SECURITY AND PRACTICAL MODELS FOR QUANTUM KEY DISTRIBUTION

by

Juliette van Mil

To obtain the degree of Bachelor of Science in Applied Mathematics and Applied Physics at the Delft University of Technology

Student number:	4775899
Supervisors:	Dr. D. Elkouss Coronas
	Dr. M. Möller
Committee members:	Prof. Dr. A. F. Otte
	Dr. J. L. A. Dubbeldam
Supporting teammembers:	Dr. J. A. Slater
	K. Goodenough

August 18, 2021

ACKNOWLEDGEMENTS

I would like to thank my supervisors Dr. David Elkouss and Dr. Matthias Möller for all their advice, support and feedback throughout the whole course of researching, modelling and writing this report. I also want to thank Dr. Joshua Slater for his endless passion and enthousiasm for quantum key distribution. Thank you for checking in on me every week and being happy to answer all the questions I had. I definitely also want to thank Kenneth Goodenough for always being ready to help me with understanding everything I learned these past months and answering all my questions. Thank you for asking questions back and checking my notion on both the physical and mathematical part of concepts.
I want to thank my parents for their support and for always welcoming me with open arms when I needed a weekend off to wind down a little. And lastly, but very importantly, I want to thank Jasper for his patience and his support. I know it has been a stressful past few weeks and it must not have been very nice to live with me in these times, but still you were always there to support me and help me brainstorm, when I was stuck on a problem or hard decision.

ABSTRACT

This report reviews two quantum key distribution (QKD) protocols: the BB84 protocol and the measurement device independent (MDI) QKD protocol. The goal of this report is to recreate the security proof of the BB84 protocol, to generate the secret key rate for a practical application of the BB84 protocol, both with and without decoy states, and to review the MDI-QKD protocol and look at the advantages it has for practical QKD. The proof security of the BB84 protocol is done by designing an equivalent theoretical protocol and proving its security by bounding the information of the eavesdropper to an exponentially small number. The best distance over which secret key rate can be generated with the practical model of the BB84 protocol that is presented in this report, is 165 km. This is achieved by employing decoy states and is more than three times as far as the model can achieve without decoy states, which gave a distance of 52 km at best. The zero distance key rate of the model with decoy states is $R = 1.21 \cdot 10^{-2}$ bits per pulse (bpp), at best. This is an order of magnitude larger than $R = 1.02 \cdot 10^{-3}$ bpp, which is what the model without decoy states could produce. The decoy state method is therefore an improvement for practical QKD. By reviewing the MDI-QKD protocol it can be concluded that because it eliminates the measurement device side channels, it is very useful for practical QKD, since it makes security analysis simpler and more precise. It also necessarily employs decoy states, which improves the secret key rate. MDI-QKD therefore is a promising protocol to use for future quantum communications.

CONTENTS

1

INTRODUCTION

In the world that we know today, we are almost always making use of technology to communicate with each other and share information. For all this communication and transfer of information, we use encryptions to make sure that only the intended receiver and sender can access the information being handled. No third party should have a notion on their private exchange of information.
Popular public key cryptography is employed in most commercial communication services. This kind of cryptography uses pairs of keys, public keys that may be known to anyone and private keys that is known only by the owner. Current developments in quantum computers pose a threat to this kind of key distribution. Algorithms on quantum computers will be able retrieve information about the private key and therefore compromise public key cryptography (Buchanan and Woodward, 2017, and Mavroeidis et al., 2018). Therefore in the very near future there will be a demand for a new way of distributing keys for encrypting messages.
But where quantum will be able to compromise cryptography, it is also able to propose another way of securely conveying messages. The field of establishing a secure communication link by using transmission of quantum states, is known as Quantum Key Distribution.
Quantum Key Distribution (QKD) is a way to securely distribute a key between two users. The shared key can then be used to encrypt the messages that are to be conveyed.
In the past decades there have been many developments in the field of quantum cryptography. A lot of progress has been made in both theory and experiments (Gisin et al., 2002).
In more recent years there have also been successful experiments of practical QKD system distributing key over distances that are not unusual for commercial applications (Pirandola et al., 2019). Also has there been a lot of progress in proving the security of practical QKD protocols, bringing us even closer to applications of QKD in real life (Scarani et al., 2009 and Xu et al., 2020).

In this report we will review two of the most interesting protocols of QKD: the BB84 protocol and the Measurement-Device Independent QKD (MDI-QKD) protocol. The goal of this thesis is threefold. We will recreate the full security proof of the BB84 protocol in idealised circumstances. We will generate the secret key rate for a model of practical QKD that uses the BB84

protocol. We will do this with and without the use of the decoy state method; a method that should boost the secret key rate, and compare the results. Lastly, the protocol of MDI-QKD is reviewed and we will look at the advantages it has for practical QKD.

In Chapter 2 the four most important preliminaries are discussed which will provide the foundation of subjects in this report. First, some important concepts of quantum mechanics and quantum information are introduced. Second, we discuss the definitions of different entropies from information theory. Then we will explain the purpose and working of cryptography. And lastly the theory of linear optimisation models will be introduced which describes models to optimise variables that are subject to linear (in)equality constraints.
In Chapter 3 we introduce the concept of quantum key distribution and give the definition of a secure QKD protocol. Afterwards we will discuss the BB84 protocol with a thorough description of the different steps of the protocol. Next, the full security proof of ideal BB84 will be given, using concepts from quantum mechanics, information theory, coding theory and probability theory.
Chapter 4 focuses on practical QKD. In this chapter we will derive a model based on BB84 that can generate secure key while taking into account the imperfect workings of practical devices. The secret key rate will be derived for two versions of the model; one that assumes the worst case of errors, as to ensure security and one that employs decoy states, a smart source setting that provides linear constraints on the variables of the model. In the last part of this chapter, the theory of linear optimisation models will be used to derive the secret key rate for decoy state QKD.
Then in Chapter 5 we will introduce another protocol, that of MDI-QKD. The working of the protocol will be explained and we will discuss why MDI-QKD is very useful as a practical quantum key distribution protocol.

This thesis has been written as part of the double bachelor's degree Applied Mathematics and Applied Physics at the Delft University of Technology, under supervision of Dr. M. Möller and Dr. D. Elkouss.

2

PRELIMINARIES

In the four sections in this chapter we introduce the groundwork for the subjects of this report. First we will introduce several important concepts from quantum mechanics and quantum information. Then from information theory the entropy is described and concepts related to entropy. We will also give a small introduction to cryptography and the definition of a key distribution protocol. Lastly an insight will be given in the workings of a linear optimisation model.

2.1. FORMALISMS OF QUANTUM INFORMATION

In quantum theory we base our calculations on two mathematical constructs: wave functions and operators. These are enough to describe a quantum mechanical system. The state of the system is represented by its wave function. Wave functions are (complex) vectors that live in a Hilbert space.

Definition 2.1. A complete, normed vector space $\mathcal{H}$ with inner product $\langle \cdot, \cdot \rangle$, such that it induces the norm $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$, is called a Hilbert space.

By complete, we mean that in this vector space, every converging sequence converges to an element in the vector space. We usually take the set $\mathbf{C}^n$ as our Hilbert space. In quantum information the 'bra-ket' notation is often used to denote vectors. In this notation a vector or wave function ψ is denoted by $|\psi\rangle$, this is the 'ket'. Its adjoint, the complex conjugate ψ^*, that lives in the dual space, is denoted by $\langle\psi|$, this is the 'bra'. In this notation, inner products will be written as $\langle \cdot | \cdot \rangle$.
Let us for an example look at the Hilbert space that is given by $\mathbf{C}^2$. This vector space is spanned by the basis $|0\rangle$ and $|1\rangle$, which have the following computational coordinates:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.1}$$

Any vector in $\mathbf{C}^2$ can be written as a linear combination of $|0\rangle$ and $|1\rangle$. $|0\rangle$ and $|1\rangle$ are orthogonal vectors, as their inner product $\langle 0|1\rangle$ equals zero. Since their norm is also equal to 1, we

call them orthonormal. The adjoint of $|0\rangle$ for example is given as

$$\langle 0| = \begin{pmatrix} 1 & 0 \end{pmatrix}. \tag{2.2}$$

Another spanning set for the vector space $\mathbf{C}^2$ is the basis $|+\rangle$ and $|-\rangle$ given by

$$|+\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}; \qquad |-\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{2.3}$$

These vectors are also orthonormal.

The two dimensional vector space $\mathbf{C}^2$ is used to describe a single qubit. If we want to describe simultaneously more qubits, multiparticle systems, we will need a larger vector space. The tensor product is a way of putting vector spaces together to form larger vector spaces. Suppose that V and W are m and n dimensional Hilbert spaces, respectively. Then $V \otimes W$ is a mn dimensional Hilbert space, where $\otimes$ is the tensor product. The elements of $V \otimes W$ are linear combinations of tensor products $|v\rangle \otimes |w\rangle$ of elements $|v\rangle$ of V and $|w\rangle$ of W. We will also use the abbreviated notation $|vw\rangle$ for $|v\rangle \otimes |w\rangle$. The matrix representation of the tensor product is known as the Kronecker product, which is explained for example by Nielsen and Chuang, 2002. Lastly, the notation $|\psi\rangle^{\otimes k}$ means $|\psi\rangle$ tensored k times with itself. So for example $|\psi\rangle^{\otimes 2} = |\psi\rangle \otimes |\psi\rangle$.

A linear operator $A : V \mapsto W$ is a function which is linear in its inputs: $A\left(\sum_i a_i |v_i\rangle\right) = \sum_i a_i A(|v_i\rangle)$. To denote $A(|v\rangle)$ we will write in the rest of this report simply $A|v\rangle$. A linear operator can also map from V to V. We say that the linear operator is defined on V. The identitiy operator I is a linear operator that can be defined on any vector space V by the equation $I|v\rangle = |v\rangle$ for all vectors $|v\rangle$. Operators can be described with matrices, defined for a fixed basis. For example, the identity operator in $\mathbf{C}^2$ is given as

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.4}$$

In this vector space we will define three other very important matrices, the Pauli matrices,

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.5}$$

All of the Pauli matrices have eigenvalues -1 and 1. The Pauli matrices are all operators that can be applied to single qubits. The X operator, for example, is the bit flip operator since it flips $|0\rangle$ to become $|1\rangle$ and vice versa.
We will introduce one more operator on one qubit: the Hadamard operator,

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2.6}$$

The Hadamard operator for example maps $|0\rangle$ to $|+\rangle$ and maps $|1\rangle$ to $|-\rangle$, and maps them back again, too. We can also apply the Hadamard transform on n qubits using the tensor product, this operator is then written as $H^{\otimes n}$.

Linear operators also have an adjoint or Hermitian conjugate. The adjoint of the linear operator A on a Hilbert space $\mathcal{H}$ is $A^\dagger$ such that for all vectors $|v\rangle, |w\rangle \in \mathcal{H}$,

$$\langle v|Aw\rangle = \langle vA^\dagger|w\rangle. \tag{2.7}$$

An operator A whose adjoint is A, is called a Hermitian or self-adjoint operator.
For the matrix product $\langle v|Aw\rangle$ we will also use the notation $\langle v|A|w\rangle$. The two notations mean exactly the same.
An operator A is defined as normal if $AA^\dagger = A^\dagger A$ holds. Clearly, Hermitian operators are normal.
Normal linear operators can be represented in a different, useful way, which makes use of the inner product. This is the outer product representation.

Definition 2.2. (Nielsen and Chuang, 2002) Let $|v\rangle$ be a vector in a Hilbert space V and $|w\rangle$ be a vector in a Hilbert space W. Then we define $|w\rangle\langle v|$ to be the linear operator from V to W whose action is defined by $(|w\rangle\langle v|)(|v'\rangle) = |w\rangle\langle v|v'\rangle = \langle v|v'\rangle|w\rangle$.

In this notation we can present $I = \sum_i |i\rangle\langle i|$, where $|i\rangle$ is an orthonormal basis of the Hilbert space V. We can generalise this representation for an arbitrary normal operator A. Let the vectors $|i\rangle$ form an orthonormal set of eigenvectors for A, with corresponding eigenvalues λ_i. The diagonal representation of the operator A on a vector space V is then $A = \sum_i \lambda_i |i\rangle\langle i|$. An operator is diagonalisable if it has a diagonal representation. All Pauli matrices are diagonalisable. For example, the Pauli Z matrix may be written as

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|, \tag{2.8}$$

where the diagonal representation is with respect to the two orthonormal vectors $|0\rangle$ and $|1\rangle$. Diagonal representations are also known as orthonormal decompositions.
To summarise this property of normal operators, we have the following theorem.

Theorem 2.1. (Nielsen and Chuang, 2002, *Spectral decomposition*) Any normal operator M on a vector space V is diagonal with respect to some orthonormal basis for V. Conversely, any diagonalisable operator is normal.

Matrices can also be unitary. A matrix U is said to be unitary if $U^\dagger U = I$. Note that U also satisfies $UU^\dagger = I$, so U is normal and has a spectral decomposition. In quantum mechanics we use unitary matrices to describe operations on quantum states. This is because all the eigenvalues of unitary matrices have the absolute value of 1 and therefore operations with unitary matrices preserve the norm of the state and they preserve the inner product on a Hilbert space. Intuitively this can be interpreted as that unitary operations preserve the 'length' and 'angles' of quantum states.
We now introduce an important class of Hermitian operators known as the projectors.

Definition 2.3. (Nielsen and Chuang, 2002) Suppose W is a k-dimensional vector subspace of the d-dimensional vector space V. Using the rules of linear algebra, it is possible to construct an orthonormal basis $|1\rangle, ..., |d\rangle$ for V such that $|1\rangle, ..., |k\rangle$ is an orthonormal basis for W. By definition,

$$P = \sum_{i=1}^{k} |i\rangle\langle i| \tag{2.9}$$

is the projector onto the subspace W. P is Hermitian.

Projectors are used to do projective measurements, which we will introduce in the next section.
Another important subclass of Hermitian operators are the positive operators.

Definition 2.4. (Nielsen and Chuang, 2002) A positive operator A is defined to be an operator such that for any vector $|v\rangle$, $\langle v|Av\rangle$ is a real, non-negative number. If $\langle v|Av\rangle$ is strictly greater than zero for all $|v\rangle \neq 0$, then A is positive definite.

We can also define functions for operators. Given a function f from the complex numbers to the complex numbers, it is possible to define the corresponding matrix function on some subclass of matrices by the following construction. For example, if $A = \sum_a a|a\rangle\langle a|$ is a spectral decomposition for some normal operator A, we define $f(A) = \sum_a f(a)|a\rangle\langle a|$. This procedure can be used to define different functions on operators; for example the exponential of a normal operator or the square root of a positive operator (Nielsen and Chuang, 2002).

An important matrix function that we will use on operators is the trace of a matrix. The trace of A is defined to be the sum of its diagonal elements,

$$\mathrm{Tr}(A) = \sum_i A_{ii}. \tag{2.10}$$

The commutator between two operators A and B is defined to be

$$[A, B] = AB - BA \tag{2.11}$$

If $[A, B] = 0$, that is, $AB = BA$, then we say that A commutes with B.
To do measurements in quantum mechanics, we use observables. Observables are hermitian operators, that, when applied to quantum states, return the eigenvalue of the operator that corresponds to the measured quantum state. How measurements work exactly will be explained in the next section.
When two observables commute the measurement of one observable, has no effect on the result of measuring the other observable. Therefore, for observables that commute, it is not necessary to specify the order in which they are measured.

QUANTUM OPERATIONS

We have defined many operators on vector spaces, but we also want to model quantum operations, Φ, between quantum states, ψ. To communicate with quantum information we need to transmit states. To do this we use quantum channels. Quantum channels are completely positive, trace preserving (CPTP) maps. Quantum channels are thus a type of quantum operation that transmits information. This is why it needs to be trace preserving; to keep the information intact, so to say. Quantum channels need to be completely positive, since they work on density matrices, which are positive. Stinespring, 1954, came up with a theorem that characterises completely positive maps. We will first state the theorem and then discuss its aspects.

Theorem 2.2. (Stinespring, 1954) Let $A, B \subset \mathbb{B}(\mathcal{H})$, both A, B unital and let $\Phi : A \to B$ be a completely positive map. Then there exists a Hilbert space K, a bounded map $B : \mathcal{H} \to K$ and a $\dagger-$ homomorphism $\pi : A \to \mathbb{B}(K)$ such that,

$$\Phi(\psi) = B^{\dagger}\pi(\psi)B. \tag{2.12}$$

Moreover, $\|B\|^2 \leq \|\Phi\|$.

The full proof of this theorem can be found in Stinespring, 1954.
Here $\mathbb{B}(\mathcal{H})$ is the set of bounded operators on $\mathcal{H}$ and unital is 'containing an identity element'. A $\dagger-$homomorphism, in the original paper a $*-$homomorphism, $x \mapsto x^*$, is an involution, meaning that it is a function that is its own inverse. This implies that $\pi(\psi)^* = \pi(\psi^*)$ for all $\psi \in \mathbb{B}(\mathcal{H})$, or in our notation $\pi(\psi)^{\dagger} = \pi(\psi^{\dagger})$.
Stinespring's theorem basically states that all quantum channels can be modelled by unitary evolution after a suitable ancilla is applied to the system. This allows us to do calculations much easier. Equation 2.12 is known as the decomposition of a quantum channel.

2.1.1. DESCRIBING QUANTUM STATES

Suppose a quantum system whose state is not completely known. It could for example be in one of a number of states $|\psi_i\rangle$, where i is an index, with respective probabilities p_i. We call $\{p_i, |\psi_i\rangle\}$ an ensemble of pure states. To describe an ensemble of quantum states, we will introduce the density operator ρ. It is defined by Nielsen and Chuang, 2002 by the equation

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|. \tag{2.13}$$

A state is called pure if the density operator ρ can be written as $\rho = |\psi\rangle\langle\psi|$ for some $\psi \in \mathcal{H}$ with $\|\psi\| = 1$. The characterisation of ρ is given in the following theorem:

Theorem 2.3. (Nielsen and Chuang, 2002, *Characterisation of density operators*) An operator ρ is the density operator associated to some ensemble $\{p_i, |\psi_i\rangle\}$ if and only if it satisfies the conditions:

1. (Trace condition) ρ has trace equal to one;
2. (Positivity condition) ρ is a positive operator.

Now that we have the density operator to describe quantum states, we introduce the four postulates of quantum mechanics as stated by Nielsen and Chuang, 2002.

Postulate 1. Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the *state space* of the system. The system is completely described by its *density operator*, which is a positive operator ρ with trace one, acting on the state space of the system. If a quantum system is in the state ρ_i with probability p_i, then the density operator for the system is $\sum_i p_i \rho_i$.

Postulate 2. The evolution of a *closed* quantum system is described by a *unitary transformation.* That is, the state ρ of the system at time t_1 is related to the state ρ' of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2,

$$\rho' = U\rho U^{\dagger}. \tag{2.14}$$

Postulate 3. Quantum measurements are described by a collection $\{M_m\}$ of *measurement operators*. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system in ρ immediately before the measurement then the probability that result m occurs is given by

$$p(m) = \mathrm{Tr}\Big(M_m^\dagger M_m \rho\Big), \tag{2.15}$$

and the state of the system after the measurement is

$$\frac{M_m \rho M_m^\dagger}{\mathrm{Tr}\Big(M_m^\dagger M_m \rho\Big)}. \tag{2.16}$$

The measurement operators satisfy the *completeness equation*,

$$\sum_m M_m^\dagger M_m = I. \tag{2.17}$$

Postulate 4. The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through n, and system number i is prepared in the state ρ_i, then the joint state of the total system is $\rho_1 \otimes \rho_2 \otimes \ldots \otimes \rho_n$.

These postulates describe how to connect the physical world of quantum mechanics with mathematical descriptions and operations.

PROJECTIVE MEASUREMENTS

The measurements of quantum states that will be treated in this report are the projective measurements. Projective measurements can be seen as a special case of Postulate 3. As stated by Nielsen and Chuang, 2002:

Projective measurements A projective measurement is described by an observable, M, a Hermitian operator on the state space of the system being observed. The observable has a spectral decomposition,

$$M = \sum_m m P_m, \tag{2.18}$$

where P_m is the projector onto the eigenspace of M with eigenvalue m. The possible outcomes of the measurement correspond to the eigenvalues, m, of the observable. Upon measuring the state $|\psi\rangle$, the probability of getting result m is given by

$$p(m) = \langle \psi | P_m | \psi \rangle. \tag{2.19}$$

Given the outcome m occurred, the state of the quantum system immediately after the measurement is

$$\frac{P_m |\psi\rangle}{\sqrt{p(m)}}. \tag{2.20}$$

We will look at an example of projective measurements on a single qubit, a quantum particle. We take the measurement of the observable Z given in Equation 2.5. This has eigenvalues +1 and -1 with corresponding eigenvectors $|0\rangle$ and $|1\rangle$, respectively. Thus, for example, measurement of Z on the state $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ gives the result +1 with probability $\langle\psi|0\rangle\langle 0|\psi\rangle = 1/2$, and similarly the result -1 with probability 1/2.
But if we measure a state that is precisely one of the eigenvectors of the observable, for example, a measurement of Z on the state $|0\rangle$, we will get the result +1 with probability $\langle 0|0\rangle\langle 0|0\rangle = 1$. The result -1 will automatically have probability 0.

In conclusion, projective measurements only give a deterministic outcome when we measure a qubit that is in an eigenvector state of our observable. Otherwise, we will get a probabilistic outcome.

MEASURING THE DISTANCE BETWEEN QUANTUM STATES

To find out how close two quantum states are we define the quantum generalisations of trace distance and fidelity.

Definition 2.5. (Nielsen and Chuang, 2002) The trace distance between quantum states ρ and σ is defined as

$$D(\rho,\sigma) = \frac{1}{2}\operatorname{Tr}\left\|\rho-\sigma\right\|, \tag{2.21}$$

Where we define $\|A\| = \sqrt{A^\dagger A}$ to be the positive square root of $A^\dagger A$. Trace distance forms a metric on density operators.

A second measure of distance between quantum states is the fidelity.

Definition 2.6. (Nielsen and Chuang, 2002) The fidelity of states ρ and σ is defined to be

$$F(\rho,\sigma) = \operatorname{Tr}\sqrt{\rho^{1/2}\sigma\rho^{1/2}}. \tag{2.22}$$

The fidelity is not a metric on density operators.
If we want to calculate the fidelity between a pure state $|\psi\rangle$ and arbitrary state ρ, then from Equation 2.22 it follows that

$$F(|\psi\rangle\langle\psi|,\rho) = \operatorname{Tr}\sqrt{\langle\psi|\rho|\psi\rangle|\psi\rangle\langle\psi|} = \sqrt{\langle\psi|\rho|\psi\rangle}. \tag{2.23}$$

Fidelity is equal to the square root of the overlap between the two states.

2.2. INFORMATION THEORY

The Shannon entropy, or just entropy, is a measure of the average amount of uncertainty of an event that is being observed. Consequently, it is also a measure of the amount of new information that is gained, when an event actually occurs. If the entropy is zero, there is no uncertainty about the outcome of the event, and therefore the event will not supply new information. When the entropy is maximal then the event is highly uncertain and the outcome will be new and unexpected.

Suppose we have a random variable X that has a certain probability distribution $p_1,\ldots,p_n$. We have the following definition of the Shannon entropy.

Definition 2.7. (Nielsen and Chuang, 2002) The Shannon entropy of X measures the amount of uncertainty about X before we learn its value. The Shannon entropy may be written as a function of the probability distribution

$$H_S(X) = H(p_1, \ldots, p_n) = -\sum_x p_x \log p_x, \tag{2.24}$$

where 'log' indicates the logarithm with base two and we define $0 \log 0 = 0$.

Throughout the rest of this report, log will always be the logarithm with base 2.
The Shannon entropy of a two-outcome random variable is called binary entropy.

Definition 2.8. (Nielsen and Chuang, 2002) The binary entropy is defined as

$$H(p) = -p \log p - (1-p) \log\big(1-p\big), \tag{2.25}$$

where p and $1-p$ are the probabilities of the two outcomes of a random variable.

In Figure 2.1 the binary entropy is plotted against probability p.
$H(p)$ is zero for $p = 0$ and $p = 1$. This is what we expect, since for these probabilities there is no uncertainty about the outcome of the random variable. For a two-outcome random variable, it is also easy to see that the binary entropy is maximal for $p = 1/2$. When the possibility of both outcomes is equal, the uncertainty of the outcome is the highest.

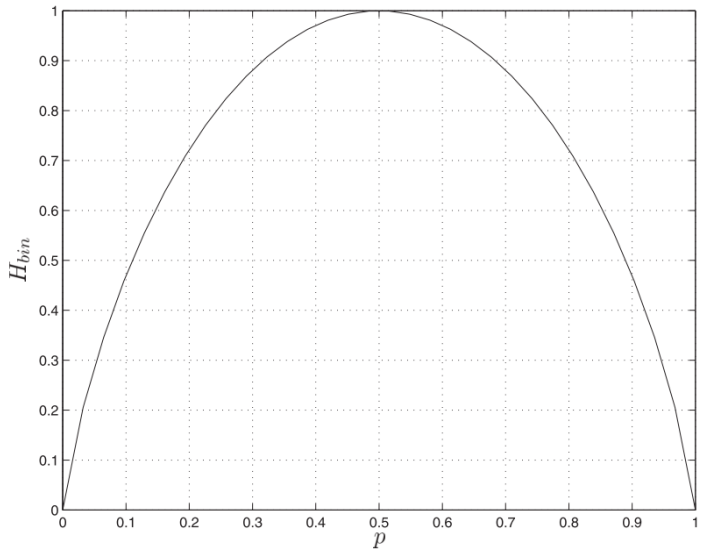

Figure 2.1: Nielsen and Chuang, 2002. The binary entropy function H(p) of probabilities p. The binary entropy attains its maximum at 1/2.

For the Shannon entropy H_S we will write H, just as the binary entropy, since context will make clear which one we are talking about.

Now suppose that we have two random variables X and Y. We want to relate the information content of X to that of Y. To do so, we introduce a few concepts.

Definition 2.9. (Nielsen and Chuang, 2002) The joint entropy of a pair of random variables measures the total uncertainty about the pair (X, Y) and is defined as

$$H(X,Y) = -\sum_{x,y} p(x,y) \log p(x,y) \tag{2.26}$$

with $p(x, y)$ the joint probability distribution of X and Y.

Assuming that we know Y, so we know the $H(Y)$ bit of information about the pair (X, Y), then the remaining uncertainty about the pair is the entropy of X conditional on knowing Y. This conditional entropy is defined by

$$H(X|Y) = H(X,Y) - H(Y). \tag{2.27}$$

Lastly, we have the mutual information content of X and Y which measures how much information X and Y have in common.

Definition 2.10. (Nielsen and Chuang, 2002) The mutual information of X and Y is given by

$$H(X:Y) = H(X) + H(Y) - H(X,Y). \tag{2.28}$$

This is the information we have on X plus the information we have on Y, but subtracted the information we counted twice, namely the information which is common to X and Y.

We can relate conditional entropy and mutual information with the equality $H(X:Y) = H(X) - H(X|Y)$.

Note that the Shannon entropy measures the uncertainty that is associated with classical probability distributions. A generalisation of the Shannon entropy for quantum states, is known as the Von Neumann entropy.

Definition 2.11. (Nielsen and Chuang, 2002) The Von Neumann entropy of a quantum state ρ is given by the formula

$$S(\rho) = -\operatorname{Tr}\big(\rho \log \rho\big). \tag{2.29}$$

If λ_x are the eigenvalues of ρ then Von Neumann's definition is expressed as

$$S(\rho) = -\sum_x \lambda_x \log \lambda_x \tag{2.30}$$

where again we define $0 \log 0 = 0$.

2.3. CRYPTOGRAPHY

Symmetric secret key cryptography describes algorithms that require the use of the same key for encrypting and decrypting a secret message. This key represents a shared secret between the two parties, conventionally we call them Alice and Bob, that want to use a private information link. The main drawback of this kind of cryptography is the requirement of both parties

to have the same secret key. For this, the key must first be transmitted via some trusted means. To do that we have key distribution protocols.

Such a key distribution protocol thus needs to do the following things. Suppose that Alice has a key, a random string of bits, $\mathcal{K}_A$. The protocol must provide a way to transmit to Bob such that Bob obtains a string of bits $\mathcal{K}_B$ and that $\mathcal{K}_A = \mathcal{K}_B$. Also, the protocol needs to make sure that $\mathcal{K}_B$ and $\mathcal{K}_A$ are completely secret, i.e. an eavesdropper, conventionally Eve, cannot have any information about the key. If these requirements are not met, the protcol needs to abort.

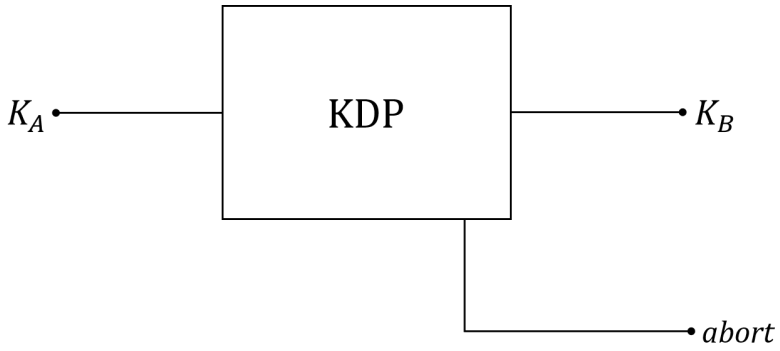

Figure 2.2: Abstract representation of a Key Distribution Protocol. If Alice has her key $\mathcal{K}_A$ then the protocol needs to produce a secret and identical key $\mathcal{K}_B$ on Bob's side. Or if a secure and identical (enough) key is not possible, it must abort.

To prove the security of a key distribution protocol, we want to look for two requirements. Firstly, the key bit strings that Alice and Bob have after completing the used protocol need to be identical. In other words, the keys need to be correct. Secondly, the key bit string should be secret. So, no one other than Alice and Bob should have information about the key.
It is impossible for Alice and Bob to create an ideal key that meets these requirements. This is because of practical issues such as the finite data size and non-ideal error corrections (Xu et al., 2020, p. 8).
The security of an actual key can be parameterised by a deviation ε from a perfect key. The actual key is then called ε-secure. A definition of security is a choice of the quantity that is required to be bounded by ε (Scarani et al., 2009, p. 10).
Usually ε is decomposed into some quantification of correctness and secrecy; $\varepsilon = \epsilon_{\text{cor}} + \epsilon_{\text{sec}}$. Then ϵ_{cor} is the deviation from a correct key, it quantifies the amount in which the keys of Alice and Bob differ. ϵ_{sec} is the deviation of a perfectly secret key, it quantifies the amount of information that Eve has about the key. We will make these definitions more precise in Chapter 3 when we will introduce ε-security for quantum key distributions.

2.4. LINEAR OPTIMISATION

Linear optimisation or linear programming is a method to optimise in a mathematical model, where the requirements are presented by linear (in)equality constraints.

In linear programming the aim is to find the optimum value, usually a minimum or a maximum, of a linear expression called the objective function,

$$f = \mathbf{c}^\top \mathbf{x} = c_1 x_1 + \cdots + c_n x_n, \tag{2.31}$$

where $\mathbf{c}$ and $\mathbf{x}$ are n-dimensional vectors. Note that $\mathbf{c}^\top$ is the transpose of a regular column vector, which is a rowvector, and is presented in this way to form the matrix product with $\mathbf{x}$.

The objective function is subject to linear equality and linear inequality constraints. The set of possible points that satisfy all the constraints is called the feasible region. It forms a convex polytope, which is the set defined as the intersection of finitely many half spaces, each of which is defined by a linear inequality.

Definition 2.12. (Aardal et al., 2020) A set of points that satisfies a linear inequality $\mathbf{ax} \leq b$, $HS = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{ax} \leq b\}$, is called a half space.

A linear program then finds a point inside this polytope for which the objective functions has an extreme value, either small or large, depending on the problem. If the associated polytope is the empty set, such a point cannot be found and the problem is infeasible. If the problem is feasible, i.e. the polytope contains points, then there are two options. Either the optimal objective function value is unbounded. Then there exist feasible solutions with arbitrarily small/large objective function values. Or the optimal objective function value is bounded and we find one or more optimal solutions. In the cases of an infeasible or a feasible but unbounded problem, there exists no optimal solution.

Linear programs are expressed in the following standard form

$$\begin{aligned} &\text{minimise} && \mathbf{c}^\top \mathbf{x} && (2.32)\\ &\text{subject to} && A\mathbf{x} \leq \mathbf{b} && (2.33)\\ & && \mathbf{x} \geq \mathbf{0} && (2.34) \end{aligned}$$

The variables to be determined are the components of the vector $\mathbf{x}$. $\mathbf{c}$ and $\mathbf{b}$ are given vectors and A is a given matrix. Note that we used the linear algebra notation of vectors and not the 'bra-ket' notation that we introduced before. This is because the vectors in linear programs are used as a way of writing linear relationships and the vectors in 'bra-ket' notation are used to describe quantum states.
The inequalities $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$ are the constraints which specify a convex polytope. To show that Equation 2.33 describes m linear restrictions on $\mathbf{x}$, we will repeat this equation in a more elaborate form:

$$\begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ \vdots & & \ddots & \\ A_{m,1} & \cdots & & A_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \leq \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \tag{2.35}$$

In short we can write the whole problem in standard form

$$\min \left\{ \mathbf{c}^\top \mathbf{x} \middle| \mathbf{x} \in \mathbb{R}^n \wedge A\mathbf{x} \leq \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0} \right\} \tag{2.36}$$

Other forms, such as maximisation problems or problems with 'greater-equal' or 'equality' constraints or even problems with negative variables can be rewritten to an equivalent problem in this standard form.

Linear problems can be solved analytically or numerically. However most of the time the problems are complicated, whereby numerical is the way to go.

3

QUANTUM KEY DISTRIBUTION AND THE SECURITY OF THE BB84 PROTOCOL

In the first section of this chapter we will introduce quantum key distributions and derive a security definition for general QKD protocols. Then in Section 3.2, we will present the BB84 protocol, which was the first protocol for quantum key distribution. Apart from a description of the different steps of the protocol, also a formal, iterative protocol is given. In Section 3.3 we describe a deviation of the BB84 protocol called entanglement based BB84. Again there will be a description that highlights the differences from the regular BB84 protocol and a formal, iterative protocol. In Section 3.4 first we introduce CSS codes, a type of quantum-error correcting codes using the concepts from linear codes. In the second part of this section we will give the full proof of the security of entanglement based BB84. We do this by deriving the secure modified Lo-Chau protocol (also an entanglement based BB84 protocol) and providing proofs and justification of all the steps we take to get there. In the last section of this chapter we derive the security of the BB84 protocol from the secure entanglement based protocol. For this derivation we follow the methods presented by Shor and Preskill, 2000, and Nielsen and Chuang, 2002.

3.1. QUANTUM KEY DISTRIBUTION

One particularly secure way of distributing the secret key between the two parties in symmetric secret key cryptography is quantum key distribution. This is because the security of QKD relies on fundamental principles of quantum physics. For example, when Eve wants to extract any form of information out of quantum states, she will have to perform a generalised form of measurement. Measurements in general modify the state of the measured system. As it turns out, such a modification will not go unnoticed by Alice and Bob, thereby notifying them of Eve's presence.
Alternatively, Eve might want to have a perfect copy of the quantum state that Alice sends to

Bob. However, the no-cloning theorem (Wootters and Zurek, 1982) forbids this. This theorem states that one cannot duplicate an unknown quantum state while keeping the original intact.

We will now derive conditions for an ε-secure QKD protocol where $\varepsilon = \epsilon_{\text{cor}} + \epsilon_{\text{sec}}$, as has been done by Xu et al., 2020, p. 8. Suppose that at the end of the protocol, Alice and Bob both have a key bit string of length m; call them $\mathcal{K}_A$ and $\mathcal{K}_B$, respectively. Eve will also have obtained some information which we describe with the quantum state ρ_E. The joint state between Alice, Bob and Eve is

$$\rho_{ABE} = \sum_{k_A,k_B} p(k_A,k_B)|k_A\rangle\langle k_A| \otimes |k_B\rangle\langle k_B| \otimes \rho_E^{(k_A,k_B)}, \tag{3.1}$$

where $k_A, k_B \in \{0,1\}^m$ are the bit values of the key string. In the ideal case, Alice and Bob share a state that results in the same bit string and Eve's state is independent of the shared state between Alice and Bob. We describe this ideal state as

$$\rho_{ABE}^{\text{ideal}} = 2^{-m} \sum_k \underbrace{|k\rangle\langle k|}_{A} \otimes \underbrace{|k\rangle\langle k|}_{B} \otimes \rho_E. \tag{3.2}$$

Now $k_A = k_B = k$; Alice and Bob have the same bit string, thus the key is correct. Also, ρ_E is independent of k, thus the key is secret, Eve does not know anything about the key string. For a QKD protocol to be ϵ_{sec}-secret, the state ρ_{AE} of Alice and Eve needs to be ϵ_{sec} close in trace distance to the private state of Alice, ρ_{AE}^{ideal}:

$$\min_{\rho_E} (1 - p_{\text{abort}}) D(\rho_{AE}, \rho_{AE}^{\text{ideal}}) \leq \epsilon_{\text{sec}}, \tag{3.3}$$

where p_{abort} is the probability that the protocol aborts without giving any results, $\rho_{AE}^{\text{ideal}} = 2^{-m} \sum_k |k\rangle\langle k| \otimes \rho_E$, and $D(\rho,\sigma)$ is the trace distance as defined in Equation 2.21. For a QKD protocol to be ϵ_{cor}-correct, the probability distribution $p(k_A,k_B)$ of ρ_{ABE} in Equation 3.1 should satisfy

$$p(k_A \neq k_B) \leq \epsilon_{\text{cor}}. \tag{3.4}$$

In general, a QKD protocol is ε-secure if the final state ρ_{ABE} is ε-close to the ideal key state $\rho_{ABE}^{\text{ideal}}$ given in Equation 3.2, with a proper chosen ρ_E

$$\min_{\rho_E} (1 - p_{\text{abort}}) D(\rho_{ABE}, \rho_{ABE}^{\text{ideal}}) \leq \varepsilon. \tag{3.5}$$

We can interpret this security definition as the probability that the QKD protocol did not abort and Eve knows the key, is smaller than ε.

LIGHT PARTICLES AS QUBITS

In theory, quantum information can be transmitted and processed with any quantum system, such as ions, atoms, light, spins. But for QKD, the only practical choice is light. The first advantage of using light is that researchers can benefit from all the tools developed for optical telecommunications; think of optical fibers and detectors (Gisin et al., 2002).

Also beneficial is that light has a high speed and can travel over long distances. The problem with light, however, is scattering or losses. These cause that photons do not reach the other side on many occasions. Quantum Key Distribution is affected by losses in various ways.
Firstly, losses place bounds on the secret key rate, which is the fraction of secure key bits that can be extracted from the whole of transmitted bits. The transmittivity of the quantum channel is a deciding factor for the final number of key bits. Also, losses impose bounds on the achievable distance. Secondly, the eavesdropper may use the lost qubits to gain information. For single photon states this is not the case, but in practical implementations, often coherent pulses are used as data carrier. These coherent pulses are light pulses that contain a single photon with high probability but may contain more photon numbers. So for these implementations, losses will certainly leak information, which is problematic (Scarani et al., 2009, p. 4).
Losses will have to be taken into account when creating bounds for the secret key rate.

3.2. THE BB84 PROTOCOL

The first protocol for quantum cryptography was proposed in 1984 by Bennett and Brassard, hence this protocol is nowadays known under the name BB84.
The BB84 protocol allows two users, Alice and Bob, who are in isolated labs and wish to exchange a secret key. They both have access to two channels: a quantum channel which is uncharacterised, it might even be fully controlled by the eavesdropper, and a classical channel, which we assume is authenticated. This means that, when Alice and Bob communicate over this channel, they can be absolutely sure that they are talking to each other and that there is no impostor who can pretend to be the other person or change their messages. However a third party is allowed to listen to the messages, without changing them.

The protocol uses four quantum states in two bases. Usually these are $|0\rangle$, $|1\rangle$ in the Z-basis, the eigenvectors of the Pauli Z-matrix, which are given in Equation 2.1, and $|+\rangle$, $|-\rangle$ in the X-basis, which are given in Equation 2.3 and here we show them again, to show that they are a superposition of $|0\rangle$ and $|1\rangle$:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{3.6}$$

These are the eigenvectors of the Pauli X-matrix. For photons that have their properties so sharply defined that the only degree of freedom left is polarisation, Alice and Bob could agree to align their polarisers in such a way that the mentioned states correspond with the horizontal, vertical, +45 and -45 polarisation, respectively.
To convey messages using these states, the binary value 0 is connected to the states $|0\rangle$ and $|+\rangle$ and binary value 1 to $|1\rangle$ and $|-\rangle$.
In the general, so-called prepare-and-measure BB84 protocol Alice iteratively chooses uniformly a binary bit for the key (Step 1). She stores this bit in her string of initially sent bits $(s_r)_{r=1}^{N} \in \{0,1\}^N$, where N is the random variable that indicates the total number of qubits that Alice will send.
Next, in Step 2, she chooses with probabilities p_x and p_z to encode this bit in either the X-basis or Z-basis. The probabilities p_x and p_z are variables of the protocol. Step 3, she sends the photon to Bob using the quantum channel.

In Step 4, Bob chooses the X-basis or Z-basis with probability p_x and p_z, respectively, to measure the received photon. He stores the bit he found in the string $(t_r)_{r=1}^{N'} \in \{0,1\}^{N'}$, where N' is the total number of qubits that Bob will receive.
Note that whenever they used the same basis to encode and decode (measure), they get perfectly correlated results (given that no error occurred during transmission). When they used a different basis, the results are uncorrelated and the bit that Bob reads will be random (0 or 1).

In Step 5 we will move to classical communication, where Alice and Bob use the authenticated channel to report the bases they used. If Bob did not receive anything, Alice discards her bit. Now, if they used the same basis, they keep the bit and add it to the first sifted bit string, $(\hat{a}_j)_{j=1}^{J}$ for Alice and $(\hat{b}_j)_{j=1}^{J}$ for Bob, where J is the total number of times that Alice and Bob used the same basis. Per added bit, Alice and Bob also document which base, X or Z, was used. With this information, they then check the termination condition: if for at least m qubits that they have shared so far, they both used the Z-basis and for at least k qubits they used the X-basis, they move on to Step 6. Here, both m and k are parameters of the protocol. The loop phase of the protocol, the Steps 1 through 5 is repeated if the termination condition is not met (Pfister et al., 2016).
After the loop phase, in which all the information is produced, Alice and Bob will move to the phase of information processing. In Step 6 and 7 they discard a random subset of the obtained bits, so as to both have a bit string of precisely $n = m + k$ bits. Respectively, these bit strings are $(a_i)_{i=1}^{n}$ and $(b_i)_{i=1}^{n}$.
In the final part of the protocol, Alice and Bob will perform parameter estimation to decide if the protocol needs to abort. They devote the k bits that they measured in the X-basis to be their test bits and reveal these over the public channel. They determine the error rate of these test bits λ_X. If this number exceeds a predetermined maximum tolerated error rate q_{tol}, the protocol aborts.
Otherwise, they output the m bit that were measured in the Z-basis as their raw key. For Alice this is the m-bit string $(\alpha_i)_{i=1}^{m} \in \{0,1\}^m$, for Bob $(\beta_i)_{i=1}^{m} \in \{0,1\}^m$.
We shall present the protocol below as an iterative protocol, like the one by Pfister et al., 2016.

BB84 protocol

Parameters: $m, k \in \mathbb{N}_+$; $p_x, p_z \in [0,1]$ with $p_x + p_z = 1$, $q_{\text{tol}} \in [0,1]$.

Output: For $n = m + k$ the outputs after the sifting are

Alice: n-bit string $(a_i)_{i=1}^{n} \in \{0,1\}^n$ (sifted outcomes),

Bob: n-bit string $(b_i)_{i=1}^{n} \in \{0,1\}^n$ (sifted outcomes),

public: the set K of size k, which contains the indices of the bits that were measured in the X-basis and the set M of size m which contains the indices of the bits that were measured in the Z-basis.

Final output: Either no output if the protocol aborts in Step 9 or:

Alice: m-bit string $(\alpha_i)_{i=1}^m \in \{0,1\}^m$ (raw key),

Bob: m-bit string $(\beta_i)_{i=1}^m \in \{0,1\}^m$ (raw key).

Number of rounds: Random variable N, determined by the termination condition (TC).

The protocol

Loop phase: The steps 1 to 5 are iterated round wise until the termination condition is met. Set $j = 0$. Define the sets M' and K' which are initially empty. Starting with round $r = 1$ (where $r = 1, 2, ...$), Alice and Bob do:

Step 1: (Random bit generation): Alice chooses a bit uniformly at random and records it as s_r.

Step 2: (Preparation): Alice encodes bit s_r in the X-basis or Z-basis with probability p_x, p_z respectively. Binary value 0 is encoded in the X-basis as $|+\rangle$ and in the Z-basis as $|0\rangle$, binary value 1 is encoded in the X-basis as $|-\rangle$ and in the Z-basis as $|1\rangle$.

Step 3: (Channel use): Alice uses the quantum channel to send the prepared qubit to Bob.

Step 4: (Measurement): If Bob received a qubit, he chooses the X-basis or the Z-basis with probability p_x and p_z to measure the received qubit. He stores the binary value he obtains as t_r.

Step 5: (Public communication): Alice and Bob use the authentic classical channel to communicate their basis choice. If the bases were the same:

- $j = j + 1$,
- Alice sets $\hat{a}_j = s_r$,
- Bob sets $\hat{b}_j = t_r$,
- If both used the Z-basis, $M' = M' \cup \{j\}$,
- If both used the X-basis, $K' = K' \cup \{j\}$.

If the bases were different, Alice and Bob do nothing.

TC If the condition ($|M'| \geq m$ and $|K'| \geq k$) is reached (where $|M'|$ is the size of set M'), Alice and Bob set $N = r$ and $J = j$ and proceed with Step 6. Otherwise they set $r = r + 1$ and repeat from Step 1.

Final sifting phase: The following steps are performed only once. Alice now has a bit string $(\hat{a}_j)_{j=1}^J \in \{0,1\}^J$ and Bob has a bit string $(\hat{b}_j)_{j=1}^J \in \{0,1\}^J$ which resulted from the loop phase of the protocol.

Step 6: Alice and Bob will now choose at random a subset of size k from K', this is the set K. They also choose a random subset of size m from M', this is the set M.

Step 7: (Output): From her bit string $(\hat{a}_j)_{j=1}^J \in \{0,1\}^J$, Alice discards all the bits for which the index $j \notin M \cup K$. She then re-indexes her remaining bit string from 1 to n, while preserving the order. She obtains the sifted n-bit string $(a_i)_{i=1}^n \in \{0,1\}^n$. The new indexing is also translated to M and K so that the index numbers of the bits measured in the X-basis are in the set K and the index number of the bits measured in the Z-basis are in the set M and $M \cup K$ contains the numbers 1 to n. Bob carries out the same procedure for $(\hat{b}_j)_{j=1}^J \in \{0,1\}^J$ to obtain the sifted n-bit string $(b_i)_{i=1}^n \in \{0,1\}^n$.

Parameter estimation: In this part, Alice and Bob will test the errors that occurred and decide if the protocol needs to abort.

Step 8: Alice and Bob communicate their test bits over the public, authenticated channel. These are the bits a_i and b_i for which $i \in K$.

Step 9: (Correlation test): Alice and Bob determine the test bit error rate:

$$\lambda_X = \frac{1}{k} \sum_{i \in K} a_i \oplus b_i \tag{3.7}$$

where $\oplus$ is addition modulo 2. If $\lambda_X \leq q_{\text{tol}}$ they continue the protocol in Step 10. If $\lambda_x > q_{\text{tol}}$ they abort.

Step 10: (Raw key output) Alice outputs the m-bit string $(\alpha_i)_{i=1}^m \in \{0,1\}^m$ which consists of all a_i for which $i \in M$. Bob outputs the m-bit string $(\beta_i)_{i=1}^m \in \{0,1\}^m$ which consists of all b_i for which $i \in M$.

POST-PROCESSING

After the protocol Alice and Bob share the so-called raw key. They will move to post-processing. Post-processing is a procedure for Alice and Bob to distill a secure key from the raw key with the help of public communication. The procedure normally consists of the following two steps (from Scarani et al., 2009, p. 22).

The first step is error correction. After this step, the strings of bits that Alice and Bob have, have become shorter, but they are perfectly correlated. It was proved by Shannon that it is possible to extract a set of perfectly correlated bits from list of partially correlated bits. This perfect fraction is bounded by the mutual information $H(\alpha : \beta)$. The mutual information was described as the amount of information α and β have in common. This statement can be made intuitive if we remember the equality $H(\alpha : \beta) = H(\alpha) - H(\alpha|\beta)$. Alice ($\alpha$) has reveal a large enough amount of information to cover the uncertainty that Bob (β) has on her raw key given what he knows about his own (which is the description of conditional entropy $H(\alpha|\beta)$).

The second step is privacy amplification. This procedure aims to delete Eve's knowledge on the raw key. By bounding the amount of errors in the raw key in Step 9, Alice and Bob have

already created some bounds on how much Eve can know. Like we said before, whenever Eve tries to gain information about the key, she will introduce some errors. So by only allowing a small number of errors, we also make Eve's knowledge about the key small. By privacy amplification, we are looking to make Eve's information on the key even smaller. How small we are actually able to make Eve's knowledge is something we will show, when we prove the security of the BB84 protocol, later in this chapter.
If we let I_{EA} and I_{EB} be the information Eve has on the raw keys of Alice and Bob, respectively, then we will need to subtract the quantity $\min(I_{EA}, I_{EB})$ from the fraction of perfectly correlated bits, $H(\alpha : \beta)$, that we obtained from error correction. What remains is the shared secret key.

In the protocol, Alice and Bob abort when they find an error rate that exceeds a certain tolerated number q_{tol}. To find this tolerable error rate we will do a derivation on the errors Alice and Bob might have.
When Alice and Bob both measure a qubit and the outcomes are different, an error has occurred. If they both measured in the local Z-basis, we call it a bit error. If they both measure in the X-basis and the outcomes are different, a phase error occurs. We define the bit and phase error rates to be number of bit (phase) errors per total number of bits in the raw key, which is n. Bit and phase errors are denoted as e_b and e_p, respectively. Errors are caused by losses and imperfections of the system, but also by eavesdropping attacks from Eve. Since we set no restrictions on the quantum channel, it was uncharacterised, Eve can in principle perform any attacks, or measurements, on the qubits in the channel, that physics allows. In the next section we will give an example of one attack. Bit and phase errors can be defined in any two complementary bases, not only in the X- and Z-basis. For most attacks, these two orthogonal errors are correlated. Their correlation works as follows: for the signals that are measured in the basis corresponding with a bit error, the phase error is the hypothetical error that would have been found, were these signals measured in the other basis, and vice versa (Xu et al., 2020, p. 11).

The discussed error correction and privacy amplification, can be seen as separately correcting bit errors, then phase errors. Now we will only give the result of secret fraction of the key that remains after these corrections. Later in this chapter, we will demonstrate this principle more elaborate in the security proof.
When correcting their errors, it turns out that Alice and Bob will have to discard information, i.e. bits from their keys. In the infinite data size limit, the number of bits in Alice's and Bob's keys this costs is given by the binary entropy. $nH(e_b)$, the total number of raw key bits, n, times the entropy of the bit errors, are the bits that need to be discarded due to bit error correction, $nH(e_p)$ are the bits that will be discarded due to phase error correction. So the secret fraction of the key after error correction is given by

$$s \geq 1 - H(e_b) - H(e_p) \tag{3.8}$$

as will turn out.
Now if we consider equal bit and phase error rates, we find the highest tolerable error rate $e = q_{\text{tol}}$ where the function

$$\sigma = 1 - 2H(e) \tag{3.9}$$

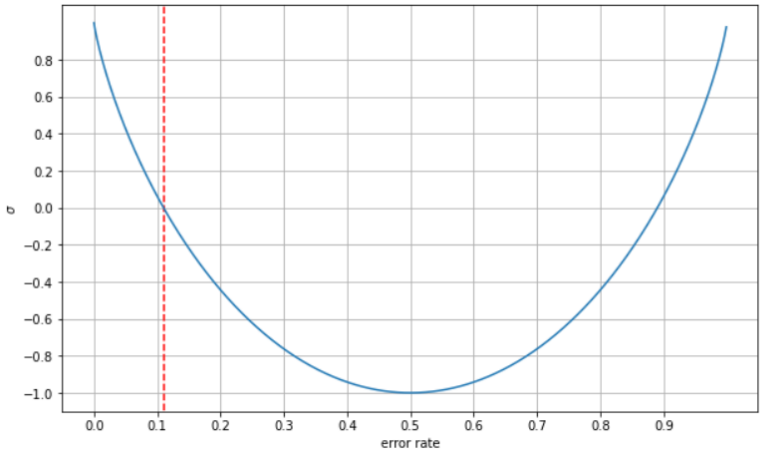

Figure 3.1: Plot of the function $\sigma = 1 - 2H(e)$. The red dotted line indicates an error rate of 11%, for which the function value is zero.

hits zero. This is for an error rate of 11% (Xu et al., 2020, p. 14). In Figure 3.1, σ is plotted against the error rate. The error rate for which the function value is zero is indicated.
For the BB84 protocol the quantum bit error rate (QBER) of 11% is recognised as the highest allowed error rate for which it is still possible to extract secure key.

3.2.1. INTUITION BEHIND THE SECURITY OF BB84

The main intuition behind the security of this quantum key distribution protocol is the fact that Alice and Bob can know, in the case of ideal transmission devices, if the key they established is secret or that some eavesdropper, Eve, might have information about the key. We will show this idea in an example of an attack that might be performed by Eve.

Assume that Eve intercepts a state and proceeds to do the same as Bob; she chooses with some random probability a basis (Z or X) and measures the state. She will then resend the same state she measured to Bob. This is the intercept-resend attack.
If the basis that Eve used to measure the state was the same as the one Alice used for preparation, Eve will have a deterministic measurement result and the photon is the same immediately after the measurement as it was before. In short: Eve has full information and has not caused a disturbance. But if Eve used the other basis to measure, the result she gets is no longer deterministic, but it is equally likely to give a 0 or a 1. This non-deterministic measurement also disturbed the photon, so that its state immediately after the measurement is no longer the same as after sending. Now when Bob would use the same basis to measure as Alice used to send, his measurement result will be uncorrelated with hers. In half of the cases he has a wrong bit (Scarani et al., 2009, p. 5).
Using this attack on every bit of an asymptotically long key, Eve will on average have full information on 50% of the bits. The price she pays is that Alice and Bob will detect a QBER of 25%, revealing Eve's presence.

What if Eve, for example, does not intercept every signal, but just some? Then the QBER will be lower, but Eve still has information on the key. In the rest of this chapter, we will prove that with the BB84 protocol it is possible to extract secure key from the transmitted bits, even if there is noise during transmittance or when Eve has interfered and possibly has information on the key bits.

3.3. ENTANGLEMENT BASED BB84

To prove the security of BB84, we will first introduce an equivalent protocol that is based on entangled states, of which Alice and Bob both measure one half, instead of the prepare-and-measure protocol, which is stated above. The protocols are almost the same, except for the part where quantum information is exchanged between Alice and Bob.
In the entanglement based protocol (from Xu et al., 2020, p. 10), Alice starts off by preparing a qubit pair such that its state is maximally entangled (Step 1). These states are known as the Bell basis and are given as:

$$\Psi^{\pm} = \frac{|01\rangle \pm |10\rangle}{\sqrt{2}}, \qquad \Phi^{\pm} = \frac{|00\rangle \pm |11\rangle}{\sqrt{2}}. \tag{3.10}$$

These states form an orthonormal basis for the quantum state space of two qubits.
She then uses the quantum channel to send half of the pair to Bob, half of the pair, she keeps herself. This is the new Step 2.
In Step 3, both Alice and Bob measure their half of the qubit pair in either the X or Z-basis with the respective probabilities p_x and p_z.
Note that, when Alice first measures half of the entangled state and afterwards sends half to Bob, she essentially sends a pure state to Bob. This is already an easy way to see that the entanglement based protocol can be reduced to the prepare-and-measure protocol.
Then, in Step 4, they store the obtained classical bits in the strings $(s_r)_{r=1}^N \in \{0,1\}$ (Alice) and $(t_r)_{r=1}^N \in \{0,1\}$ (Bob).
From Step 5 and on they do the same as in the prepare-and-measure protocol. The full entanglement based protocol is given in Appendix A.

3.4. SECURITY OF ENTANGLEMENT BASED BB84

In this section we will prove the security of the entanglement based BB84 protocol. We will do this by following the design of the security proof of BB84 that was created by Shor and Preskill, 2000, where we will fill in the missing links by providing proofs and justifications of all the steps.
Before we give this proof, however, we first introduce some concepts. At some point, Alice and Bob will have to do quantum-error correction on their qubits. To show how they will do this, we will introduce the concept of quantum-error correcting codes. These codes use a lot of the ideas from classical linear codes. The following three definitions are paraphrased from Hankerson et al., 1991, Chapter 2.

Definition 3.1. A classical linear $[n,k]$-code C encodes k bits of information into n-bit code words. The code space is specified by G, an n by k generator matrix whose entries are only zeros and ones. The matrix G maps the information strings to their encoded equivalent, the

code word. Thus the k-bit message x is encoded as Gx, where the message x is treated as a column vector.
Let F_2 be the field on two elements, 0 and 1, with additive identity 0 and multiplicative identity 1. C is then a k-dimensional linear subspace of $(F_2)^n$, which is the binary vector space on n bits. C contains 2^k vectors that are code words.

Definition 3.2. The dual code $C^\perp$ of a linear code C is the orthogonal complement of C,

$$C^\perp = \left\{x \in (F_2)^n \middle| c \cdot x = 0 \text{ for all } c \in C\right\}. \tag{3.11}$$

Here $\cdot$ is the dot product.

The correction of a linear code is done by measuring the syndrome of a received word (i.e. an n-string of bits, also a vector). The syndrome can be found as follows. First we need a parity check matrix.

Definition 3.3. The parity check matrix of a $[n,k]$-code C is a $n \times (n-k)$ matrix H over F_2 such that $c \in C \iff cH = 0$.

The syndrome of a word w is then the vector Hw. If the syndrome is 0, then w is a code word, $w \in C$. Otherwise, the most likely error pattern ϵ can be calculated from the syndrome. An error pattern is a vector of length n (the length of the code words) with ones at the places where a bit has flipped and zeros everywhere else. The code word and the error pattern added modulo 2 gives the corrupted word. Correcting may be done by again adding the error pattern modulo 2. Finding the error pattern from the syndrome works as follows.
Every linear code has that the zero word (a vector consisting of only zeros) is a code word. If any error pattern occurs in this code word, we will find a nonzero syndrome. But then the syndrome is easy to connect to the error pattern since the corrupted word is the error pattern. This way we can connect all possible syndromes to the most likely error pattern. Normally these are the ones with the least errors, i.e. the least number of ones. The number of ones in a word or error pattern is called the weight of the word.
So when we find a certain nonzero syndrome for another word, we can simply look up which error pattern caused it and correct the word accordingly.
For classical linear codes, it is specified up to how many errors they can correct. This property is something that is specified by the design of the code. If a classical linear code can correct at most t errors, we mean that for this code we can detect and correct all error patterns that occur, that have at most weight t.

3.4.1. CSS CODES

We will now introduce a large class of quantum-error correcting codes called the Calderbank-Shor-Steane (CSS) codes (from Nielsen and Chuang, 2002, p. 450-451).
We consider the classical linear $[n,k_1]$-code C_1 and classical linear $[n,k_2]$-code C_2, where $C_2 \subset C_1$. Suppose that C_1 and $C_2^\perp$ both correct at most t errors. Let $v \in C_1$ be any code word in the code C_1. Then the quantum state $|v + C_2\rangle$ is defined by

$$|v + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{u \in C_2} |v + u\rangle, \tag{3.12}$$

where + is bit wise addition modulo 2 and $|C_2|$ denotes the number of elements of C_2. If v' is an element of C_1 such that $v - v' \in C_2$, then $|v + C_2\rangle = |v' + C_2\rangle$. All code words defined like this, correspond to cosets of C_2 in C_1.
If v and v' are in different cosets of C_2, then there exist no $u, u' \in C_2$ such that $v + u = v' + u'$, therefore $|v + C_2\rangle$ and $|v' + C_2\rangle$ are orthonormal states.
The quantum code CSS(C_1, C_2), the CSS code of C_1 over C_2, is defined by the vector space spanned by the states $|v + C_2\rangle$ for all $v \in C_1$. There are $|C_1|/|C_2|$ cosets of C_2 in C_1, thus CSS(C_1, C_2) has dimension $|C_1|/|C_2| = 2^{k_1 - k_2}$. We have defined the $[n, k_1 - k_2]$ quantum code CSS(C_1, C_2) that is capable of correcting at most t errors (Nielsen and Chuang, 2002, p. 450).

We will show that the CSS(C_1, C_2) code is capable of correcting up to t bit and phase flip errors by using of the fact that the classical codes C_1 and $C_2^\perp$ both correct t errors. Let e_1 be the n-bit error pattern that describes the bit flips and let e_2 be the n-bit error pattern that describes the phase flips. The following steps will only work when the weight of e_1 and the weight of e_2 are both smaller or equal to t. So this is what we will assume. Write the corrupted state as:

$$\frac{1}{\sqrt{|C_2|}} \sum_{u \in C_2} (-1)^{(v+u)\cdot e_2} |v + u + e_1\rangle. \tag{3.13}$$

Now let H_1 be the parity check matrix of the code C_1. The syndrome of $|v + u + e_1\rangle$ is then $|H_1(v + u + e_1)\rangle = |H_1 e_1\rangle$, since $|v + u\rangle \in C_1$ has a syndrome of zero. When we know the error syndrome $H_1 e_1$, the error pattern e_1 can be retrieved, since its weight is no larger than t. Recovery of the state is then performed by applying an operation of the X matrix to the qubits at the positions of a one in e_1, i.e. the position of a bit flip. This will again flip the qubit and with all the bit flip error removed, the following state remains,

$$\frac{1}{\sqrt{|C_2|}} \sum_{u \in C_2} (-1)^{(v+u)\cdot e_2} |v + u\rangle. \tag{3.14}$$

To detect the phase flip errors, we need to take an extra step. Hadamard gates are applied to each qubit. This takes the state in 3.14 to

$$\frac{1}{\sqrt{|C_2|2^n}} \sum_{w} \sum_{u \in C_2} (-1)^{(v+u)\cdot(e_2+w)} |w\rangle, \tag{3.15}$$

where the sum is over all possible values for some n-bit vector w. Now by setting $w' = w + e_2$, we may rewrite

$$\frac{1}{\sqrt{|C_2|2^n}} \sum_{w'} \sum_{u \in C_2} (-1)^{(v+u)\cdot w'} |w' + e_2\rangle. \tag{3.16}$$

If $w' \in C_2^\perp$ then $\sum_{u \in C_2} (-1)^{u \cdot w'} = |C_2|$, by the definition of the dual code (Definition 3.2). If $w' \notin C_2^\perp$ then $\sum_{u \in C_2} (-1)^{u \cdot w'} = 0$. Thus we rewrite Equation 3.16

$$\frac{1}{\sqrt{2^n/|C_2|}} \sum_{w' \in C_2^\perp} (-1)^{v \cdot w'} |w' + e_2\rangle, \tag{3.17}$$

and this state looks exactly like a bit flip error described by the vector e_2! Now, let H_2 be the parity check matrix for $C_2^\perp$, and in the same way as before, we obtain e_2, since its weight is not

larger than t. We can correct the error e_2 by again applying the X operator to the concerning qubits, obtaining the state

$$\frac{1}{\sqrt{2^n/|C_2|}} \sum_{w' \in C_2^\perp} (-1)^{v \cdot w'} |w'\rangle. \tag{3.18}$$

If we now apply Hadamard gates again to each qubit and rewrite, we return to the state of Equation 3.14 with $e_2 = 0$:

$$\frac{1}{\sqrt{|C_2|}} \sum_{u \in C_2} |v + u\rangle, \tag{3.19}$$

which is the original encoded state.
We see that for CSS codes the phase error correction is decoupled from bit error correction, and that they can be corrected as long as they are at most t.

Let $Q_{x,z}$ be an $[n, m]$ code of C_1 over C_2, which encodes m qubits in n qubits, corrects up to t errors and is parameterised by two vectors x and z. For $v \in C_1$ we define the corresponding code word as

$$|\chi_{v,x,z}\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{u \in C_2} (-1)^{z \cdot u} |x + v + u\rangle. \tag{3.20}$$

Recall that C_1 and C_2 are $[n, k_1]$ and $[n, k_2]$ classical linear codes such that $C_2 \subset C_1$ and C_1 and $C_2^\perp$ both correct t errors, thus $m = k_1 - k_2$. C_2 has then 2^m cosets in C_1 and the basis vectors of $Q_{x,z}$ are indexed by the cosets of C_2 in C_1.
We will show that the quantum codes $Q_{x,z}$ (for arbitrary x and z) are equivalent to CSS(C_1, C_2), by which we mean that both codes have the same error-correcting properties.
Let us first consider the correction of a bit flip error. Previously for the CSS(C_1, C_2) code, bit error correction was done by calculating the syndrome of $|v + u + e_1\rangle$. If we let e_1 again be the error pattern of bit flips. In this case we will want to calculate the syndrome of $|x + v + u + e_1\rangle$. We have $|H_1(x + v + u + e_1)\rangle = |H_1(x + e_1)\rangle$, since $|u + v\rangle \in C_1$ has syndrome of zero. This means we can retrieve the vector $|x + e_1\rangle$. So as long as we know x, we can calculate e_1 and complete the error correction.
Now for the correction of phase flip errors we take again e_2 to be our error pattern. Let

$$\frac{1}{\sqrt{|C_2|}} \sum_{u \in C_2} (-1)^{(x+v+u)\cdot e_2} (-1)^{z \cdot u} |x + v + u\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{u \in C_2} (-1)^{(x+v+u)\cdot e_2 + (z \cdot u)} |x + v + u\rangle \tag{3.21}$$

be the state corrupted by phase flip errors. We apply again Hadamard gates to each qubit, just as we did in Equation 3.15. We obtain

$$\frac{1}{\sqrt{|C_2| 2^n}} \sum_{w} \sum_{u \in C_2} (-1)^{(x+v+u)\cdot(e_2+w) + (z \cdot u)} |w\rangle, \tag{3.22}$$

where we again sum over all possible values of some n-bit vector w. Again set $w' = w + e_2$, to write

$$\frac{1}{\sqrt{|C_2| 2^n}} \sum_{w}{}' \sum_{u \in C_2} (-1)^{(x+v+u)\cdot(w') + (z \cdot u)} |w' + e_2\rangle, \tag{3.23}$$

and by properties of the dual code this becomes

$$\frac{1}{\sqrt{2^n/|C_2|}} \sum_{u \in C_2} (-1)^{(x+v)\cdot(w')+(z\cdot u)} |w' + e_2\rangle, \tag{3.24}$$

and we see that we can correct e_2 the way that is familiar to us, and apply again Hadamard gates to each qubit to return to the original encoded state, just as for a CSS(C_1, C_2) code word. Hence conclude that $Q_{x,z}$ and CSS(C_1, C_2) are equivalent, meaning that they have the same error-correcting properties.

Lastly for CSS codes, we want to give the Gilbert-Varshamov bound for these quantum codes as stated by Nielsen and Chuang, 2002 on p. 449. In the limit as n becomes large, an $[n, k]$ quantum code protecting against errors on up to t qubits exists for some k such that

$$\frac{k}{n} \geq 1 - 2H\left(\frac{2t}{n}\right). \tag{3.25}$$

Thus, good quantum-error correcting codes exist, as long as we do not try to pack too many qubits k into an n qubit code.

3.4.2. SECURITY PROOF

To prove the security of entanglement based BB84, we want to show that the probability that Alice and Bob agree on a key about which Eve can obtain more than an exponentially small amount of information, is exponentially small. When we say that they agree on a key, we mean that the protocol terminates without aborting and Alice and Bob both have a key which they therefore believe is identical and secret. Alice and Bob will then use this key for encryption.
Note that during the use of the quantum channel, Eve can perform any attack on the qubits that the laws of physics allow. But when she obtains information about an entangled pair, shared by Alice and Bob, then this must imply that pair of Alice and Bob is no longer perfectly entangled.
Also the channel or the setup of Alice and Bob may have imperfections. This will also cause errors in the shared qubit pairs.
So it is safe to say that, be it by Eve's doing or by imperfections of the setup, the shared states will contain errors. We will prove that we can still extract a secure key from this.

For the proof we will quantify the security by bounding the information that Eve can have about the key. Note that this is a different criterion than we presented in Section 3.1. From Nielsen and Chuang, 2002, p. 593:

> "A QKD protocol defined as being secure if, for any security parameters $s > 0$ and $l > 0$ chosen by Alice and Bob, and for any eavesdropping strategy, either the scheme aborts, or it succeeds with probability at least $1 - \mathcal{O}(2^{-s})$, and guarantees that Eve's mutual information with the final key is less than 2^{-l}. The key string must also be essentially random."

The first step to getting a secure key is by making sure that Alice and Bob share maximally entangled states. Suppose that Alice has n qubit pairs that are in the maximally entangled state

$|\Phi^+\rangle^{\otimes n}$. When she transmits half of the states to Bob, the resulting state may be impure, so we denote it generally as ρ_{ABE}.

The following lemma and theorem tell us that the fidelity of ρ_{ABE} and $|\Phi^+\rangle^{\otimes n}$ places an upper bound on Eve's mutual information with the key.

Lemma 3.1. (Nielsen and Chuang, 2002, p. 594, *High fidelity implies low entropy*)
If $F(|\Phi^+\rangle^{\otimes n}\langle\Phi^+|^{\otimes n}, \rho_{ABE})^2 > 1-2^{-s}$, then $S(\rho_{ABE}) < (2n+s+1/\ln 2)2^{-s} + \mathcal{O}(2^{-2s})$.

The following proof of this lemma is sketched by Nielsen and Chuang, 2002, on page 594. We shall provide it with all the missing links included.

Proof
Using Equation 2.23, $F(|\Phi^+\rangle^{\otimes n}\langle\Phi^+|^{\otimes n}, \rho_{ABE})^2 = \langle\Phi^+|^{\otimes n}\rho_{ABE}|\Phi^+\rangle^{\otimes n}$. Now since ρ_{ABE} is a density matrix, and suppose it has eigenvalues λ_i, then it is diagonalisable, and can be written as the following: $\rho_{ABE} = \sum_i \lambda_i |i\rangle\langle i|$, where $|i\rangle$ forms an orthonormal basis. We may now write our fidelity as

$$F(|\Phi^+\rangle^{\otimes n}\langle\Phi^+|^{\otimes n}, \rho_{ABE})^2 = \langle\Phi^+|^{\otimes n}\rho_{ABE}|\Phi^+\rangle^{\otimes n}$$
$$= \sum_i \lambda_i \langle\Phi^+|^{\otimes n} i\rangle\langle i|\Phi^+\rangle^{\otimes n} = \sum_i \lambda_i |\langle i|\Phi^+\rangle^{\otimes n}|^2 = \sum_i \lambda_i |c_i|^2$$

where c_i are some complex numbers. Since $|\Phi^+\rangle^{\otimes n}$ is pure and normalised and $|i\rangle$ are orthonormal, we find that $\sum_i |c_i|^2 = 1$. Thus $\sum_i \lambda_i |c_i|^2 \leq \lambda_{\max}$, where $\lambda_{\max}$ is the largest eigenvalue of ρ_{ABE}. From the statement in the lemma we have then that $\lambda_{\max}$ must be larger than $1-2^{-s}$. Now let $\rho_{\max}$ be a diagonal density matrix that is more entropic than ρ_{ABE}. We design it to have diagonal entries $1-2^{-s}, 2^{-s}/(2^{2n}-1), 2^{-s}/(2^{2n}-1), ..., 2^{-s}/(2^{2n}-1)$, that is, $\rho_{\max}$ has one large entry $1-2^{-s}$ and the remaining probability is distributed uniformly among the remaining $(2^{2n}-1)$ entries. Then, the Von Neumann entropy of ρ_{ABE} is bounded above by the Von Neumann entropy of this diagonal density matrix $\rho_{\max}$, $S(\rho_{ABE}) < S(\rho_{\max})$. We calculate $S(\rho_{\max})$:

$$\begin{aligned}
S(\rho_{\max}) &= -(1-2^{-s})\log\left(1-2^{-s}\right) - 2^{-s}\log\frac{2^{-s}}{2^{2n}-1} \\
&= -(1-2^{-s})\log\left(1-2^{-s}\right) - 2^{-s}\left(\log\left(2^{-s}\right) - \log\left(2^{2n}-1\right)\right) \\
&= -(1-2^{-s})\log\left(1-2^{-s}\right) + 2^{-s}\left(s + \log\left(2^{2n}-1\right)\right) \\
&\approx -(1-2^{-s})\frac{\ln(1-2^{-s})}{\ln 2} + 2^{-s}\left(s + \log\left(2^{2n}\right)\right) \\
&= -(1-2^{-s})\frac{\ln(1-2^{-s})}{\ln 2} + 2^{-s}(s+2n)
\end{aligned}$$

where we assumed n to be large to make the simplification in the second term at the fourth equality. To simplify the first term, we will use the Maclaurin series to express $\ln(1-x)$:

$$\ln(1-x) = -\sum_{k=1}^{\infty}\frac{x^k}{k} = -x - \frac{x^2}{2} - \frac{x^3}{3} - \ldots \tag{3.26}$$

and we obtain

$$\ln\left(1-2^{-s}\right) = -2^{-s} - 2^{-2s}/2 - \ldots$$

The first term then becomes

$$\begin{aligned}&\frac{-(1-2^{-s})(-2^{-s}-\mathcal{O}(2^{-2s}))}{\ln 2}\\ &= \frac{2^{-s}+\mathcal{O}(2^{-2s})}{\ln 2}\\ &= 2^{-s}(1/\ln 2)+\mathcal{O}(2^{-2s}).\end{aligned}$$

Putting the first and second term together again, we get

$$S(\rho_{\max}) = (2n+s+1/\ln 2)2^{-s}+\mathcal{O}(2^{-2s}), \tag{3.27}$$

which is the desired result. □

The entropy $S(\rho_{ABE})$ forms an upper bound on the information that Eve can access, by Holevo's bound.

Theorem 3.2. (Nielsen and Chuang, 2002, p. 531, *The Holevo bound*) Suppose Alice prepares a state ρ_X where $X = 0,\ldots,n$ with probabilities $p_0,\ldots,p_n$. Bob performs a projective measurement on that state, with measurement outcome Y. The Holevo bound states that for any such measurement Bob may do:

$$H(X:Y) \leq S(\rho) - \sum_x p_x S(\rho_x), \tag{3.28}$$

where $\rho = \sum_x p_x \rho_x$.

The proof of this theorem can be found in Nielsen and Chuang, 2002, on the pages 532 to 534.

Thus, if we can find a QKD protocol that provides Alice and Bob with entangled pairs that have a fidelity of at least $1-2^{-s}$ with the perfect state, then the protocol is s-secure. In fact what we want is that the QKD protocol is able to put a lower bound on the fidelity of the pairs shared between Alice and Bob. We will show how this is done.
If we send a qubit with state $|\psi\rangle$ through a noisy channel it can pick up arbitrary errors. Such an arbitrary error may be described as a linear combination of four operations: nothing happens (I), a bit flip (X), a phase flip (Z), or a combination of a bit and phase flip (XZ). However if we would measure the error syndrome, this combination of different corruptions will collapse to one of the four states $|\psi\rangle$, $X|\psi\rangle$, $Z|\psi\rangle$ or $XZ|\psi\rangle$.
Now, let us recall the Bell basis in Equation 3.10 and suppose that the first qubit in the pair is the one that Alice keeps and the second is the one that Bob receives. If a bit flip occurs, $|\Phi^+\rangle$ is transformed to $|\Psi^+\rangle$. $|\Phi^-\rangle$ is created by a phase flip and we find $|\Psi^-\rangle$ for a combination of the two errors.
To detect bit flips by a measurement, we construct the projectors $\Pi_{\text{bf}} = |\Psi^+\rangle\langle\Psi^+| + |\Psi^-\rangle\langle\Psi^-|$ and $I - \Pi_{\text{bf}}$. Likewise, the measurement described by projectors $\Pi_{\text{pf}} = |\Phi^-\rangle\langle\Phi^-| + |\Psi^-\rangle\langle\Psi^-|$ and $I - \Pi_{\text{pf}}$ detect phase flips.
The measurement operators we just made both commute with all of the Bell states, meaning that a measurement does not corrupt the whole state. This implies that their outcomes obey classical probability arguments. Therefore, we will claim that Alice and Bob are able to bound their entangled pairs' fidelity with a perfect state ($|\Phi^+\rangle$) by measuring a random subset

of them and checking for bit and phase flips.

For example, suppose that Alice prepares $2n$ entangled pairs and sends the second half of each pair to Bob. Next, they select randomly n of the pairs and for each pair they choose either Π_{bf} or Π_{pf}, at random, and use that operator to jointly measure their qubits. We call this a random sampling test. In the case that δn bit or phase flips are detected, we may conclude that the remaining n entangled pairs, the ones that were not tested, should be almost certain to have also this number of errors, if they were to be measured in this Bell basis.
To justify this statement, we summarise in the following claim the bounds that we want a random sampling test to be able to put on the errors of the untested bits. A description of the claim and the idea of proof can be found in Nielsen and Chuang, 2002, on page 589. We will proceed to give a full proof this claim.

Claim For any $\delta > 0$ (and $\epsilon > 0$), the probability of obtaining less than δn errors on the check bits and more than $(\delta+\epsilon)n$ error on the remaining bit is asymptotically less than $\exp\left[-\mathcal{O}(\epsilon^2 n)\right]$, for large n (Note that this is the exponent of base 2).

Proof
To show this, without loss of generality, we assume that there are μn error in the $2n$ bits, where $0 \leq \mu \leq 2$. Now, if there are δn errors on the check bits, and $(\delta+\epsilon)n$ errors on the rest, then $\delta = (\mu - \epsilon)/2$. Now from this definition of δ we derive the following implication:

$$> (\delta+\epsilon)n \text{ errors on rest} \quad \Longrightarrow \quad > (\mu-\delta)n \text{ errors on rest.} \tag{3.29}$$

But from our assumption that there are μn errors in the $2n$ bits, we also find

$$< \delta n \text{ errors on check bits} \quad \Longrightarrow \quad > (\mu-\delta)n \text{ errors on rest} \tag{3.30}$$

Now let p be the probability that we would like to bound in our claim. Then by these implications, we see that it is the same as the probability of having less than δn errors on the check bits. So we have, $p = p(x < \delta n)$, where x is the number of errors.
Now we would like to introduce the hypergeometric distribution, a discrete probability distribution, that looks at the elements of a certain set being sampled, where the result of each draw can be classified into one of two mutually exclusive categories; a success or a fail. The probability mass function is given as

$$p(x = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}, \tag{3.31}$$

with parameters $N, K, n \in \mathbb{N}$, where $K, n < N$ and $k \in \{0, 1, \ldots, n\}$. Here N is the size of the set, K is the number of successes in the set, n is the number of draws, k is the number of observed successes and $\binom{a}{b}$ is a binomial coefficient (Fetsje Bijma, 2016).
For our problem we take $N = 2n$, the total number of bits, $K = \mu n$, the number of errors in the $2n$ bits, n is the number of check bits and $k = \delta n$, the observed errors in the check bits. Then

$$p(x = \delta n) = \frac{\binom{\mu n}{\delta n}\binom{(2-\mu)n}{(1-\delta)n}}{\binom{2n}{n}}. \tag{3.32}$$

Now we want to find $p(x<\delta n)=\sum_{i=0}^{\delta n-1} p(x=i)$, for a discrete probability distribution. Since the number of bits is twice the number of check bits, the probability mass function of the hypergeometrical distribution is symmetric with a maximum at $K/2$, that is in this case $(\mu n)/2$. With our definition of $\delta=(\mu-\epsilon)/2$, δn is to the left of the center of this distribution, where the probability mass function is increasing. So $p(x=i)<p(x=\delta n)$ for all $i<\delta n$. Therefore we may say

$$p(x<\delta n)=\sum_{i=0}^{\delta n-1} p(x=i)<p(x=\delta n)\cdot\delta n. \tag{3.33}$$

We find the following bound for our probability

$$p(x<\delta n)<\binom{2n}{n}^{-1}\binom{\mu n}{\delta n}\binom{(2-\mu)n}{(1-\delta)n}\delta n. \tag{3.34}$$

We will now find an upper bound for the binomial coefficient $\binom{an}{bn}$ for large n by using Stirling's approximation, $n!\approx\sqrt{2\pi n}(\frac{n}{e})^n$ for large n. We have that

$$\begin{aligned}
\binom{an}{bn}&=\frac{(an)!}{(bn)!((a-b)n)!} && (3.35)\\
&\approx\frac{\sqrt{2\pi an}\left(\frac{an}{e}\right)^{an}}{\sqrt{2\pi bn}\left(\frac{bn}{e}\right)^{bn}\sqrt{2\pi(a-b)n}\left(\frac{(a-b)n}{e}\right)^{(a-b)n}} && (3.36)\\
&=\frac{\sqrt{a}\left(\frac{an}{e}\right)^{an}}{\sqrt{2\pi n}\sqrt{b(a-b)}\left(\frac{bn}{e}\right)^{bn}\left(\frac{(a-b)n}{e}\right)^{(a-b)n}} && (3.37)\\
&=\frac{\sqrt{a}(an)^{an}}{\sqrt{2\pi n}\sqrt{b(a-b)}(bn)^{bn}((a-b)n)^{(a-b)n}} && (3.38)
\end{aligned}$$

When we just focus on the terms that are not in square roots:

$$\begin{aligned}
&(an)^{an}(bn)^{-bn}((a-b)n)^{-(a-b)n}\\
=&a^{an}b^{-bn}(a-b)^{-(a-b)n}\\
=&a^{an}b^{-bn}(1-b/a)^{-(a-b)n}a^{-(a-b)n}\\
=&(b/a)^{-bn}(1-b/a)^{-(a-b)n}\\
=&2^{-bn\log b/a-(a-b)n\log(1-b/a)}\\
=&2^{an(-b/a\log b/a-(1-b/a)\log(1-b/a)}\\
=&2^{anH(b/a)}
\end{aligned} \tag{3.39}$$

where $H(\cdot)$ is the binary entropy. Since n is taken to be large, we may conclude that

$$\sqrt{\frac{a}{2\pi nb(a-b)}}\leq 1 \tag{3.40}$$

and we have found an upper bound for the binomial coefficient $\binom{an}{bn}$,

$$\binom{an}{bn} \leq 2^{anH(b/a)}. \tag{3.41}$$

For the lower bound, let us look at the function $f(x) = x^\kappa (1-x)^{\mu-\kappa}$ on the interval $[0,1]$. With integration by parts we find

$$\begin{aligned}
\int_0^1 f(x)\,\mathrm{d}x &= \int_0^1 x^\kappa (1-x)^{\mu-\kappa}\,\mathrm{d}x \\
&= \frac{1^{\kappa+1}(1-1)^{\mu-\kappa}}{\kappa+1} - \frac{0^{\kappa+1}(1-0)^{\mu-\kappa}}{\kappa+1} + \frac{\mu-\kappa}{\kappa+1}\int_0^1 x^{\kappa+1}(1-x)^{\mu-\kappa-1}\,\mathrm{d}x \\
&= \frac{\mu-\kappa}{\kappa+1}\int_0^1 x^{\kappa+1}(1-x)^{\mu-\kappa-1}\,\mathrm{d}x \\
&= \frac{(\mu-\kappa)(\mu-\kappa-1)}{(\kappa+1)(\kappa+2)}\int_0^1 x^{\kappa+2}(1-x)^{\mu-\kappa-2}\,\mathrm{d}x \\
&\;\;\vdots \\
&= \frac{\kappa!(\mu-\kappa)!}{\mu!}\int_0^1 x^\mu\,\mathrm{d}x = \frac{\kappa!(\mu-\kappa)!}{\mu!}\frac{1}{\mu+1} \\
&= \frac{1}{(\mu+1)\binom{\mu}{\kappa}}.
\end{aligned} \tag{3.42}$$

By taking two times the derivative we find that, on $[0,1]$, f achieves its maximum at $x = \kappa/\mu$ so we get an upper bound

$$\int_0^1 f(x)\,\mathrm{d}x \leq 1 \cdot f(\kappa/\mu) \tag{3.43}$$

Now, let us take $\mu = an$ and $\kappa = bn$, then we obtain

$$f(bn/an) = f(b/a) = (b/a)^{bn}(1-b/a)^{(a-b)n} = 2^{-anH(b/a)}, \tag{3.44}$$

where the last equality can be justified by the derivation in 3.39. If we also substitute $\mu = an$ and $\kappa = bn$ in Equation 3.42, we find the inequality

$$\frac{1}{(an+1)\binom{an}{bn}} \leq 2^{-anH(b/a)}, \tag{3.45}$$

which, if we rearrange the terms gives us the lower bound on the binomial coefficient:

$$\binom{an}{bn} \geq \frac{1}{an+1}2^{anH(b/a)}. \tag{3.46}$$

Combining the upper and lower bound, we get:

$$\frac{1}{an+1}2^{anH(b/a)} \leq \binom{an}{bn} \leq 2^{anH(b/a)}. \tag{3.47}$$

If we apply this bound on all the binomial coefficients in Equation 3.34, we obtain the following expression for p

$$p < \frac{2n+1}{2^{2nH(1/2)}} 2^{\mu n H(\delta/\mu)} 2^{(2-\mu)nH((1-\delta)/(2-\mu))} \delta n \tag{3.48}$$

We now apply the bound $0 \leq H(x) < 1-2(x-1/2)^2$, which states that the binary entropy given in Equation 2.25 is smaller than the parabola that attains its maximum value in $(1/2,1)$ and is zero for $x = \sqrt{1/2} \pm 1/2$. If we also assume for μ the worst case, $\mu = 1/2$, and use $\delta = (\mu - \epsilon)/2$, we will get

$$\begin{aligned} p <& \frac{2n+1}{2^{2n}} 2^{n/2 H(1/2-\epsilon)} 2^{(3n)/2 H(1/2+\epsilon/3)} \delta n \\ <& \frac{2n+1}{2^{2n}} 2^{n/2-\epsilon^2 n} 2^{(3n)/2-(\epsilon^2 n)/3} \delta n \\ =& (2n+1)\delta n \exp\left[2n - 2n - (4/3)\epsilon^2 n\right] \\ =& (2n+1)\delta n \exp\left[-(4/3)\epsilon^2 n\right] \end{aligned} \tag{3.49}$$

where we have the second inequality since 2^n for $n \geq 1$ is an increasing function. With "exp" we mean the exponent with base 2 and we obtain the final result $p < \exp\left[-\mathcal{O}(\epsilon^2 n)\right]$. □

To repeat, Alice and Bob are able to bound the fidelity of their entangled pairs by measuring a random subset of them and checking for bit and phase flips. Note that $\Pi_{\text{bf}} = (I \otimes I - Z \otimes Z)/2$ and $\Pi_{\text{pf}} = (I \otimes I - X \otimes X)/2$. So we see that it is actually possible for Alice and Bob to do these checks simply by making local measurements with the Pauli operators X and Z.
Note however that a local measurement such as $X \otimes I$ or $I \otimes X$ does not commute with the Bell basis. Therefore we want to emphasise that Alice and Bob will only use information on qubits that they measured both in the same basis, and that they will share the share the results of the measurements with these projectors. Hence, two local measurements in the exact same basis will give the same statistics as when Alice and Bob had actually measured Π_{bf} and Π_{pf}.
Thus, by measuring a random sample in the Bell basis, Alice and Bob end up with entangled pairs in the state ρ_{ABE}, which fidelity to the ideal state $|\Phi^+\rangle$ is known. This fidelity bounds the mutual information Eve has by performing any measurement on ρ_{ABE}. However, to get a secure key by the criterion we set up before, Alice and Bob need to reduce Eve's mutual information with their state so that it will be exponentially small. To do this, Alice and Bob will perform so called entanglement distillation to obtain ρ'_{ABE}, a state close to $|\Phi^+\rangle^{\otimes m}$ for some $m < n$, and afterwards measure this final state (Nielsen and Chuang, 2002, p. 595).
Entanglement distillation can be accomplished by performing quantum-error correction. By our previous claim, we know that ρ_{ABE} should have close to δn errors. If we then use a quantum code that corrects up to δn errors to encode the qubits from ρ_{ABE}, it should be possible to correct all of the errors (Nielsen and Chuang, 2002, p. 596). Suppose Alice and Bob use the previously introduced code $\text{CSS}(C_1, C_2)$ with $t = \delta n$ to do entanglement distillation. At the moment, they share n pairs of qubits in a state that is close to $|\Phi^+\rangle^{\otimes n}$. For the entanglement distillation, they separately measure the eigenvalues of $Z^{[r]}$ for each row $r \in H_1$ and $X^{[r']}$ for each row $r' \in H_2$. Here, X and Z are the corresponding Pauli matrices given in 2.5. Note that $Z^{[r]}$ and $X^{[r']}$ commute because the vector spaces $C_1^\perp$ and C_2 are orthogonal. Note too that these syndrome measurements commute with the Bell basis, since Alice and Bob perform identical tasks. To demonstrate entanglement distillation, we will look at two cases.

Suppose that Alice and Bob have n perfect maximally entangled pairs, $|\Phi^+\rangle^{\otimes n}$, to begin with. Then measuring $Z^{[r]}$ for $r \in H_1$ and $X^{[r']}$ for $r' \in H_2$ will project those states onto the code subspace of $Q_{x,z}$. For $Q_{x,z}$ let x and z be binary vectors such that $H_1 x$ and $H_2 z$ are precisely equal to the measured bit and phase syndromes, respectively. $|\Phi^+\rangle^{\otimes n}$ projected onto the code subspace will then give the state $|\Phi^+\rangle^{\otimes m}$ which is encoded by $Q_{x,z}$.
Now, suppose that Alice and Bob have a state that is close to $|\Phi^+\rangle^{\otimes n}$ that has t or fewer random errors. If they then compare their measurements of $Z^{[r]}$ and $X^{[r']}$, where they do not agree, the rows r and r' of these measurements indicate the bits which are 1 in the bit and phase syndromes. Alice and Bob can then locate and correct the errors by the method given in the previous section. Then they can decode $Q_{x,z}$ to obtain an error-corrected state having fidelity with $|\Phi^+\rangle^{\otimes m}$ that is in the order of one minus the probability that more than δn errors occurred (Nielsen and Chuang, 2002, p. 596).
Putting all the steps we took so far together, gives us the modified Lo-Chau protocol (from Shor and Preskill, 2000):

Modified Lo-Chau protocol

1: Alice creates $2n$ entangled pairs in the state $|\Phi^+\rangle^{\otimes n}$.

2: Alice selects n of the $2n$ encoded entangled 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 pair for which b is 1.

4: Alice sends the second half of each pair to Bob.

5: Bob receives the qubits and publicly announces this fact.

6: Alice announces the bit string b, and which n pairs are to be 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 pairs in the $|0\rangle, |1\rangle$ basis and share the results. If more than t of these measurements disagree, they abort the protocol.

9: Alice and Bob make the measurements on their code qubits of $Z^{[r]}$ for each row $r \in H_1$ and $X^{[r']}$ for each row $r' \in H_2$. Alice and Bob share the results, compute the syndromes for bit and phase flips, and then transform their state so as to obtain m nearly perfect entangled pairs.

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

In Step 3 and 7, the random Hadamard transforms are meant to confuse Eve, since they create a symmetry for detecting information encoded in the X- and Z-bases. They also allow Alice and Bob to choose randomly if they measure the check qubits with Π_{bf} or Π_{pf}. By the Gilbert-Varshamov bound for CSS codes in Equation 3.25 we deduce that as long as we have a large word length n, it is possible to find a suitable quantum codes. Therefore, for a δn error correcting $[n, m]$ quantum code, if n is large enough, the criteria for security are satisfied.

3.5. SECURITY OF BB84 FROM ENTANGLEMENT BASED BB84

We have now proved that the above stated modified Lo-Chau protocol is secure. Except for some cosmetic details, it is possible to see that the modified Lo-Chau protocol is similar to the entanglement BB84 protocol which we presented in Section 3.3.
In this section, we will adapt some steps of the protocol to obtain a secure BB84 protocol. To do this we will follow the same steps that are taken in Nielsen and Chuang, 2002, Chapter 12.
An important aspect is that the modified Lo-Chau protocol expects that Alice and Bob are able to store quantum states to measure later on. We will have to get rid of this requirement. Also, we need to lose the need of using entangled states, since, in practice, it is easier to prepare single states than to prepare entangled states. Thus, not needing entangled states makes the protocol more useful.

From Shor and Preskill, 2000, we should observe that it is inconsequential if Alice measures her check bits before she sends half of an entangled pair to Bob or afterwards. Measuring the syndrome before or after transmission will also not make a difference. So let us suppose that she will measure the check bits first, then this is the same as if she would choose just a random state out of $|0\rangle$ or $|1\rangle$. First measuring the syndrome is then the same as sending m halves of entangled pairs encoded by the CSS code $Q_{x,z}$ for two random vectors $x, z \in (F_2)^2$, determined by syndrome measurements $Z^{[r]}$ for rows $r \in H_1$ and $X^{[r']}$ for rows $r' \in H_2$.
Moreover, Alice is also free to choose whether she will measure her half of the encoded entangled pairs before transmitting the rest to Bob or afterwards. First measuring her half is then is equal to deciding upon a random key $|k\rangle$ consisting of m bits and encoding $|k\rangle$ using $Q_{x,z}$. We let v be a representative of one of the 2^m cosets of C_2 in C_1 and be indexed by k to generate $\chi_{v,x,z}$. Then Alice sends the encoded n qubits to Bob. This gives the modified steps (as taken from Nielsen and Chuang, 2002)

1': Alice creates n random check bits, a random m-bit key k, and two random n-bit strings x and z. She encodes $|k\rangle$ in the code $Q_{x,z}$. She also encodes n qubits as $|0\rangle$ or $|1\rangle$ according to the check bits.

2': Alice randomly chooses n positions out of $2n$ and puts the check qubits in these positions and encoded qubits in the remaining positions.

6': Alice announces b, x, z and which n qubits are to provide check bits.

9': Bob decodes the remaining n qubits from $Q_{x,z}$.

10': Bob measures his qubits to obtain the shared secret key k.

The protocol altered with these steps is known as the CSS codes protocol and is secure since it is directly reduced from the modified Lo-Chau protocol. We will still make further reductions on this protocol to finally land at secure BB84.
In the new last two steps it may be observed that Bob goes ahead and measures his qubits using the Z-basis, immediately after he decoded them. He is only interested in the bit values of the encoded key, thus the information for performing phase corrections that Alice still sends is not needed (Shor and Preskill, 2000).
Recall that C_1 and C_2 are classical codes. Now Bob decodes the qubits and then measures them, but instead he could immediately measure the encoded qubits to obtain $x + v + u + e$, where e is some possible error, due to channel losses and the influence of Eve. The next step

is then to decode this message classically by subtracting x, which was sent by Alice, from $x+v+u+e$. Classical error correction can be used to find the corresponding code word in C_1. This code word is then definitely $v+u$ if the weight of the error did not exceed t. The final key k is the coset of $v+u+C_2$ in C_1. This gives us then (from Nielsen and Chuang, 2002):

9": Bob measures the remaining qubits to get $x+v+u+e$, and subtracts x from the result, correcting it with code C_1 to obtain $v+u$.

10": Bob computes the coset of $v+u+C_2$ in C_1 to obtain the key k.

It is not necessary for Alice to reveal the vector z, therefore effectively, the state she sends is a mixed one, averaged over random values of z,

$$\rho_{v,x} = \frac{1}{2^n}\sum_z |\chi_{v,x,z}\rangle\langle\chi_{v,x,z}| \tag{3.50}$$

$$= \frac{1}{2^n|C_2|}\sum_z \sum_{u_1,u_2\in C_2} (-1)^{z\cdot(u_1+u_2)} |x+v+u_1\rangle\langle x+v+u_2| \tag{3.51}$$

$$= \frac{1}{|C_2|}\sum_{u\in C_2} |x+v+u\rangle\langle x+v+u|. \tag{3.52}$$

To create this state, Alice needs to classically choose $u\in C_2$ at random and construct $|x+v+u\rangle$, using her randomly determined x and k.
In the scheme we have so far, Alice sends $|x+v+u\rangle$, which Bob receives and measures with the occurred error, resulting in $x+v+u+e$. Alice then also sends x and Bob subtracts this vector to get $v+u+e$. Consider the following: if Alice chooses $v\in C_1$ immediately, instead of in the coset of C_2 in C_1, then u has become unnecessary. On top of that, the n-bit string $v+x$ is completely random. Thus for the result, it would be the same if Alice randomly chooses a string x and sends $|x\rangle$. After receiving, Bob measures $x+e$. Then Alice will send the classical bit string $x-v$ and by subtraction, Bob is able to get $v+e$. Now there is no longer any difference between random check bits and the bits for the code. We obtain the following altered steps (from Nielsen and Chuang, 2002):

1": Alice chooses a random $v\in C_1$, and creates $2n$ qubits in the state $|0\rangle$ or $|1\rangle$ according to $2n$ random bits.

2": Alice randomly chooses n positions (out of $2n$) and designates these as check qubits, and the remainder as $|x\rangle$.

6": Alice announces b, $x-v$, and which n qubits are to provide check bits.

9"': Bob measures the remaining qubits to get $x+e$, and subtracts $x-v$ form the result, correcting it with code C_1 to obtain v.

10": Alice and Bob compute the coset of $v+C_2$ in C_1 to obtain the key k.

Note in Step 3, that instead of performing Hadamard gates, Alice can also directly encode her qubits in either the Z-basis or X-basis, so that it corresponds to the bits of b.

1"': Alice creates $(4+\delta)n$ random bits. For each bit, she creates a qubit in either the X or Z-basis, according to a random bit string b.

After all these alterations, encoding and decoding of the bits are performed classically. The remaining problem to be solved is removing the need for a quantum memory. So suppose that Bob does his measurements immediately after he received the qubits from Alice and chooses to measure either in the X- or Z-basis according to some random probability distribution (we introduces this as p_x and its complement p_z). Then, as soon as Alice announces the string b, so which bases she used, they can keep those bits for which their bases turned out to be the same. Note here that Alice need not make a random vector b to determine for each qubit in which basis it is encoded. Since she announces this basis after each transmission, she might as well just choose to encode a bit in the X or Z-basis according to probabilities p_x and p_z.
Bob no longer needs to store quantum states, but since they will discard half of their bits with high probability, it is advisable to start with a little (δ) over twice the number of random bits as before. Also choosing which bits are going to be check bits needs to be postponed until the discarding of the bits with non-matching bases is complete. Alice might choose to use all the bits that were encoded in either the X- or Z-basis or a random set. This gives us then the final BB84 protocol which is exactly the same as given in Section 3.2.

4

MODELS FOR PRACTICAL QUANTUM KEY DISTRIBUTION

In this Chapter we will derive a model to perform practical quantum key distribution based on the BB84 protocol.
We will do this by generally following the tutorial by Marcos Curty, 2013. This means that we will do all of the derivations and calculations ourselves, but we will use the same setup and parameters as Marcos Curty, so that we will be able to compare our results to see if our model is correct.
In Section 4.1 we will give precise characterisations of all the components of our QKD system. These are the same components as used by Marcos Curty, 2013. From these characterisations, we will then derive all the variables we need to calculate the secret key rate, the amount of secure key bits per photon that was initially sent, i.e. secure key bits per pulse, that we are able to generate with this system.
Then in Section 4.2 we will calculate this secret key rate and plot the results for varying transmission distances.
In Section 4.3 we introduce a new strategy of sending photons from Alice to Bob, the decoy state method. We will explain why this method is useful and show how it is applied. Again the secret key rate is calculated and the results for varying distances are plotted.

4.1. CHARACTERISATION OF SETUP COMPONENTS

Before we start building the model of the system that will generate the secret key rate, we will first find the necessary parameters to characterise our components.

THE SOURCE

For the BB84 protocol to be perfectly secure, we need a source that produces single photon states. The best way to approximate these single photon states is by using coherent states that have an extremely low mean photon number μ. These signals can be realised using semiconductor lasers and calibrated attenuators. We call these type of signals Weak Coherent Pulses

(WCPs). The probability to find n photons in such a coherent state follows the Poisson distribution with mean μ,

$$P_\mu(n) = \frac{\mu^n}{n!} e^{-\mu}. \tag{4.1}$$

(Gisin et al., 2002 p. 12, Xu et al., 2020, p. 20)
The probability that a non-empty weak coherent pulse contains more than one photon can be made arbitrarily low, since it depends on μ, which is a variable we can adjust.

The coherent states are given by

$$|\alpha\rangle = e^{-|\alpha|^2/2} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle, \tag{4.2}$$

where $\mu = |\alpha|^2$ and $|n\rangle$ is the n-photon number state (Xu et al., 2020, p. 20). For the calculations later on, it will be useful to give a definition of the n-photon number state

$$|n\rangle = \frac{1}{\sqrt{n!}} (a^\dagger)^n |0\rangle, \tag{4.3}$$

where $a^\dagger$ is a creation operator and $|0\rangle$ is the state of vacuum.
By randomising the phase of these coherent pulses, Alice can make it a mixture of photon number states

$$\rho = \frac{1}{2\pi} \int_0^{2\pi} |\alpha e^{i\phi}\rangle\langle\alpha e^{i\phi}| \, \mathrm{d}\phi = \sum_{n=0}^{\infty} P_\mu(n) |n\rangle\langle n|. \tag{4.4}$$

Since we prefer only the single photon states for secure key distribution, typically $\mu = \mathcal{O}(1)$ (Xu et al., 2020, p. 21).

THE CHANNEL

Let's look at the channel we use to send the information.
We use the parameter η_{channel} to model the losses in the quantum channel. This quantum channel is usually a single mode fiber and losses depend exponentially on the transmission distance.

$$\eta_{\text{channel}} = 10^{-\frac{\alpha \cdot \text{distance}}{10}} \tag{4.5}$$

where the transmission distance is given in km and α is the loss coefficient in dB/km.
To implement the losses in the channel in our calculations, we need to introduce an optical device called the beam splitter. The beam splitter has two input modes, a and b, and two output modes c and d. If we neglect absorption and other imperfections, we have the following relation between the creation operators of the input and output modes

$$\begin{pmatrix} a^\dagger \\ b^\dagger \end{pmatrix} = e^{i\phi} \begin{pmatrix} te^{i\phi_t} & re^{i\phi_r} \\ -re^{-i\phi_r} & te^{-i\phi_t} \end{pmatrix} \begin{pmatrix} c^\dagger \\ d^\dagger \end{pmatrix} \tag{4.6}$$

with r, t, ϕ_r and ϕ_t the reflection and transmission coefficients of the amplitude and phase.

If we want to use beam splitters to model the losses in the quantum channel, we will have for input b a vacuum. a is then the signal that enters the channel, c is the output and d will represent the qubits that are lost. A schematic representation of a beam splitter with a vacuum as input b is given in Figure 4.1. The relation between the input and output modes is given as:

$$\begin{pmatrix} a^\dagger \\ b^\dagger \end{pmatrix} = \begin{pmatrix} \sqrt{\eta_{\text{channel}}} & \sqrt{1-\eta_{\text{channel}}} \\ -\sqrt{1-\eta_{\text{channel}}} & \sqrt{\eta_{\text{channel}}} \end{pmatrix} \begin{pmatrix} c^\dagger \\ d^\dagger \end{pmatrix}, \tag{4.7}$$

where we see the parameter η_{channel} influences all of the output.

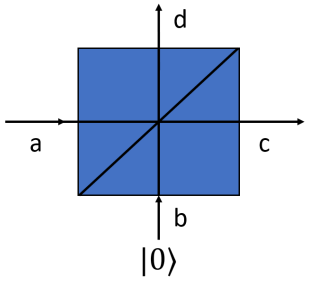

Figure 4.1: Schematic drawing of a beam splitter, with input modes a and b, and output modes c and d. To model the losses in the quantum channel, we use a beam splitter with input b a vacuum and channel parameter η_{channel}.

THE DETECTORS

Because our source will not always produce a perfect single photon state, we need a detector that is at least very good at detecting single photon states. The ideal detector needs to satisfy the following requirements (Gisin et al., 2002, p. 18):

- it should have a high quantum detection efficiency,
- the probability of giving a click without a photon arriving, should be small,
- for a good timing resolution, the time between detection of a photon and generating a click should be constant (small time jitter),
- the recovery time (dead time) should be small.

For the detectors in Bob's setup, we use so called threshold detectors. These can only distinguish the vacuum from single photon or multi-photon cases. So in the ideal case we expect the detectors to give a click in the event that at least one photon is detected and give no click if no photon is detected.
The detectors can be characterised by for example the detection efficiency η_{det}, which indicates how often we have a false "no click" (non-vacuum signals are not detected), the dark count rate p_{dark}, which indicates how often we have a false "click" (vacuum signals cause a click) and more imperfections such as the dead time of afterpulses. For now we will just consider the detection efficiency and the dark count rate. Therefore we can characterise the detectors with the following two operators.

$$D_{\text{no click}} = (1 - p_{\text{dark}}) \sum_{n=0}^{\infty} (1 - \eta_{\text{det}})^n |n\rangle\langle n| \tag{4.8}$$

$$D_{\text{click}} = 1 - D_{\text{no click}}. \tag{4.9}$$

These really indicate the probability of receiving no click, which is every time we have no dark count times the probability for every n-photon state that it is not detected. And of course the probability of receiving a click is all the other times.

Before a photon falls into one of our detectors, we want to send it through a so called polarised beam splitter to separate the orthogonal polarisation of a light pulse into different spatial modes. The working of the polarised beam splitter is captured in the following two equalities

$$\begin{aligned} a_H^\dagger &= c_H^\dagger \\ a_V^\dagger &= d_V^\dagger. \end{aligned} \tag{4.10}$$

That way, if we detect a signal from one of the output modes of the polarised beam splitter, we will know the polarisation of the photon.

4.1.1. VARIABLES OF THE QKD SYSTEM

As we have seen, the real components of the setup cause realistic QKD systems to deviate from ideal protocols. Since Alice and Bob still want to be able to achieve practical security, need to consider all possible deviations and imperfections. To find a secret key rate for realistic devices we will follow the idea of Gottesman, Lo, Lütkenhaus and Preskill (GLLP).
A possible way to characterise their devices to find the deviation from ideal QKD, is for Alice and Bob to do so called virtual measurements on the devices per each round to see whether it works like an ideal device or if it does an orthogonal operation. The related sifted key bit can then be marked as either a "good" bit or a "bad" bit respectively. When the "good" bits are mixed with "bad" ones, we still want to be able to extract a secret key. The GLLP security analysis tells us how to do this (Xu et al., 2020, p. 32).
If $1 - \Delta$ is the ratio of Bob's detected bits that are the "good" bits, then the secure key rate formula of GLLP is given by

$$R \geq (1 - \Delta)[1 - H(e_p)] - H(E). \tag{4.11}$$

Here E is the total QBER, which we will explain shortly, and e_p is the phase error of the "good" bits. H is the binary entropy function. The secret key rate r may be interpreted as the number of secure bits of key we generate per pulse that Alice sends. This number is between 0 and 1 and determined by the amount of bits that are lost due to losses in the channel, detectors and other devices (we lost these as the "bad" bits), and bits that are lost in the process of making sure that the key is secret.

The QBER E is defined as the probability of having an error, given that we have a detection event, or the fraction of wrong bits to total received bits.

The Gain Q is the probability of having a detection event, when Alice sent a state. Now given that we know that Alice sends n-photon states that are Poisson distributed, we can express Q as follows

$$\begin{aligned} Q &= \sum_{n=0}^{\infty} p[\text{send } n\text{-photon state and the state is detected}] \\ &= \sum_{n=0}^{\infty} p[\text{send } n\text{-photon state}]\, p[\text{detection event} \mid \text{an } n\text{-photon state was sent}] \\ &= \sum_{n=0}^{\infty} P_\mu(n)\, Y_n = \sum_{n=0}^{\infty} e^{-\mu} \frac{\mu^n}{n!} Y_n \end{aligned} \tag{4.12}$$

where we used in the second equality that for conditional probabilities $p[A \cap B] = p[B]\, p[A|B]$ In this definition we see that the yield Y_n of an n-photon state is the conditional probability of a detection event in Bob's setup given that Alice sent an n-photon state.

In the same way we can find an expression for E

$$\begin{aligned} E &= p[\text{error} \mid \text{detection event}] \\ &= \frac{p[\text{error and detection event}]}{p[\text{detection event}]} = \frac{\sum_{n=0}^{\infty} p[\text{error and } n\text{-photon state is sent and detected}]}{\sum_{n=0}^{\infty} p[\text{send } n\text{-photon state and the state is detected}]} \\ &= \frac{1}{Q} \sum_{n=0}^{\infty} p[\text{error and } n\text{-photon state is sent and detected}] \\ &= \frac{1}{Q} \sum_{n=0}^{\infty} p[\text{send } n\text{-photon state and the state detected}] \\ &\qquad p[\text{error} \mid \text{send } n\text{-photon state and the state detected}] \\ &= \frac{1}{Q} \sum_{n=0}^{\infty} P_\mu(n)\, Y_n e_n = \frac{1}{Q} \sum_{n=0}^{\infty} e^{-\mu} \frac{\mu^n}{n!} Y_n e_n \end{aligned} \tag{4.13}$$

and we can define the photon number error rate e_n, which is the probability of an error occurring, given that an n-photon state is sent and detected.
The gain Q and QBER E can be seen as properties of our system.

MODEL OF THE QKD SYSTEM

We will now derive expressions for Q and E using the characteristics from the model of our system. To simplify our calculations, we represent the setup on Bob's side including the quantum channel as is shown in Figure 4.2. This means that we only consider two detectors that have the same efficiency. We represent the quantum channel and the collection of Bob's devices as beam splitters. In this model we use an active polarisation mode shifter to indicate whether we have rectilinear or diagonal polarisations and shift all of them to rectilinear. And lastly we have a polarised beam splitter to distinguish between the orthogonal polarisation modes, H and V. This setup is simplified, but it is equivalent and can be easily adapted to one that is used in practical QKD, that usually involves more detectors. The exact workings of a practical setup is beyond the scope of this report.

Note that when a photon has reached the active polarisation shifter, the only losses that could still be caused are by the efficiency of the detectors. Therefore, for our calculations, we move this efficiency to the beam splitter that models all the losses due to Bob's devices. For the parameter of that we then get $\eta_B = \eta_{\text{det}}\eta_s$, where η_s indicates all the losses due to Bob's other devices.

Now that we have moved this efficiency η_{det} out of the actual detectors, we may give a new expression for our detection operators in Equations 4.8 and 4.9:

$$D_{\text{no click}} = (1 - p_{\text{dark}})|0\rangle\langle 0| \tag{4.14}$$

$$D_{\text{click}} = 1 - D_{\text{no click}}. \tag{4.15}$$

The no click event is now only determined by the probability of not having a dark count for vacuum.

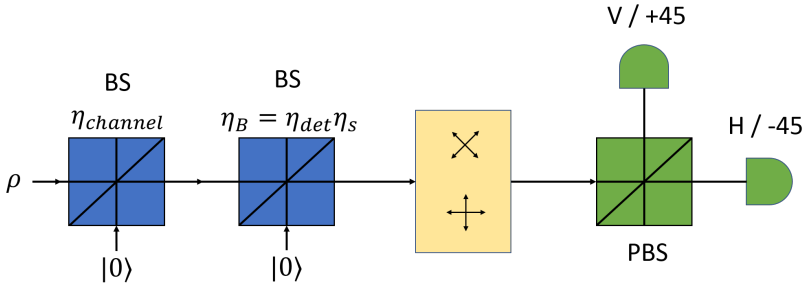

Figure 4.2: Model of the QKD system used. ρ is the quantum states of the qubit(s) that enter the channel. The channel is modelled by the first beam splitter (BS) with efficiency η_{channel}. The second beam splitter models the losses due to Bob's devices (η_B) which consist of the efficiency of the detectors (η_{det} and of the other setup components (η_s). The third block is the active polarisation shifter, and in the fourth component we find the polarised beam splitter and the two threshold detectors.

To find Q we need to know the yield, Y_n, the probability that Bob detects a signal, given that Alice sent an n-photon state, as a result of losses in the channel. In Figure 4.2 we see that we have losses in the channel and due to Bob's devices. We can combine these two components into one beam splitter with efficiency $\eta_{\text{sys}} = \eta_B\eta_{\text{channel}}$. This is then the only component that will cause losses. Note that for simplicity we can also combine the two detectors. For the new combined detector we will then again have to define new detection operators. Equation 4.14 becomes $D_{\text{no click}} = (1 - p_{\text{dark}})^2|0\rangle\langle 0|$ to account for dark counts in both detectors. The new model is shown in Figure 4.3.
If we now send a n-photon number state as given in Equation 4.3 through the system, we obtain with the beam splitter relations:

$$|n\rangle_a = \frac{1}{\sqrt{n!}}(a^\dagger)^n|0\rangle \quad \xrightarrow[\text{BS}]{} \quad |n\rangle_{cd} = \frac{1}{\sqrt{n!}}\left(\sqrt{\eta_{\text{sys}}}c^\dagger + \sqrt{1-\eta_{\text{sys}}}d^\dagger\right)^n|0\rangle \tag{4.16}$$

$$|n\rangle_{cd} = \sum_{k=0}^{n}\sqrt{\binom{n}{k}}\sqrt{\eta_{\text{sys}}}^{n-k}\sqrt{1-\eta_{\text{sys}}}^{k}|n-k,k\rangle_{cd} \tag{4.17}$$

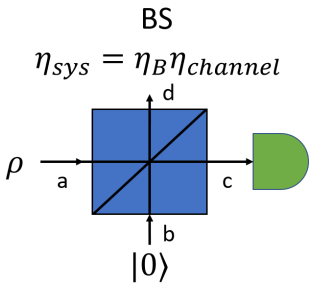

Figure 4.3: Simplified model of the QKD system used to calculate the expression for Q. All the factors that induce losses of photons are captured in one beam splitter with efficiency η_{sys}. The two detectors are combined into one detector at output c. Input a takes the state sent by Alice, input b is a vacuum, output d are the losses.

where we used that

$$|n-k\rangle_c = \frac{1}{\sqrt{(n-k)!}}(c^\dagger)^{n-k}|0\rangle \quad \text{and} \quad |k\rangle_d = \frac{1}{\sqrt{k!}}(d^\dagger)^k|0\rangle. \tag{4.18}$$

The yield can then be calculated:

$$\begin{aligned} Y_n =& \operatorname{Tr}(|n\rangle_{cd}\langle n|(D_{\text{click}} \otimes I_d)) && (4.19)\\ =& 1 - \operatorname{Tr}(|n\rangle_{cd}\langle n|(D_{\text{no click}} \otimes I_d)) && (4.20)\\ =& 1 - (1-p_{\text{dark}})^2 \operatorname{Tr}(|n\rangle_{cd}\langle n|(|0\rangle_c\langle 0| \otimes I_d)) && (4.21)\\ =& 1 - (1-p_{\text{dark}})^2(1-\eta_{\text{sys}})^n, && (4.22) \end{aligned}$$

where I_d is the identity matrix of the dimension of d and the detector operators are applied to the output state c, since we do not care what happens to output d, since those are the losses. In the last equality we used that the n-photon number states are orthogonal. As can be expected, the yield is 1 minus the probability of not having a detection event, i.e. that there is no dark count in both the detectors times the probability that n photons are not detected. We obtain Q by simply filling in the yield into equation 4.12:

$$Q = 1 - (1-p_{\text{dark}})^2 e^{-\mu\eta_{\text{sys}}}. \tag{4.23}$$

Figure 4.4 shows a plot of the gain Q against the transmission distance in km, for two different values of the loss coefficient α of the quantum channel. As we expect, the gain decreases over the distance, due to more and more losses that occur. This effect is enhanced if α has a bigger value.

To calculate the QBER E we need to consider photons that do give a click but in the wrong detector. To model this, we will add a misalignment in the channel to the model in Figure 4.2. Its input and output relations are given by

$$\begin{pmatrix} c_H^\dagger \\ c_V^\dagger \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} e_H^\dagger \\ e_V^\dagger \end{pmatrix}. \tag{4.24}$$

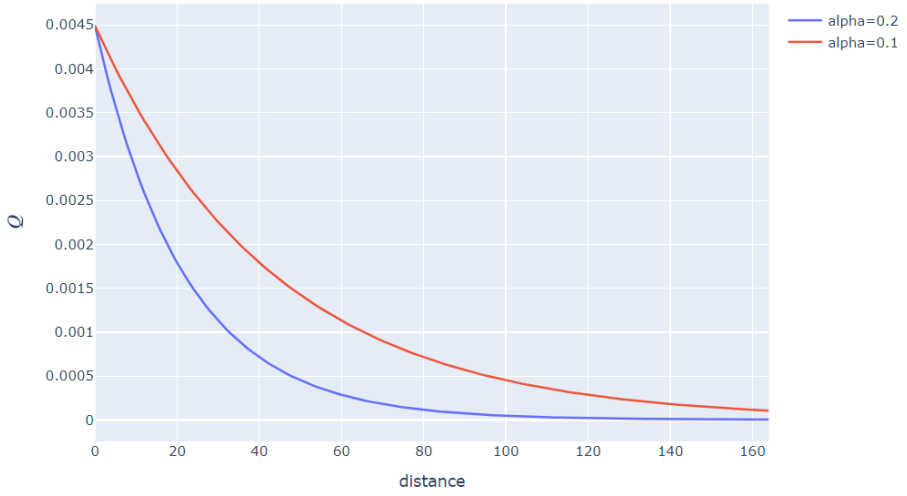

Figure 4.4: Plot of the gain Q against the distance in km, for two different values of α. The gain decreases exponentially with distance. If the loss coefficient α is bigger, Q decreases faster.

We again combine the channel and Bob's devices into one beam splitter with efficiency η_{sys}. To calculate the error rate we will look at what happens to a horizontally aligned state that enters the channel. The adapted model that we use to calculate E is given in Figure 4.5. We see that essentially our state splits into three beams: one is lost in d, one falls onto the first detector D_1 and one onto the second D_2. Recall that because of the polarised beam splitter the horizontally aligned signals should fall onto D_1 and the vertically aligned signals should fall onto D_2. We can then find the probability that a horizontally aligned n-photon signal produces a detected event associated with an error, this is the product $Y_n e_n$, as the following trace

$$Y_n e_n = \text{Tr}\left(\left[D_{1,\text{no click}} \otimes D_{2,\text{click}} \otimes I_d + \frac{1}{2} D_{1,\text{click}} \otimes D_{2,\text{click}} \otimes I_d\right] |n\rangle_{dfg}\langle n|\right) \tag{4.25}$$

where the detection operators are the ones from Equations 4.14 and 4.15, since we again use two detectors. We get the second term since we associate the possible event of a double click (in both detectors) as a random single click with uniform distribution. The state $|n\rangle_{dfg}\langle n|$ is the total output state of modes d, f and g.
To find the total output state $|n\rangle_{dfg}\langle n|$, we will do the same calculations as we did for Q but with the components in Figure 4.5. We start with our input state $\rho_H = |n\rangle_H\langle n|$, with $|n\rangle_H = \frac{1}{\sqrt{n!}}(a_H^\dagger)^n|0\rangle$. Since we know from Equations 4.16 and 4.17 what the states will look like, we will here only give the creation operators of the input and output modes as a result of the different

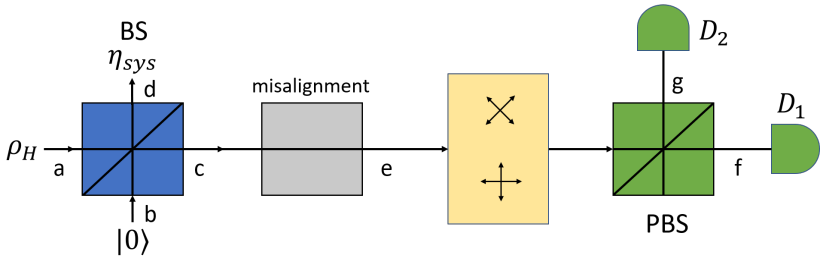

Figure 4.5: Adapted model to calculate the QBER E of the system. The horizontally aligned state in a first passes through the channel and Bob's devices, represented by a beam splitter, and some information is lost at output d. Next the state from output c will undergo misalignment in the channel, resulting in the state in e. Next after the polarisation shifter and the polarised beam splitter, the state will end up in one of the detectors D_1 or D_2 and give a click.

operations of the components.

$$a_H^\dagger \xrightarrow[\text{BS}]{} \sqrt{\eta_{\text{sys}}}c_H^\dagger + \sqrt{1-\eta_{\text{sys}}}d_H^\dagger \tag{4.26}$$

$$\xrightarrow[\text{misalignment}]{} \sqrt{\eta_{\text{sys}}}\left(\cos\theta e_H^\dagger - \sin\theta e_V^\dagger\right) + \sqrt{1-\eta_{\text{sys}}}d_H^\dagger \tag{4.27}$$

$$\xrightarrow[\text{PBS}]{} \sqrt{\eta_{\text{sys}}}\left(\cos\theta f_H^\dagger - \sin\theta g_V^\dagger\right) + \sqrt{1-\eta_{\text{sys}}}d_H^\dagger. \tag{4.28}$$

Now we can express the total state of output modes d, f and g in the same way as we did in Equation 4.17 with the help of Equation 4.18. This total state is then

$$|n\rangle_{dfg} = \sum_{k=0}^{n}\sum_{l=0}^{n-k}\sqrt{\frac{n!}{k!l!(n-k-l)!}}\sqrt{\eta_{\text{sys}}}^{n-k}\sqrt{1-\eta_{\text{sys}}}^{k}(\cos\theta)^{n-k-l}(-\sin\theta)^l|k,n-k-l,l\rangle_{d_H,f_H,g_V}. \tag{4.29}$$

We can fill in this state in Equation 4.25, but before we do that, we first want to write out the term in square brackets from that equation using the detection operators 4.14 and 4.15, for arbitrary input (either horizontal or vertical). This term is then

$$\frac{1}{2}\left(I_{dfg} + (1-p_{\text{dark}})(I_d\otimes|0\rangle_f\langle 0|\otimes I_g - I_d\otimes I_f\otimes|0\rangle_g\langle 0|) - (1-p_{\text{dark}})^2(I_d\otimes|0\rangle_f\langle 0|\otimes|0\rangle_g\langle 0|)\right). \tag{4.30}$$

We obtain:

$$Y_n e_n = \frac{1}{2}\Bigg(1 + (1-p_{\text{dark}})\frac{1}{2^n}\left[\left(2-\eta_{\text{sys}}-\eta_{\text{sys}}\cos 2\theta\right)^n - \left(2-\eta_{\text{sys}}+\eta_{\text{sys}}\cos 2\theta\right)^n\right] - (1-p_{\text{dark}})^2(1-\eta_{\text{sys}})^n\Bigg). \tag{4.31}$$

We can now substitute this in Equation 4.13. The error rate that is observed in the experiment is given by the following expression:

$$E = \frac{1}{2Q}\left(1 + (1-p_{\text{dark}})\left(e^{-\mu\eta_{\text{sys}}\cos^2\theta} - e^{-\mu\eta_{\text{sys}}\sin^2\theta}\right) - (1-p_{\text{dark}})^2 e^{-\mu\eta_{\text{sys}}}\right). \tag{4.32}$$

It depends again on the dark count rate, but also on the intensity of the laser and the misalignment in the channel, given by θ.

4.2. IMPLEMENTATION OF THE BB84 PROTOCOL

Now that we have characterised all of the setup components, we can start to find the key rate for an asymptotic regime model. We want to get a secret key rate such as the one given in Equation 4.11, but now for a source of weak coherent pulses and the characteristics of the model that we just derived. The "good" bits are now only the ones that are "good" according to the description of GLLP and that are in a single photon state, since must take into account that Eve will use a photon number splitting attack. Therefore we may only use information from single photon signals. The ratio $(1-\Delta)$ should then be $p_1 Y_1$, the probability of Alice sending a single photon state, $p_1 = \mu e^{-\mu}$, times the yield of a single photon. The phase error of the "good" bits should also be the single photon error rate, e_1. We also need to multiply the entropy of the QBER E with the gain Q to account for the probability of detecting a photon. Lastly, we want to multiply the whole expression with q, which is the probability that Alice and Bob use the same basis to measure the qubit. This factor is imposed by using the BB84 protocol. The secret key rate in bits per pulse (bpp) for the BB84 protocol with weak coherent pulses is then:

$$R \geq q\left(p_1 Y_1 (1 - H(e_1)) - Q H(E)\right). \tag{4.33}$$

Two unknowns are Y_1, the yield of the single photon states, and e_1, the phase error of the single photon states

Since we still have limited information about these two values, we need to assume the worst-case scenario, that all the errors are in the single photon states. We give two alternative expressions of Q and E in Equations 4.12 and 4.13:

$$Q = p_0 Y_0 + p_1 Y_1 + p_{\text{multi}} Y_{\text{multi}}, \tag{4.34}$$

$$EQ = p_0 Y_0 e_0 + p_1 Y_1 e_1 + p_{\text{multi}} Y_{\text{multi}} e_{\text{multi}}, \tag{4.35}$$

where $p_n = e^{-\mu}\frac{\mu^n}{n!}$ is the probability of emitting an n-photon state, thus $p_{\text{multi}} = 1 - e^{-\mu} - \mu e^{-\mu}$ is the probability of emitting more than one photon, i.e. one minus the probability of emitting zero or one photons, and Y_{multi} and e_{multi} are the according multi photon yield and error rate. Now we take the yield of vacuum to be $Y_0 = 0$, and for the worst-case scenario let $Y_{\text{multi}} = 1$ and $e_{\text{multi}} = 0$. Then we obtain for Y_1 and e_1

$$Y_1 = \frac{Q - p_{\text{multi}}}{p_1} \tag{4.36}$$

$$e_1 = \frac{E}{1 - \frac{p_{\text{multi}}}{Q}}. \tag{4.37}$$

Now we use the following parameters to find the gain Q and the QBER E of the system: $p_{\text{dark}} = 10^{-6}$, $\eta_B = 0.045$, $\alpha = 0.2$, $\sin^2\theta = 0 \wedge 0.015$, $q \approx 1$, $\mu = \eta_{\text{sys}}$.
Note that to get the best optimised outcome we set the intensity of the source μ to be equal to the efficiency of the system η_{sys}, which is distance dependent, since it is proportional to

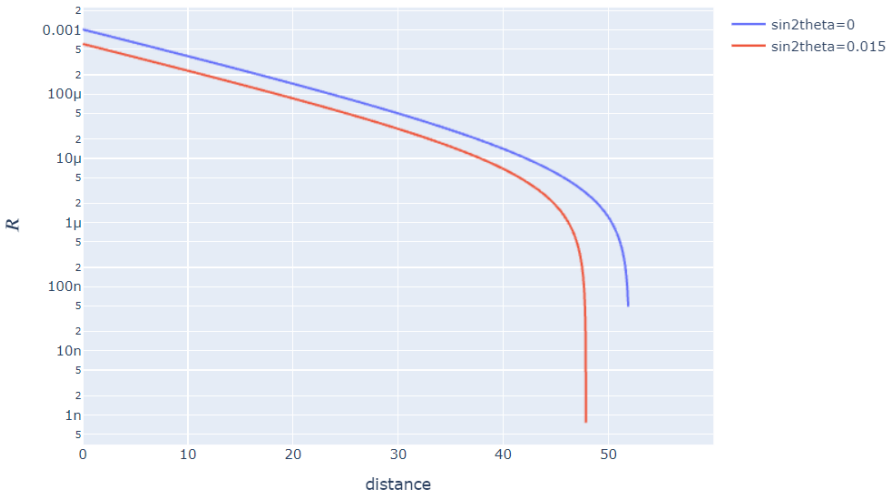

Figure 4.6: Plot of the secret key rate R in bits per pulse for the BB84 protocol with weak coherent pulses against the distance in km for two different values of $\sin^2\theta$. The vertical axis has a logarithmic scale (with base 10). The secret key rate decays exponentially over distance and drops to zero at 48 km for $\sin^2\theta = 0.015$ and at 52 km for $\sin^2\theta = 0$.

η_{channel}. So every time we calculate the secret key rate for a certain distance, we use an optimal μ for that distance. A plot of the resulting secret key rate in bpp against distance in km is given in Figure 4.6 for two different values of $\sin^2\theta$, the case that there is zero misalignment in the channel ($\sin^2\theta = 0$) and when there is some misalignment in the channel ($\sin^2\theta = 0.015$). For $\sin^2\theta = 0$ we can generate secret key rate up to distances of 52 km. For $\sin^2\theta = 0$ we are able to generate secret key for distances up to 48 km. The best value of secret key rate, for 0 distance, is $R = 1.02 \cdot 10^{-3}$ bpp and $R = 6.05 \cdot 10^{-4}$ bpp respectively for zero and some misalignment. As is to be expected, the secret key rate is better when there is no misalignment in the channel.

4.3. THE DECOY STATE METHOD

Because of the photon number splitting attack, Alice and Bob may only use information for the key that is sent in single photon signals. In the last model we needed to use worst-case scenario approximations for Y_1 and e_1. But there is a better way to approximate these numbers.

With the decoy state method, Alice will use the same source that emits weak coherent states, but she will not use only one intensity for encoding her bits and sending them to Bob, she will use some additional intensities as decoy states. After Bob detected all the signals, Alice will use the classical channel to announce for each pulse which intensities she used. Alice and Bob can then easily characterise the detection rates of signal and decoy states.

The photon source is thus operated at different settings for the mean photon number μ, leading to different measurement outcome statistics. The decoy state method has one very crucial assumption, which is that the detection rates for the signal states and for the decoy states are

the same. The consequence of this for the photon number splitting attack, is that after Eve did her photon number measurement, it is impossible for her to tell if the n-photon state is from the signal state or one of the decoy states. When Eve simply catches away some of the photons in the intercepted states, the detection rates of the signal and decoy states are changed in different ways per kind of state. Therefore, in order to preserve the ratio of detection rates among signal and decoy states, Eve will have to let through a significant amount of quantum states, since she cannot know the intensity beforehand (Xu et al., 2020, p. 32).

The signals that Alice will be able to send are given by

$$\rho_l = e^{-\mu_l} \sum_{n=0}^{\infty} \frac{\mu_l^n}{n!} |n\rangle\langle n|, \qquad l \in \{s, d_1, d_2, \ldots, d_N\}, \tag{4.38}$$

where $s, d_1, d_2, \ldots, d_N$ indicate the different intensity settings for the signal state and N decoy states respectively. Alice chooses between the different intensity settings randomly.
Since the states from different intensity settings are identical except for their average photon numbers, the yield Y_n and the error rate e_n only depend on the photon number n, but not on which distribution (decoy or signal) the state is from:

$$\begin{aligned} Y_n(\text{signal}) &= Y_n(\text{decoy}), \\ e_n(\text{signal}) &= e_n(\text{decoy}). \end{aligned} \tag{4.39}$$

We define the secret key rate in bits per pulse (bpp) for the BB84 protocol with weak coherent pulses and decoy states fairly similar to the model without decoy states:

$$R \geq q\left(p_{1|s} Y_1 (1 - H(e_1)) - Q_s H(E_s)\right) \tag{4.40}$$

where now we have $p_{1|s} = \mu_s e^{-\mu_s}$ is the probability that Alice prepares a single-photon state with the signal setting. Therefore also we use Q_s and E_s, the gain and QBER observed in the experiment when using the signal state. It should be observed that it is also possible to define a secret key rate based on observations of signals that were sent in a decoy state instead of the signal state. However, by design of the model, the signal setting is the preferred setting to generate key bits and decoy states are simply meant as decoy. That is why we will also base our secret key rate on information from the signal states.
To estimate the single photon yield Y_1 and error e_1, Alice and Bob can solve two systems of linear equations provided by the observed gains and QBERs that are related to the different settings $s, d_1, d_2, \ldots, d_N$. Applying again the Equations 4.12 and 4.13 but now for each intensity setting we obtain:

$$
\begin{aligned}
Q_s &= \sum_{n=0}^{\infty} e^{-\mu_s} \frac{\mu_s^n}{n!} Y_n \\
Q_{d_1} &= \sum_{n=0}^{\infty} e^{-\mu_{d_1}} \frac{\mu_{d_1}^n}{n!} Y_n \\
&\vdots \\
\underbrace{Q_{d_N}}_{\text{observed}} &= \underbrace{\sum_{n=0}^{\infty} e^{-\mu_{d_N}} \frac{\mu_{d_N}^n}{n!}}_{\text{known}} \underbrace{Y_n}_{\text{unknown}}
\end{aligned}
$$

and

$$
\begin{aligned}
E_s Q_s &= \sum_{n=0}^{\infty} e^{-\mu_s} \frac{\mu_s^n}{n!} Y_n e_n \\
E_{d_1} Q_{d_1} &= \sum_{n=0}^{\infty} e^{-\mu_{d_1}} \frac{\mu_{d_1}^n}{n!} Y_n e_n \\
&\vdots \\
\underbrace{E_{d_N} Q_{d_N}}_{\text{observed}} &= \underbrace{\sum_{n=0}^{\infty} e^{-\mu_{d_N}} \frac{\mu_{d_N}^n}{n!}}_{\text{known}} \underbrace{Y_n e_n}_{\text{unknown}}
\end{aligned}
$$

Here, the subscripts of E, Q and μ indicate which intensity setting is used; signal or one of the decoys. Because of the infinite sum over n, the number of unknowns Y_n and e_n in each equation is infinite. We will truncate the number of unknowns to make the relations useful. If we limit the unknown yields to say $Y_1, \ldots, Y_{M_{\text{cut}}}$ then we can get upper and lower bounds on the gain:

$$
Q_l \geq \sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!} Y_n, \qquad l \in \{s, d_1, d_2, \ldots, d_N\}, \tag{4.41}
$$

$$
\begin{aligned}
Q_l &\leq \sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!} Y_n + \sum_{n=M_{\text{cut}}+1}^{\infty} e^{-\mu_l} \frac{\mu_l^n}{n!} \\
&= \sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!} Y_n + 1 - \sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!}, \qquad l \in \{s, d_1, d_2, \ldots, d_N\}.
\end{aligned} \tag{4.42}
$$

The same can be done for $Y_n e_n$ in the equations of $E_l Q_l$, $l \in \{s, d_1, d_2, \ldots, d_N\}$.
Observe that we now have a set of linear inequality constraints that we can use to bound Y_1 from below:

$$
\begin{aligned}
\min \quad & Y_1 \\
\text{s.t.} \quad & Q_l \geq \sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!} Y_n, \qquad \forall l \in \{s, d_1, d_2, \ldots, d_N\} \\
& Q_l \leq \sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!} Y_n + 1 - \sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!}, \qquad \forall l \in \{s, d_1, d_2, \ldots, d_N\} \\
& 1 \geq Y_n \geq 0, \qquad \forall n \in \{0, \ldots, M_{\text{cut}}\}
\end{aligned}
$$

and if we let $\gamma_n = Y_n e_n$ we also have a set of linear equations to bound γ_1 from above:

$$
\begin{aligned}
\max \quad & \gamma_1 \\
\text{s.t.} \quad & E_l Q_l \geq \sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!} \gamma_n, \qquad \forall l \in \{s, d_1, d_2, \ldots, d_N\} \\
& E_l Q_l \leq \sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!} \gamma_n + 1 - \sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!}, \qquad \forall l \in \{s, d_1, d_2, \ldots, d_N\} \\
& 1 \geq \gamma_n \geq 0, \qquad \forall n \in \{0, \ldots, M_{\text{cut}}\}
\end{aligned}
$$

And from γ_1 we can then get an upper bound on e_1:

$$e_1 \leq \frac{\gamma_1}{Y_1}.$$

We will solve this minimisation and maximisation problem using linear optimisation, as was introduced in Section 2.4. We will write the objective function and the constraints in the standard form given in Equations 2.32 through 2.34. For convenience, we will repeat them here.

Standard form linear program

$$
\begin{aligned}
&\text{minimise} && \mathbf{c}^\top \mathbf{x} && (4.43)\\
&\text{subject to} && A\mathbf{x} \leq \mathbf{b} && (4.44)\\
& && \mathbf{x} \geq \mathbf{0} && (4.45)
\end{aligned}
$$

So for example, if we want to write the linear program to find Y_1 as given above, we use that

$$
\begin{aligned}
&\mathbf{x} = \begin{bmatrix} Y_0 & Y_1 & Y_2 & \ldots & Y_{M_{\text{cut}}} \end{bmatrix}^\top, \\
&\mathbf{c} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \end{bmatrix}^\top \text{ and} \\
&\mathbf{b} = \Big[Q_s \quad Q_{d_1} \quad Q_{d_2} \quad \ldots \quad Q_{d_N} \\
&\quad -\left(Q_s - 1 + \textstyle\sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!}\right) \quad -\left(Q_{d_1} - 1 + \textstyle\sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!}\right) \quad \ldots \quad -\left(Q_{d_N} - 1 + \textstyle\sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!}\right) \Big]^\top.
\end{aligned}
$$

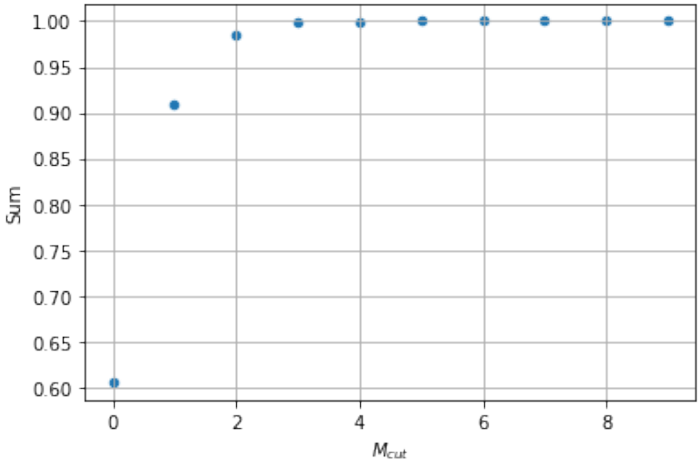

Figure 4.7: Scatter plot of $\sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!}$ for different values of M_{cut}. The saturation point of the sum lies at $M_{\text{cut}} = 4$. Summing up to a higher value will not give a significantly better result.

Then **x** and **c** are vectors of length $M_{\text{cut}} + 1$ and **b** is a vector of length $2(N+1)$. The matrix A consists of the matrices A_1 and $-A_1$ stacked on top of each other as follows,

$$A = \begin{bmatrix} A_1 \\ -A_1 \end{bmatrix},$$

where A_1 has dimensions $(N+1) \times (M_{\text{cut}} + 1)$ and has matrix elements $A_{1(l,n)} = e^{-\mu_l} \frac{\mu_l^n}{n!}$.
It is easy to see that this linear program is exactly the same as the linear equations we gave earlier to bound Y_1.
In a similar manner we can write a linear program for γ_1.
We will solve these two linear programs numerically in Python using the linprog functions from the scipy.optimize library. In the model we used for the different intensity settings signal and 3 decoys, thus $N = 3$. The respective values for the average photon number for these intensities are $\mu = \{0.5, 0.05, 0.005, 0.0005\}$. To find a suitable value for M_{cut}, not too large that the program would demand a lot of computing power, but not too small to be inaccurate, the sum $\sum_{n=0}^{M_{\text{cut}}} e^{-\mu_l} \frac{\mu_l^n}{n!}$ for different values of M_{cut} were calculated and the result is given in Figure 4.7. It can be seen that summing to a value higher than $M_{\text{cut}} = 4$ will not give a significantly better estimation of the infinite sum. Therefore we take $M_{\text{cut}} = 4$ for our model.

The results of the two linear models Y_1 and e_1 can now be used to generate the secret key rate for the decoy state model. We calculate Q_s and E_s with the same parameters as before, when we had no decoy states. But of course this time we use μ_s the intensity for the signal setting. So, $p_{\text{dark}} = 10^{-6}$, $\eta_B = 0.045$, $\alpha = 0.2$, $\sin^2\theta = 0 \wedge 0.015$, $q \approx 1$, $\mu_s = 0.5$.

A plot of the secret key rate R in bpp for BB84 with weak coherent pulses and decoy states

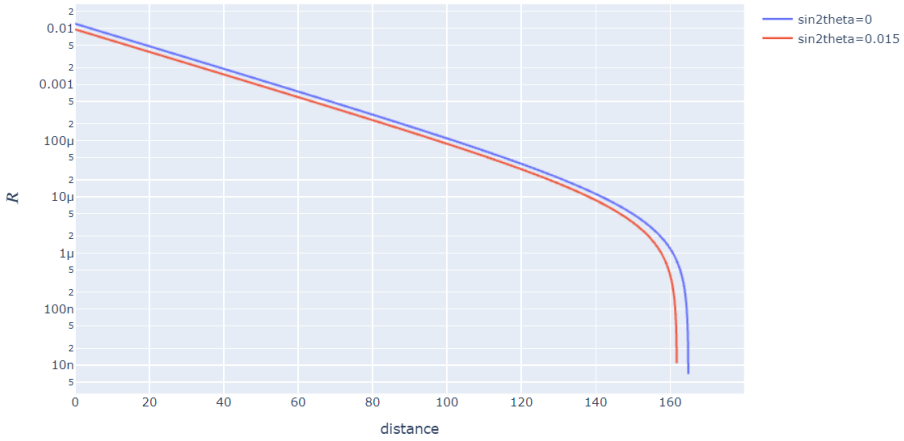

Figure 4.8: Plot of the secret key rate R in bpp for the BB84 protocol with weak coherent pulses and decoy states against the distance in km for two different values of $\sin^2\theta$. The vertical axis has a logarithmic scale (with base 10). The secret key rate decays exponentially over distance and drops to zero at 162 km for $\sin^2\theta = 0.015$ and at 165 km for $\sin^2\theta = 0$.

against the distance in km for two values of $\sin^2\theta$ is given in 4.8. For no misalignment, $\sin^2\theta = 0$, it is possible to generate secret key rate up to distances of 165 km. When there is misalignment in the channel, $\sin^2\theta = 0.015$, it is possible to generate secret key rate up to 162 km of distance. The best value of secret key rate, for 0 distance, is $R = 1.21 \cdot 10^{-2}$ bpp and $R = 9.60 \cdot 10^{-3}$ bpp respectively for zero and some misalignment.
If we compare this result with the secret key rate for the model that did not use decoy states, we see that the distance over which we are still able to generate secret key has more than tripled, both with and without misalignment in the channel. The highest value of secret key rate, at a zero distance, is also roughly ten times as large for the decoy state model. The use of decoy states in practical QKD is thus a drastic improvement on the amount of secret key and the distance over which it can be transmitted.

5

MEASUREMENT DEVICE INDEPENDENT QKD

In the previous chapter we wanted to fully characterise all of our devices to be able to extract secure key, even though the eavesdropper was attacking us through our devices' imperfections. However, fully describing all of the practical components is very complex and limited by our understanding of the devices.
In this chapter we will discuss an alternative protocol for quantum key distribution, measurement device independent QKD. As can be guessed from the name, the security of this protocol does not depend on the (non-ideal) properties of the measurement devices used and therefore eliminates the possibility of all detection attacks. It offers a security advantage, since we no longer require detailed knowledge of the measurement devices. Moreover, MDI-QKD is practical with current technology, making it a very promising protocol for performing practical quantum key distribution.
We will present a description of the protocol, its main advantages as opposed to BB84 and we will give an expression for the secret key rate that can be achieved with this protocol

5.1. THE MDI-QKD PROTOCOL

The protocol for MDI-QKD is based on a "time-reversed" entanglement protocol. Notably, it leaves all of the detections to a public, not necessarily trusted party, Charlie (Xu et al., 2015, p. 3, Xu et al., 2020, p. 36).
The entanglement protocol with a third party is different from entanglement based BB84 that we presented in Section 3.3. In this entanglement protocol, Alice and Bob both individually prepare an entangled qubit pair in one of the four Bell states (given in Equation 3.10). They both send one half of their pair to the center party Charlie. He will then perform a projective Bell state measurement on both qubits he receives. This measurement is a joint measurement on both qubits that returns which of the four Bell states the qubits are in. The result is then publicly announced. After that announcement, Alice and Bob measure the other photon of their entangled pairs locally in either the X- or Z-basis, which they choose at random. After

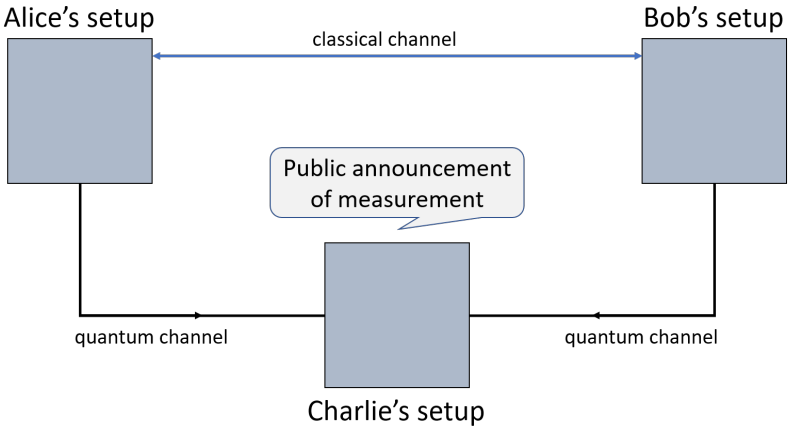

Figure 5.1: Diagram of the total setup for QKD with a third party, Charlie, who receives all the qubits and measures them. Charlie publicly announces his results and Alice and Bob are able to compare their results through a classical channel.

all transmissions and measurements, Alice and Bob will use a classical, authenticated channel to compare a subset of their measurement outcomes to know whether Charlie is honest.
A diagram of this setup is given in Figure 5.1.
Alice and Bob can now communicate if they sent states that were either one of $|\Phi^{\pm}\rangle$ or one of $|\Psi^{\pm}\rangle$. With the outcome of Charlies Bell state measurement on the combination of their qubits, it will turn out that they can deduce the parity of the bit they have. With post-processing steps, they can now generate a secret key (Xu et al., 2020, p. 36).

Note that this entanglement protocol can also work in a time reversed version. Alice and Bob first measure their local qubit, then wait for Charlie's announcement. This is possible since Charlie's operations commute with those of Alice and Bob (Xu et al., 2015, p. 3). And since Charlie's measurement is only for checking the parity of the bits sent by Alice and Bob, information about the bit values stays secret. With this time reversal, the entanglement protocol can be seen as a prepare-and-measure protocol where Alice and Bob directly send one of the BB84 states $(|0\rangle, |1\rangle, |+\rangle, |-\rangle)$ to Charlie and he measures them. This forms the main concept behind MDI-QKD.

One advantage of MDI-QKD is the following: Consider the case that Charlie cannot be trusted and he will perform a measurement on the qubits that is different from the ideal Bell state measurement. As Alice and Bob will check afterwards if Charlie did the right measurement and they will only use information from successful events, this will just lead to an increased error rate. An increased error rate will then lead to a smaller key, but it is still secret, after privacy amplification is applied. It is of no importance whether the error is due to experimental imperfections or if it is caused by an eavesdropper (Eve or possibly Charlie) who replacing or modifying measurement devices in order to gain information about the qubits from Alice and Bob.

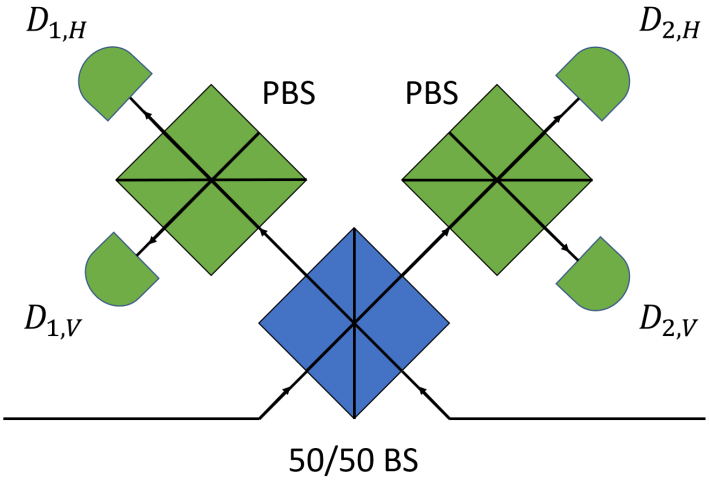

Figure 5.2: Schematic diagram of the measurement setup of Charlie. The two photons from Alice and Bob fall through a 50/50 beam splitter that has polarised beam splitters at its output modes. Which two detectors give a click, determines which Bell state is measured.

Another advantage of the MDI-QKD protocol, besides removing all detector side-channels, is that it takes into account the use of practical laser signals that employ decoy states, just as we did for the BB84 protocol to improve the generated secret key rate. Measurements are therefore again done by a combination of beam splitters and threshold detectors.
Before we present the MDI protocol, we will first show how Charlie makes the Bell state measurement on the photons he receives from Alice and Bob. He will use the setup that is given in Figure 5.2. The system consists of a 50/50 beam splitter (50/50 BS) that has polarised beam splitters (PBS) at its outputs. The outputs of the polarised beam splitters are then connected to the threshold detectors, $D_{1,H}, D_{1,V}, D_{2,H}, D_{2,V}$. The polarised beam splitter has been introduced before; it splits all incoming signals in two orthogonal modes, H and V. The 50/50 beam splitter has the following relation between the creation operators of its input and output modes:

$$\begin{pmatrix} a^\dagger \\ b^\dagger \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} c^\dagger \\ d^\dagger \end{pmatrix}, \tag{5.1}$$

and splits photons in the diagonal orientation from photons in the rectilinear orientation. When there is a click in the detectors $D_{1,H}$ and $D_{2,V}$, or in $D_{1,V}$ and $D_{2,H}$ then the photons are projected onto the state $|\Psi^-\rangle = (|01\rangle - |10\rangle)/\sqrt{2}$. If there happens to be a click in $D_{1,H}$ and $D_{1,V}$, or in $D_{2,H}$ and $D_{2,V}$, then this implies that the photons were projected onto $|\Psi^+\rangle = (|01\rangle + |10\rangle)/\sqrt{2}$. Only these two detection patterns are considered successful and will be used for the experiment. So clicks in any other combination of detectors are unsuccessful (Xu et al., 2020, p. 38).
We will now summarise the protocol (from Xu et al., 2015, p. 3 and Xu et al., 2020, p. 37-38):

1: Alice and Bob choose one of four BB84 states individually and at random and prepare a corresponding signal using phase-randomised WCPs together with decoy signals, just

as in the model we presented in Chapter 4. Then they send the states to an untrusted party, Charlie.

2: An honest Charlie performs a Bell state measurement. Projections onto the $|\Psi^+\rangle$ or $|\Psi^-\rangle$ state are successful, other detection patterns are considered unsuccessful.
3: Whether Charlie is honest or not, when he claims to obtain a successful measurement, he announces the outcome of his Bell state measurement using a classical public channel.
4: Alice and Bob only keep the bits that correspond to Charlie's successful measurement events, the rest is discarded. Next, similar to the sifting in BB84 protocol, Alice and Bob use the classical channel to communicate their basis choices and keep the bits where they used the same bases. Based on Charlie's measurement results, Alice flips part of her bits, according to the announcement from Charlie, to guarantee the correct correlation with those of Bob.
5: After sifting, the decoy-state method is used to estimate the gain and QBER of the single photon events.
6: Post-processing: Alice and Bob perform error correction and privacy amplification to generate the secret key.

In Step 4 Alice will flip part of her bits to guarantee the correct correlation with the bits of Bob. Note that whenever Charlie measured $|\Psi^-\rangle$, Alice and Bob will have sent anti-correlated qubits, if they used the same basis. So in case of this measurement, Alice will flip her bit. Now if Charlie measured $|\Psi^+\rangle$, Alice and Bob will again have anti-correlated bits if they both used the Z-basis to encode it. Alice will have to flip her bit. If they used the X-basis however, their bits are already the same, and Alice does nothing.

Though needed for implementations of MDI-QKD with WCPs, sending decoy states from two users to a common receiver is mathematically a little more trickier than before when only sending from Alice to Bob. However, it turns out that it is sufficient to have a tight estimation if Alice and Bob just use a few decoy settings each (Xu et al., 2020, p. 39).

5.2. GENERATING SECRET KEY WITH MDI-QKD

We will now look more closely into the security of the MDI-QKD protocol by deriving the secret key rate as given by Lo et al., 2012. For an unconditional security proof, we would like to refer to that paper.

Let us assume that Alice and Bob consider the data sent in two bases separately. The Z-basis is used for key generation, while the X-basis is used for testing for an eavesdropper only. For the signals that Alice and Bob send, we will denote the gain and QBER as $Q^Z_{n,m}, Q^X_{n,m}, E^Z_{n,m}$ and $E^X_{n,m}$ where n is the number of photons that Alice sent and m the number of photons Bob sent, and X and Z represents their basis choices.

Recall that when Alice and Bob both used the Z-basis and Charlie gave a successful measurement, Alice needed to flip her bit to correlate it with Bob's. Thus in the Z-basis, an error corresponds to a successful measurement from Charlie, but Alice and Bob did send the same polarisation state. Assuming ideal detectors and other optical elements and that there is no misalignment in the channel, we expect that whenever Alice and Bob indeed sent the same state, Charlie will never announce a successful measurement. Thus $E^Z_{n,m} = 0$ for all n, m.

When Alice and Bob both used the X-basis, Alice needed to flip her bit when Charlie announced the $|\Psi^-\rangle$ state and take no action when he announced the $|\Psi^+\rangle$ state. Thus an error occurs when Alice and Bob prepared the same state and Charlie announces $|\Psi^-\rangle$ or when they prepared orthogonal states and he announces $|\Psi^+\rangle$. The Hong-Ou-Mandel (HOM) effect tells us that whenever two identical single-photons enter a 50/50 beamsplitter, they will always exit both in the same output mode. Moreover, when the photons are in orthogonal polarisations and exit the beamsplitter in the same output arm, they will both reach the same detector. Thus we find $E^X_{1,1} = 0$ (from Lo et al., 2012).
Remarkably, thus, the use of WCP sources does not induce high errors. In the ideal scenario we have then simply the secret key rate in the asymptotic limit of an infinitely long key, $R = Q^Z_{1,1}$ i.e. the probability that Charlie announces a successful measurement when Alice and Bob both sent him a single photon in the Z-basis.

Now let us consider imperfections such as misalignment, dark counts and, possibly, an eavesdropper. Then, in this realistic setup, the secret key rate will be given by (from Lo et al., 2012),

$$R = Q^{1,1}_Z(1 - H(E^{1,1}_X)) - Q_Z f(E_Z) H(E_Z), \tag{5.2}$$

where $Q^Z = \sum_{n,m} Q^Z_{n,m}$, $E^Z = \sum_{n,m} Q^Z_{n,m} E^Z_{n,m}/Q^Z$ and $f(E^Z) > 1$ is an inefficiency function for the error correction process (Lo et al., 2012). In this equation $Q^Z_{1,1}$ and $E^X_{1,1}$ cannot be measured but need to be determined by the decoy state method. Just as for the decoy state analysis we did for BB84, we can find an upper bound for $E^X_{1,1}$ and a lower bound for $Q^Z_{1,1}$ by using the system of linear equations we get from the finite number of decoy states. This will then in turn give a lower bound for the secret key rate in Equation 5.2.

In conclusion, MDI-QKD is a very promising protocol for practical quantum key distribution. Since it eliminates all measurement devices in the analysis, it is for a security proof no longer needed to have very precise characterisations of these devices. This makes the security analysis simpler and more precise. It also employs decoy states, which increases the secret key rate and are relatively easy to implement in practice. MDI-QKD could well be a high-performance solution for future quantum communication networks.

6

CONCLUSION

In this report we reviewed two protocols for quantum key distribution: the BB84 protocol and the MDI-QKD protocol. The three goals of this report were to prove the unconditional security of the BB84 protocol in ideal environments, to generate the secret key rate for the practical application of the BB84 protocol, both with and without decoy states, and to review the MDI-QKD and look at the advantages it has for practical QKD.

The security of the BB84 protocol can be proved by designing an equivalent theoretical protocol and proving its security by bounding the information of the eavesdropper to an exponentially small number. The secure BB84 protocol is then derived from this theoretical protocol, by changing the order of operations without compromising their secure nature.
To generate secure key from a practical model with the BB84 protocol, it is necessary to precisely define the characteristics of the setup components. The secret key rate is calculated with variables derived from these characteristics to account for the devices' imperfections. The secret key rates of the model were plotted with and without misalignments in the quantum channel and for a model that does not employ decoy states in its signals and one that does.
We saw that by employing decoy states it was still possible to generate key rates for distances of 162 km and 165 km, with and without misalignment in the channel. Without the use of decoy states we could only reach a distance of 48 km and 52 km. Employing decoy states more than triples the communication distance. Also the best value of secret key rate, at zero distance, is an order of magnitude better for the decoy state method. $R = 1.21 \cdot 10^{-2}$ bpp and $R = 9.60 \cdot 10^{-3}$ bpp compared to $R = 1.02 \cdot 10^{-3}$ bpp and $R = 6.05 \cdot 10^{-4}$ bpp with no decoy states.
We conclude that the use of decoy states in practical QKD systems is a huge improvement on the generation of secure key.

Lastly, we reviewed the MDI-QKD protocol and concluded that, because it eliminates the measurement device side channels, it is very useful for practical QKD, since we no longer need to precisely define the setup components' characteristics. This makes the security anal-

ysis simpler and even more precise. It also necessarily employs decoy states, which improves the secret key rate. MDI-QKD therefore is a promising protocol to use for future quantum communications.
To improve this report, it would have been nice to have made a practical model that uses the MDI-QKD protocol. Then it would have been possible to generate a plot of the secret key rate and make quantitative comparisons with the BB84 protocol.

REFERENCES

Buchanan, W., & Woodward, A. (2017). Will quantum computers be the end of public key encryption? *Journal of Cyber Security Technology*, *1*(1), 1–22. https://doi.org/10.1080/23742917.2016.1226650

Mavroeidis, V., Vishi, K., Zych, M. D., & Jøsang, A. (2018). The Impact of Quantum Computing on Present Cryptography. *9*(3).

Gisin, N., Ribordy, G., Tittel, W., & Zbinden, H. (2002). Quantum cryptography. *Reviews of Modern Physics*, *74*(1), 145–195. https://doi.org/10.1103/RevModPhys.74.145

Pirandola, S., Andersen, U. L., Banchi, L., Berta, M., Bunandar, D., Colbeck, R., Englund, D., Gehring, T., Lupo, C., Ottaviani, C., Pereira, J., Razavi, M., Shaari, J. S., Tomamichel, M., Usenko, V. C., Vallone, G., Villoresi, P., & Wallden, P. (2019). Advances in quantum cryptography. *arXiv*, 1–118. https://doi.org/10.1364/aop.361502

Scarani, V., Bechmann-Pasquinucci, H., Cerf, N. J., Dušek, M., Lütkenhaus, N., & Peev, M. (2009). The security of practical quantum key distribution. *Reviews of Modern Physics*, *81*(3), 1301–1350. https://doi.org/10.1103/RevModPhys.81.1301

Xu, F., Ma, X., Zhang, Q., Lo, H. K., & Pan, J. W. (2020). Secure quantum key distribution with realistic devices. *Reviews of Modern Physics*, *92*(2), 1–68. https://doi.org/10.1103/REVMODPHYS.92.025002

Nielsen, M. A., & Chuang, I. L. (2002). *Quantum Computation and Quantum Information* (Vol. 70). Cambridge University Press. https://doi.org/10.1119/1.1463744

Stinespring, W. F. (1954). Positive functions on c*-algebras. *363*(10), 211–216.

Aardal, K., van Iersel, L., & Janssen, R. (2020). Optimization.

Shor, P. W., & Preskill, J. (2000). Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, *85*(2), 441–444. https://doi.org/10.1103/PhysRevLett.85.441

Wootters, W. K., & Zurek, W. H. (1982). A single quantum cannot be cloned. *Nature Sci. U.S.A*, *299*(9), 39–42.

Pfister, C., Lütkenhaus, N., Wehner, S., & Coles, P. J. (2016). Sifting attacks in finite-size quantum key distribution. *New Journal of Physics*, *18*(5), 1–36. https://doi.org/10.1088/1367-2630/18/5/053001

Hankerson, D. R., Hoffman, D. G., Leonard, D. A., Lindner, C. C., Phelps, K. T., Rodger, C. A., & Wall, J. (1991). *Coding theory and cryptography, the essentials*. Taylor & Francis Group.

Fetsje Bijma, A. v. d. V., Marianne Jonker. (2016). *An introduction to mathematical statistics*. Amsterdam University Press.

Marcos Curty, I. f. Q. C. (2013). *What theorists should know when working with - marcos curty*. https://www.youtube.com/watch?v=vFBSHBcLmGk&t=4066s

Xu, F., Curty, M., Qi, B., & Lo, H. K. (2015). Measurement-device-independent quantum cryptography. *IEEE Journal of Selected Topics in Quantum Electronics*, *21*(3), 1–11. https://doi.org/10.1109/JSTQE.2014.2381460

Lo, H. K., Curty, M., & Qi, B. (2012). Measurement-device-independent quantum key distribution. *Physical Review Letters, 108*(13), 1–5. https://doi.org/10.1103/PhysRevLett.108.130503

A

ENTANGLEMENT BASED BB84 PROTOCOL

BB84 protocol

Parameters: $m, k \in \mathbb{N}_+$; $p_x, p_z \in [0,1]$ with $p_x + p_z = 1$, $q_{\text{tol}} \in [0,1]$.

Output: for $n = m + k$ the outputs after the sifting are

Alice: n-bit string $(a_i)_{i=1}^n \in \{0,1\}^n$ (sifted outcomes),

Bob: n-bit string $(b_i)_{i=1}^n \in \{0,1\}^n$ (sifted outcomes),

public: the set K of size k, which contains the indices of the bits that were measured in the X-basis and the set M of size m which contains the indices of the bits that were measured in the Z-basis.

Final output: Either no output if the protocol aborts in Step 9 or:

Alice: m-bit string $(\alpha_i)_{i=1}^m \in \{0,1\}^m$ (raw key),

Bob: m-bit string $(\beta_i)_{i=1}^m \in \{0,1\}^m$ (raw key).

Number of rounds: Random variable N, determined by the termination condition (TC).

The protocol

Loop phase: The steps 1 to 5 are iterated round wise until the termination conditions is met. Set $j = 0$. Define the sets M' and K' which are initially empty. Starting with round $r = 1$ (where $r = 1, 2, ...$), Alice and Bob do:

Step 1: (Preparation): Alice prepares a qubit pair in a maximally entangled state.

Step 2: (Channel use): Alice uses the quantum channel to send half of the prepared qubit pair to Bob. The other half she keeps for herself.

Step 3: (Measurement): Bob measures the received half of the entangled qubit state in X-basis with probablity p_x or in the Z-basis with probability p_z. Alice measures her half in one of the two bases with the respective probabilities.

Step 4: (Classical bit storage): Alice stores the binary value she found with her measurement as s_r. Bob stores the binary value he obtains as t_r.

Step 5: (Public communication): Alice and Bob use the authentic classical channel to communicate their basis choice. If the bases were the same:

- $j = j + 1$,
- Alice sets $\hat{a}_j = s_r$,
- Bob sets $\hat{b}_j = t_r$,
- If both used the Z-basis, $M' = M' \cup \{j\}$,
- If both used the X-basis, $K' = K' \cup \{j\}$.

If the bases were different, Alice and Bob do nothing.

TC If the condition ($|M'| \geq m$ and $|K'| \geq k$) is reached (where $|M'|$ is the size of set M'), Alice and Bob set $N = r$ and $J = j$ and proceed with Step 6. Otherwise they set $r = r + 1$ and repeat from Step 1.

Final sifting phase: The following steps are performed only once. Alice now has a bit string $(\hat{a}_j)_{j=1}^{J} \in \{0,1\}^J$ and Bob has a bit string $(\hat{b}_j)_{j=1}^{J} \in \{0,1\}^J$ which resulted from the loop phase of the protocol.

Step 6: Alice and Bob will now choose at random a subset of size k from K', this is the set K. They also choose a random subset of size m from M', this is the set M.

Step 7: (Output): From her bit string $(\hat{a}_j)_{j=1}^{J} \in \{0,1\}^J$, Alice discards all the bits for which the index $j \notin M \cup K$. She then re-indexes her remaining bit string from 1 to n, while preserving the order. She obtains the sifted n-bit string $(a_i)_{i=1}^{n} \in \{0,1\}^n$. The new indexing is also translated to M and K so that the index numbers of the bits measured in the X-basis are in the set K and the index number of the bits measured in the Z-basis are in the set M and $M \cup K$ contains the numbers 1 to n. Bob carries out the same procedure for $(\hat{b}_j)_{j=1}^{J} \in \{0,1\}^J$ to obtain the sifted n-bit string $(b_i)_{i=1}^{n} \in \{0,1\}^n$.

Parameter estimation: In this part, Alice and Bob will test the errors that occurred and decide if the protocol needs to abort.

Step 8: Alice and Bob communicate their test bits over the public, authenticated channel. These are the bits a_i and b_i for which $i \in K$.

Step 9: (Correlation test): Alice and Bob determine the test bit error rate:

$$\lambda_X = \frac{1}{k} \sum_{i \in K} a_i \oplus b_i \tag{A.1}$$

where $\oplus$ is addition modulo 2. If $\lambda_X \leq q_{\text{tol}}$ they continue the protocol in Step 10. If $\lambda_x > q_{\text{tol}}$ they abort.

Step 10: (Raw key output) Alice outputs the m-bit string $(\alpha_i)_{i=1}^m \in \{0,1\}^m$ which consists of all a_i for which $i \in M$. Bob outputs the m-bit string $(\beta_i)_{i=1}^m \in \{0,1\}^m$ which consists of all b_i for which $i \in M$.

(from Pfister et al., 2016)