alice chooses $\frac{n}{\log n}$ parity check matrices λ_j of size $(\ell \times \log n)$ uniformly at random, where $\ell = \left(\log\left(\frac{1}{1-\epsilon}\right) + \delta\right) \log n$. She chooses a random $v \in C_\lambda$, which is the code with parity check matrix H_λ , which has the λ_j in its diagonal (see Figure 5.4).
- 4:** Alice sends the qubits to Bob.
- 5:** Bob receives these qubits and publicly announces this fact, and measures each in the Z or X basis at random.
- 6:** Alice announces b .
- 7:** Alice and Bob discard the bits Bob measured in a basis other than b . With high probability there are at least $2n$ left; if not, abort the protocol. Alice decides randomly on a set of $2n$ bits to use and selects n of these to be check bits.
- 8:** Alice and Bob publicly compare their check bits. If more than t of these disagree, they abort the protocol. Alice and Bob are each left with an n bit string.
- 9:** Alice announces a random permutation of the remaining n bits, and Alice and Bob permute the bits accordingly. Alice now has the string x and Bob has $x \oplus e$. They form $\frac{n}{\log n}$ substrings x_j^a respectively x_j^b of size $\log n$, containing their bits in positions $(j - 1) \log n + 1$ to $j \log n$.
- 10:** Alice announces $x \oplus v$. Bob adds this to his string. For each substring j Alice announces λ_j and Bob calculates $\lambda_j v_j^b$. If the outcome is not 0, they drop the substring. Let w and $w \oplus e'$ be the concatenation of Alice respectively Bob's substrings left. With high probability w will have length $\eta \geq (1 - \delta)n^{1 - \log\left(\frac{1}{1-\epsilon}\right)}$. If not, they abort the protocol.
- 11:** Alice chooses a $[\eta, \nu]$ code C_ν with parity check matrix H_ν correcting a small number of errors t' , and announces it to Bob. Let H be the parity check matrix of the code C with the λ_j of the surviving substrings in a diagonal and H_ν appended (see Figure 5.4). Alice chooses a random $z \in C$.
- 12:** Alice sends Bob $w \oplus z$. He adds it to his code obtaining $z \oplus e'$ and corrects it using C_ν , now obtaining z . He decodes it from C obtaining the word y of length $\nu - \mu\ell$.
- 13:** Alice fixes the randomness of a two-universal random function $G: \{0, 1\}^{\nu - \mu\ell} \rightarrow \{0, 1\}^s$, obtaining g , and communicates this to Bob.
- 14:** Alice and Bob compute the final key as $k = g(y)$.

Figure 5.3: A prepare-and-measure QKD protocol which drops blocks in which errors were detected and uses privacy amplification

$$H_\lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{\frac{n}{\log n}} \end{pmatrix} \quad H = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_\mu \\ & & & H_\nu \end{pmatrix}$$

Figure 5.4: Parity check matrices for the codes used in the BB84-like protocol with error detection and privacy amplification (Figure 5.3). H_λ is used in Step 3 and H in Step 11.

In BB84, when Alice holds the string x and Bob has $x \oplus e$, Alice sends him $x \oplus v$ which he adds to his string to be left with a codeword v and an error e , on which he can perform error correction. Here in Step 10 we do exactly the same. The parity check is equivalent to projecting their states in a random code. As what is important is the difference between Alice's and Bob's syndrome, Alice can choose a codeword v from the random code C_λ , which as the parity check matrix H_λ , send Bob $x \oplus v$, who adds it to his string. He can then measure the error syndrome block-wise, using the λ_j in H_λ . But instead of correcting the errors, they drop the substrings containing them. The now each have the strings w and $w \oplus e$, which are shorter than the original ones, but contain less errors.

In Steps 11 and 12 they correct the errors remaining. Alice chooses a code C_ν with parity check matrix H_ν correcting a small number of errors t' . The combined effect of the error detection and error correction projects their states into a code, which has a parity check matrix consisting of the concatenation of the error detection parity check matrix (for the surviving blocks) and the error correcting parity check matrix, i.e., H with the λ_j of the surviving blocks in the diagonal and H_ν appended, as illustrated in Figure 5.4.

So to perform the actual error correction, Alice can choose a random codeword z in C , which has the parity check matrix H , and send Bob $w \oplus z$, who adds it to his string and measures the syndrome with H_ν , corrects it and decodes from C . As the parities of the λ_j are known, there is no need for Bob to measure them again, H_ν is enough.

The final step is privacy amplification with a two-universal function.

5.5.2 Reduction

The proof of equivalence between the EPR protocol (Figure 5.1) and the prepare-and-measure one (Figure 5.3) is slightly more complicated than for the reduction in the previous chapter, because we cannot apply directly the Shor-Preskill proof as we did then. The first step of that proof gets rid of the EPR pairs by encoding the state Alice sends to Bob in a stabiliser code.

But in our new EPR protocol we do not know which blocks will survive the block confirmation in Step 10, so Alice cannot encode the state in the code used in Step 11, because she does not know which qubits need to be encoded. We will however argue that the same reduction can be applied, and that it is secure.

Let us go through the EPR-based protocol from Figure 5.1 and the BB84-like protocol in Figure 5.3 we would like to reduce it to, and show that from Eve's point of view there is no difference, so the two are equivalent. As explained in Section 5.1, the block confirmation done in the sub-protocol Conf_ℓ in Figure 5.2, is equivalent to measuring the syndrome of a random code. We have shown in Chapter 3 that this is equivalent to projecting the EPR pairs onto a code with a random syndrome, so Alice might as well choose the code/syndrome herself. We then showed that Bob could first measure before decoding, so Alice does not even need to encode. She can wait until she knows which qubits Bob measured in the right basis, then calculate which coset her non-noisy string belongs to, and tell it to Bob so he can correct and decode. We do exactly the same here, giving the Steps 1 to 10 of the new protocol.

As explained above, the difficulty comes from the error correction in Step 11 of Figure 5.1. Let us suppose for an instant, that Alice knows from some oracle which qubits will survive the block confirmation. She could then encode the correct qubits from the start in the code used in Step 11 and run an equivalent protocol with no EPR pairs. Eve would not see any difference between this oracle-aided protocol and the EPR-based one, since surviving and removed blocks are both equivalent to completely mixed states. Alice could then further reduce the protocol as we have done up to now, by not encoding the state in a code anymore, but by sending random BB84 qubits, which is what we do in Figure 5.3. This new protocol is indistinguishable by Eve, therefore equivalent to the oracle-aided one and equivalent to the EPR-based one. But she does not treat qubits which survive the test and those which fail differently any more (just as she does not treat qubits who serve as test bits and those who are used to generate the key any differently). So for this last protocol, she does not need the help of the oracle anymore. We finally conclude that our BB84-like protocol and the EPR-based one are equivalent.

Chapter 6

Conclusion

6.1 Advantages

The tolerable error rate of the protocol given in Figure 5.3, 16.9%, is slightly worse than the result from [4], which achieves 18.9%. But it has a few other advantages.

First of all, the simplicity of the analysis. We have two separate and clear error detection and privacy amplification steps. We do not alternate between several phases of one and then the other, which allows us to derive the error rate bound analytically and not numerically.

Secondly, we only perform error correction for a very small number of errors, with a ratio going to zero. Error correction is a heavy operation with an exponential complexity, so performing it for a large n is not practically realisable. But our scheme detects nearly all bit flips and only leaves a number in $o(n)$, so a very simple error correcting code may be used.

Moreover, as the privacy amplification step we used to compensate for phase flips is completely decoupled from the error correction/detection, any interactive protocol can be used to correct bit flips. We are not restricted to bit flip correction measurement operators which commute with the phase flip correction operators. Thus generalising these ideas to any interactive protocols using linear codes is straightforward.

6.2 Further improvements

The bit flip detection performed in Chapter 5 was pretty successful: it allowed us to increase the tolerable error rate from 11% to 16.9%. So it would be natural to try to apply it to phase flip detection. The idea is pretty simple: perform a block confirmation protocol for the bit flips, apply Hadamards to every qubit, and start again with the bloc confirmation for the phase flips. Such a protocol would be successful up to an error rate of 25%, but it is unfortunately not reducible to a prepare-and-measure protocol. In such a

protocol, we do not perform the measurements in the X basis. But these would be essential to identify which blocs have phase flip errors and should be dropped.

The size of Eve's memory calculated in Theorem 4.3 was an upper bound based on the number of phase flips present before the bit flip detection. But when we check parities, we reduce the dimension of our space and this might also reduce the number of phase flips. So by considering the quantity present before the error detection, we consider the worst case. Our calculations showed that the dimension reduction would only affect the size of Eve's memory if it was greater than that of Alice and Bob. And is such a case no key could be extracted anyway. However, if we take into account that the reduction is done block-wise, it could be possible that for a specific block Eve has a lot of information due to phase flips, and some of this is lost during the bit flip detection, but all together she has little enough information for it to be possible for Alice and Bob to produce a secure key.

Instead of considering blocks of size $\log n$, we could take blocks of a constant size, such as done in [4]. The error detection would not reduce the bit flip ratio to 0, because this is only true asymptotically. But we could perform fewer parity checks and then use an error correcting code correcting a higher bit flip rate. But optimising over the number of parity checks performed on blocks of constant size has shown that the tolerable error rate is lower than what we achieved using blocks of size $\log n$.

But with a block of constant size the evolution of the phase flips when we correct bit flips is more complicated, because we cannot consider the asymptotic case, and we have not done that analysis in detail. There might still be room for improvements, and that could explain why [4] achieves a slightly better bound.

Maybe a further improvement could be achieved by introducing noise. In [7, 14] it has been shown that with such a preprocessing step the protocol can resist to high eavesdropping rates. But Alice and Bob would not share entangled states any more, and our proof would have to be adapted.

Bibliography

- [1] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed state entanglement and quantum error correction. *Physical Review A*, 54:3824–3851, 1996.
- [2] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179, 1984.
- [3] D. Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, Pasadena, CA, 1997.
- [4] D. Gottesman and H.-K. Lo. Proof of security of quantum key distribution with two-way classical communications. *IEEE Trans. Inf. Theory*, 49:457, 2003.
- [5] A. S. Holevo. Statistical problems in quantum physics. In *Proceedings of the second Japan-USSR symposium on probability theory*, volume 330 of *Lecture Notes in Mathematics*, pages 104–119. Springer-Verlag, 1973.
- [6] Robert König, Ueli Maurer, and Renato Renner. On the power of quantum memory. *IEEE Transactions on Information Theory*, 51(7):2391–2401, 2005. eprint archive: <http://arxiv.org/abs/quant-ph/0305154>.
- [7] Barbara Kraus, Nicolas Gisin, and Renato Renner. Lower and upper bounds on the secret key rate for qkd protocols using one-way classical communication. <http://arxiv.org/abs/quant-ph/0410215>, October 2004.
- [8] Hoi-Kwong Lo. Method for decoupling error correction from privacy amplification. *New Journal of Physics*, 5:36.1–36.24, 2003.
- [9] Hoi-Kwong Lo and H F Chau. Unconditional security of quantum key distribution over arbitrarily long distances. *Science*, 283:2050, 1999.
- [10] Hoi-Kwong Lo, H. F. Chau, and M. Ardehali. Efficient quantum key distribution scheme and a proof of its unconditional security. *Journal of*

Cryptology: the journal of the International Association for Cryptologic Research, 18(2):133–165, April 2005.

- [11] Florence J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1977.
- [12] Dominic Mayers. Unconditional security in quantum cryptography. *Journal of the ACM*, 48(3):351–406, 2001.
- [13] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge university press, 2000.
- [14] Renato Renner, Nicolas Gisin, and Barbara Kraus. An information-theoretic security proof for qkd protocols. <http://arxiv.org/abs/quant-ph/0502064>, February 2005.
- [15] Renato Renner and Robert König. Universally composable privacy amplification against quantum adversaries. In *Theory of Cryptography — TCC 2005*, pages 407–425, 2005.
- [16] P. W. Shor and J. Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, 85:441–444, July 2000.