KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Technology

First cycle, 15 credits

Developing an educational tool for simulations of quantum key distribution systems

ERIK ÅKERBERG, ERIK ÅSGRIM

Stockholm, Sweden 2023

Abstract

Quantum key distribution (QKD) is the idea of using quantum systems to securely communicate a shared encryption key between two parties. In contrast to classical methods of encryption, QKD utilizes fundamental quantum properties such as superposition and entanglement to encode information in a way that guarantees security. Most QKD systems are based on sending photons in an optical fiber where the polarisation of the photons is the quantum property used to encode information. The different algorithms used to do this are referred to as QKD protocols. This thesis aimed to construct an educational tool to simulate simple QKD systems using four common QKD protocols, where the user can vary system parameters and study its effect on the results. Furthermore, the aim was to be able to produce simulation results that are accurate enough to provide a first approximation of how a real experimental setup would perform. The program was built in Python using the Qiskit library and all the desired features were implemented in a graphical interface. For one of the implemented protocols (BB84) the simulation results were compared to experimental data from three QKD experiments, which indicated that the program is able to produce a useful first approximation of a real experimental setup. The program could be further improved by allowing for simulations of more complex systems.

Acknowledgements

We would like to express our gratitude to our supervisor Mohammed Alqedra for his continuous help and support with this thesis. His ideas and advice have been very valuable in helping us progress throughout the project.

Contents

1 Introduction

1.1 Background

Today, quantum mechanics is utilized in many modern technologies and one of the most promising areas under development is quantum information. The idea behind quantum information is to use quantum mechanical properties of physical systems to encode, store and process information in ways useful for computing and communication [1]. Quantum mechanics allows for the measurement and manipulation of quantum systems that can occupy states that contain more information than classical systems and posses useful quantum properties, which present certain advantages for applications within information technology.

The fundamental concept that makes quantum systems useful is quantum superposition. This is the idea that properties of physical systems can be in combinations of multiple states at the same time [2]. Two common examples of physical systems exhibiting this behavior are electron spin and photon polarisation, both commonly used in the field of quantum information. The states of these systems can then be modified using different physical interactions in order to manipulate the informational content stored. They can also be measured in different ways in order to obtain the desired information. For example, some quantum computers use electrons with different spin states to store information and can perform calculations using these electrons by modifying their states and measuring them [3].

Another field within quantum information is quantum communication, which as the name suggests is the idea of using quantum information principles to communicate between two or more parties. This is often achieved by sending photons in superpositions of different polarisation states in optical fibers, however there are other types of implementations as well. Quantum communication theoretically offers the ability to transmit information completely securely and the most common technique of secure quantum communication is known as Quantum Key Distribution (QKD). QKD uses quantum mechanical properties to securely transmit an encryption key between two parties that can then be used to encrypt information that is communicated classically. To achieve this, different so called QKD protocols can be used that have different advantages and disadvantages depending on the specific setup.

These setups are often technically demanding to implement and rely on specialized components, tools and materials to operate [4]. Therefore, much of the research being done within quantum communication relies on being able to construct and fund these experimental systems to implement and test new ideas and solutions. It would therefore be useful to be able to simulate a proposed experimental setup with different QKD protocols in order to get an indication of what can be achieved with the available resources. This idea is the basis for this thesis.

1.2 Purpose

The purpose of the thesis is to construct a program for simple simulations of four essential quantum key distribution (QKD) protocols. The idea is to provide a graphical tool to simulate communications between two parties through an optical fiber. The program should work as an educational tool where the user is able to get a better understanding of how QKD protocols work and gain intuition about how varying different parameters affect the result. Furthermore, the aim is for the program to produce simulation results that can work as a first approximation of how an actual experimental setup would perform.

2 Theory

2.1 Quantum information

Whereas classical information is stored in bits, i.e. a variable that can only take the value 0 or 1, quantum information is based on storing information in quantum bits, also known as qubits [1]. Unlike a classical bit a qubit is not limited to only being in the state 0 or 1 but can also be in a superposition of the two different states, which allows for the encoding of more information compared to classical bits.

The possible superpositions a system can be in are constituted of linear combinations of base states and when the system is measured, it collapses to one of these base states with a probability determined by the state's corresponding magnitude [2]. However, these base states are not unique and any set of linearly independent states that span all possible states can be regarded as a set of base states, also called a state basis. Furthermore, even if a system is in a given base state, that base state can be described as a superposition of other base states in a different basis.

When a property of a physical system in a quantum superposition is measured, the measurement is done using a certain basis. For example, when measuring the polarisation of a photon it can be measured in the basis consisting of completely horizontal polarisation $|H\rangle$ and completely vertical polarisation $|V\rangle$, the so called rectilinear basis. This will result in a collapse to either of the base states used in the measurement with the probability corresponding to that base state in the superposition. However, if the photon is measured in a basis consisting of completely diagonal and anti-diagonal polarisation, the so called diagonal basis, it will collapse to either of these base states with different corresponding probabilities.

To describe the behavior of quantum systems mathematically, wave functions are used. A wave function is an abstract description of the system state and contains information about physical properties such as energy, position and momentum [2]. The wave function is commonly represented with the letter Ψ and it has the special property that the squared amplitude of the wave function $|\Psi|^2$ describes the probability density of obtaining a given measurement of a given system property. The vector space in which the wave function resides is called Hilbert space. Wave functions are commonly expressed in a basis, where the complete wave function of a system is a linear combination of the wave functions of the base states, which span Hilbert space.

2.2 The qubit

The state of a qubit is described as a wave function and can be written as

$$|\psi\rangle = A\,|0\rangle + B\,|1\rangle\,. \tag{1}$$

The states $|0\rangle$ and $|1\rangle$ are orthogonal and are known as the computational basis or the rectilinear basis. The coefficients A and B are complex numbers and the square of the magnitude of the coefficient can be interpreted as the probability of observing the qubit in that specific state. For example, the probability of observing the state $|0\rangle$ is given as $P(|0\rangle) = |A|^2 = AA^*$, if a measurement is performed in the computational basis.

The probabilistic interpretation of the superposition implies that the wave function must be normalized for the probabilities of the various measurement outcomes to add up to 1, i.e. the condition $\langle\psi|\psi\rangle = 1$ must be inferred.

Describing the state of a qubit might sometimes require using a different basis than the previously mentioned computational basis [1]. A common alternative basis is given by the base states

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \tag{2}$$

which are also orthogonal and normalized. This basis is often referred to as the diagonal basis.

A useful geometric representation of a qubit state is the so called Bloch sphere [5]. Because of the normalization constraint $\langle\psi|\psi\rangle = |A|^2 + |B|^2 = 1$ the wave function can be written in the alternative form

$$|\psi\rangle = cos(\theta/2)\,|0\rangle + sin(\theta/2)e^{i\phi}\,|1\rangle\,. \tag{3}$$

This way of representing a general qubit state allows for an intuitive geometric representation of the qubit as a point on a unit sphere given by the spherical coordinates (θ, ϕ), as seen in Figure 1 below.

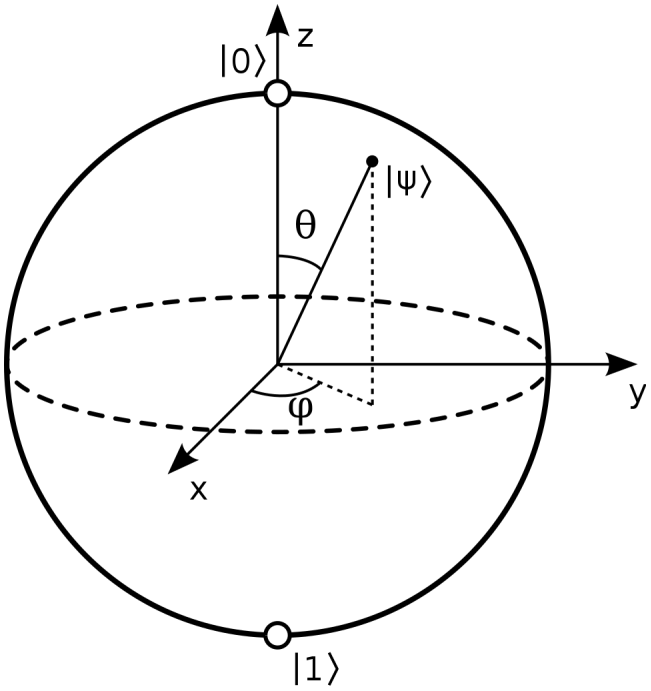

Figure 1: A general qubit state represented on the Bloch sphere [6].

On the Bloch sphere, orthogonal states are represented as antipodal points. Note that this means that the angles on the Bloch sphere are twice as large as the angles in Hilbert space.

2.3 Multi-qubit states and entanglement

Given two qubit states $|\psi_1\rangle = a_1\,|0\rangle + b_1\,|1\rangle$ and $|\psi_2\rangle = a_2\,|0\rangle + b_2\,|1\rangle$, their joint two-qubit wavefunction can be written as a tensor product of the single-qubit states [1]. Mathematically this is represented as

$$|\psi_{1,2}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = a_1a_2\,|00\rangle + a_1b_2\,|01\rangle + b_1a_2\,|10\rangle + b_1b_2\,|11\rangle\,. \tag{4}$$

A general wavefunction in a two-qubit state can thus be written as

$$|\psi\rangle = a_{00}\,|00\rangle + a_{01}\,|01\rangle + a_{10}\,|10\rangle + a_{11}\,|11\rangle \tag{5}$$

where the normalization condition is given as $|a_{00}|^2 + |a_{01}|^2 + |a_{10}|^2 + |a_{11}|^2 = 1$ [7].

The above principles generalize well to an arbitrary amount of qubits. Given n different single qubit states $|\psi_1\rangle, ..., |\psi_n\rangle$ their joint https://www.overleaf.com/project/63d911b52db67b48de8e5384wave function is given as $|\psi_{1,...,n}\rangle = \bigotimes_{i=1}^{n} |\psi_i\rangle$. Denoting the set of all possible n-bit strings as $\{0,1\}^n$, a general n-qubit state can be written as

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} a_x |x\rangle \tag{6}$$

where the normalization constraint simply reduces to $\sum |a_x|^2 = 1$.

Particularly relevant to quantum information are so called entangled states. In a multi-qubit state, entangled states are states that can not be represented as a tensor product of single-qubit states [8]. Mathematically this can be expressed as

$$|\psi\rangle \neq |\psi_1\rangle \otimes ... \otimes |\psi_n\rangle \,. \tag{7}$$

An often useful set of entangled states for a two-qubit system are the so called Bell states, defined as:

$$\begin{aligned} |\Phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\Phi^-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\Psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\Psi^-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \end{aligned} \tag{8}$$

Note that the Bell states are all normalized states and constitute an orthogonal basis for the Hilbert space of two-qubit systems. In order to test the entanglement of two qubits in a Bell state, a so called Bell test can be used. A Bell test is an experiment designed to test what is known as Bell's inequality. Bell's inequality is a mathematical statement that puts constraints on the correlation between the measurements of two entangled particles based on the assumption of local realism [9]. Local realism is the idea that physical systems have definite properties even before they are measured and that those properties are not affected by measurements made on distant systems. Quantum mechanics predicts that entangled particles can exhibit correlations that violate Bell's inequality and the Bell test tests this prediction. Essentially, a Bell test is an experiment that is intended to prove that quantum mechanics violates the principle of local realism and is not predetermined by hidden variables.

A Bell test involves preparing pairs of entangled particles, separating them, measuring them and seeing if the results violate or adhere to Bell's inequality [10]. There are many types of systems and inequalities that can be used in a Bell test and the most common Bell test that has been experimentally implemented uses photon polarisation and is called the CHSH inequality. This Bell test verifies that pairs of photons are in fact entangled and gives a metric of the fidelity of the entanglement called the S statistic. The experimental procedure is described below but the mathematical reasoning behind this experiment is left out here. Consider the experimental setup shown in Figure 2 below.

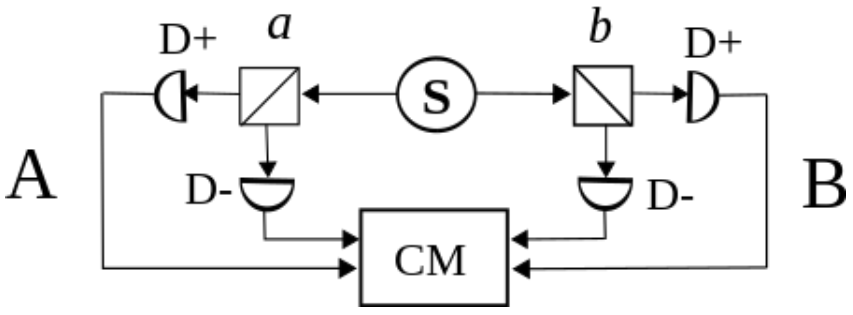

Figure 2: CHSH test experimental setup [11]

The source (S) produces an entangled pair of photons (A and B) which are sent to two two-channel polarisers (a and b). The polarisers can be set to a desired polarisation direction and the two channels of each polariser filter for positive polarisation and negative polarisation respectively in the chosen direction. When a photon passes through one of these channels, the detectors (D+ and D-) detect which channel it passed through indicating whether the photon had positive (+) or negative (-) polarisation in the chosen direction. There are therefore four possible outcomes for each entangled pair of photons A and B: both positive (++), A positive and B negative (+-), A negative and B positive (-+) and both negative (- -). The coincidence monitor (CM) counts the number of times each outcome occurs.

The direction of polariser a can be set to either $a_1 = 0°$ or $a_2 = 45°$ and the direction of polariser b can be set to $b_1 = 22.5°$ or $b_2 = 67.5°$. Four experiments can now be done, one for each choice of directions for a and b. For each experiment, the number of times each outcome occurred is counted and stored as N_{++}, N_{+-}, N_{-+} and N_{--}. An expectation value E from each experiment can now be defined as

$$E(a_i, b_i) = \frac{N_{++} - N_{+-} - N_{-+} + N_{--}}{N_{++} + N_{+-} + N_{-+} + N_{--}}. \tag{9}$$

The CHSH inequality says that if local realism is true and that the polarisations of the photons are predetermined and not actually entangled, then the so called S statistic should satisfy

$$|S| = E(a_1, b_1) - E(a_1, b_2) + E(a_2, b_1) + E(a_2, b_2) \leq 2. \tag{10}$$

In real experiments this inequality is violated which implies that local realism is not true and quantum mechanics is not actually determined by hidden variables. However, if the photons have been measured beforehand the S statistic will satisfy the inequality, since the polarisations of the photons now are predetermined. This can for example be used to test for eavesdropping in quantum communication applications.

2.4 Quantum gates

Ordinary classical bits are manipulated using so called logical gates, which are small physical devices that perform some simple operation on one or a few bits [12]. A simple example of such a logical gate is the so called NOT-gate which simply flips a bit from 0 to 1 and vice versa. In order to manipulate qubits quantum logical gates are used. Just like their classical counterpart, quantum logical gates are physical devices that manipulate one or a few qubits in some way. Unlike classical logical gates, quantum logical gates must be reversible. Each quantum gate can be represented as a matrix, and the resulting state after letting a quantum logical gate act on a qubit is acquired by calculating the matrix-vector product of the matrix corresponding to the gate and the vector corresponding to the state. Some of the most common gates are listed below.

- $\mathrm{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ (Pauli-X gate)
- $\mathrm{H} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$ (Hadamard gate)
- $\mathrm{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ (Controlled NOT gate)
- $\mathrm{RY}(\theta) = \begin{bmatrix} \cos\theta/2 & -\sin\theta/2 \\ \sin\theta/2 & \cos\theta/2 \end{bmatrix}$ (Rotate-Y gate)

The Pauli-X gate, sometimes referred to as simply the X-gate or the NOT-gate, flips the qubit state $|0\rangle$ to $|1\rangle$ and vice versa. On the Bloch sphere the effect of the Pauli-X gate can thus be thought of as rotating the qubit state 180° around the x-axis. The Hadamard gate essentially performs a change of basis between the rectilinear basis and the diagonal basis, meaning that $H|0\rangle = |+\rangle$, $H|1\rangle = |-\rangle$, $H|+\rangle = |0\rangle$ and

$H|-\rangle = |1\rangle$. The CNOT-gate acts on two-qubit systems $|\psi\rangle = |\psi_1\psi_2\rangle$ and has the effect of performing a Pauli-X gate on $|\psi_1\rangle$ if $|\psi_2\rangle = |1\rangle$ and doing nothing if $|\psi_2\rangle = |0\rangle$. When using the CNOT-gate, the $|\psi_1\rangle$ qubit is referred to as the work qubit and $|\psi_2\rangle$ is referred to as the control qubit. Finally, the RY-gate rotates a given qubit state θ degrees around the y-axis on the Bloch sphere. Note that quantum gates represented by 2x2 matrices act on single qubit states, while the 4x4 matrix representing the CNOT-gate acts on two-qubit states.

A quantum algorithm is often summarized in a quantum circuit. A quantum circuit is a computational routine consisting of quantum gates and measurements being performed on a number of qubits in a sequential manner [13]. An example of a simple quantum circuit can be seen in Figure 3 below.

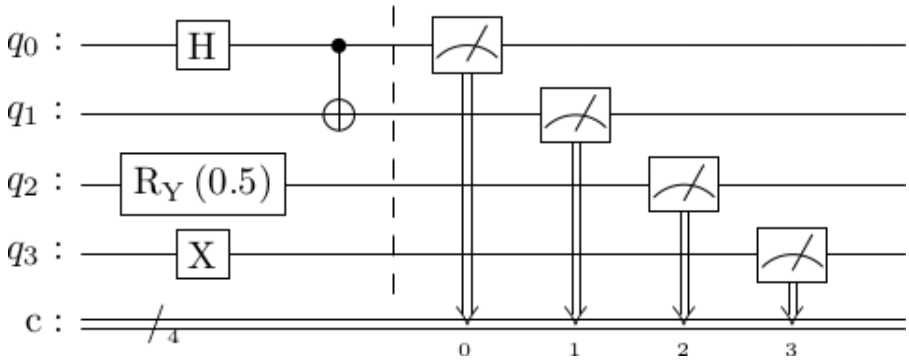

Figure 3: An example quantum circuit with 4 qubits. The different square blocks indicate various gates acting on the qubits. The rightmost blocks indicate measurements. The bottom channel indicated with a 'c' is a representation of classically stored information where 0, 1, 2 and 3 are the indices of the classical bits.

In the example circuit above four qubits are used as input. Initially, all qubit states are set to $|0\rangle$. On the q_0 qubit a Hadamard gate is applied and the result is then used as the control qubit in a CNOT gate with the q_1 qubit. On the q_2 qubit, an RY-gate is applied in order to rotate the the initial $|0\rangle$ state 0.5 radians on the Bloch sphere around the y-axis. On the q_3 qubit a Pauli-X gate is applied, thus flipping the state from $|0\rangle$ to $|1\rangle$. After the gates have been applied to the qubits, all qubit states are measured. A measurement is indicated by an icon with a gauge, as seen in the rightmost part of the circuit. The information is then stored in classical bits, which are represented with the letter 'c' on the bottom part of the circuit. Note that the gates in a quantum circuit are executed sequentially from left to right.

2.5 Quantum key distribution

2.5.1 Main idea

The goal of QKD is to develop methods for secure distribution of encryption keys between two parties (referred to as Alice and Bob) [8]. This is done by sharing a random classical bit string between the two parties which is encoded using quantum states. The various algorithms used to achieve this are referred to as QKD protocols. QKD protocols produce keys useful in symmetrical cryptosystems where the same key is used for both encryption and decryption of data.

2.5.2 Advantages over classical methods

The main advantage of quantum key distribution compared to classical methods is the security of the protocols. Currently, most data is encrypted using a method known as public-key cryptography [4]. Public-key cryptography, also known as asymmetric cryptography, is a system that uses two mathematically related keys called the public key and the private key. The public key can be distributed publicly, while the private key must be kept secret by the owner to ensure security. A public key is used to encrypt data by the sender, while the private key is used to decrypt the data by the receiver. Since the public and private key are mathematically related, the safety of public-key cryptography relies on the fact that it

must be computationally difficult and time consuming to find the private key given the public key. In RSA encryption for example, the mathematical problem of factoring large numbers into its prime factors is used for relating the public and private key. For classical computers, solving this task is a process that would take thousands of years of computation for sufficiently large numbers. However, in 1994, the physicist Peter Shor came up with an algorithm, nowadays referred to as Shor's algorithm, which would enable a quantum computer to factor large numbers much faster than classical computers due to achieving a lower computational complexity [14].

Currently, quantum computers are not powerful enough to be able to decrypt RSA encryption, however many believe that this might change in the future and that new encryption methods are required. QKD poses as a possible candidate for such a method. In contrast to public-key cryptography, where the security of the protocols rely on computational difficulty, the security of QKD relies on fundamental quantum mechanical effects such as superposition and entanglement [15]. This means that the security of QKD protocols wont be compromised by the improved computational power of quantum computers. There are however still many challenges when it comes to using QKD at large scale. Examples of such challenges are the fact that QKD requires specialized equipment and that high transmission losses in optical fibers are currently making long-distance QKD difficult [4].

2.5.3 Single qubit versus entanglement based QKD

There are many different types of QKD protocols which all work differently with different strengths and weaknesses. An important distinction to make between different QKD protocols is whether they are single-qubit or entanglement based protocols.

Single-qubit protocols, like the name suggests, are based on sending single qubits in various superpositions from one party to another. The security of the communication relies on the uncertainty of an outcome when performing measurements on qubits in superposition.

Entanglement based protocols are based on sending pairs of qubits that are in an entangled state. Unlike the single-qubit protocols, the qubits are generally not sent from one party to another when using entanglement based protocols. Instead, a separate qubit source is used, which prepares qubits in a known entangled state, after which the qubits are distributed to the two parties.

2.5.4 Eavesdropping

In cryptography, the process of a third party trying to access the information being transmitted between two legitimate users is called eavesdropping. In the context of QKD, a potential eavesdropper is often referred to as Eve. Even though eavesdropping can not be prevented by using QKD, eavesdropping can always be detected by observing an abnormal error rate in the acquired encryption keys [16]. This is a significant difference compared to classical encryption methods, where two parties can never guarantee that a third party has not acquired a copy of the encryption key. The reason an eavesdropper can not intercept quantum information without it being detected is because of the so called no-cloning theorem. The no-cloning theorem states that it is only possible to create a device that can clone mutually orthogonal quantum states, and that cloning arbitrary quantum states is impossible [17]. Therefore, encoding information in non-orthogonal quantum states makes it physically impossible for an eavesdropper to intercept communication without introducing errors into the encryption key, which can later be detected by the legitimate users.

Even though it is impossible for an eavesdropper to intercept a QKD transmission without introducing errors into the encryption key, there are still different eavesdropping strategies that might reduce the error introduced into the key. More sophisticated eavesdropping strategies are not covered in this thesis. For a description of the eavesdropping strategies implemented in the program, see section 3.3 covering the implementation of the protocols.

2.6 QKD protocols

There are many different protocols used for QKD and four of particular interest are described here. These are BB84, B92, E91 and BBM92. These protocols can be used in different types of systems and are described abstractly below but the most common use case is a fiber-optic communication between two

parties where photons are transmitted in different polarisation states. Therefore, a qubit corresponds to a photon and the state of the qubit corresponds to the polarisation state of the photon. The two parties can send these photons in a quantum channel but can also communicate classically in a classical channel. In reality, these two channels can be in the same or two separate fibers.

2.6.1 BB84

The BB84 protocol is a single qubit protocol that was developed in 1984 by Charles Bennett and Gilles Brassard [18]. In the BB84 protocol the sender (Alice) first has n bits that she wants to send to the receiver (Bob). For each bit, Alice chooses randomly whether to encode the bit using the rectilinear basis or the diagonal basis. In the rectilinear basis, 1 is encoded as $|1\rangle$ and 0 is encoded as $|0\rangle$. In the diagonal basis, 1 is encoded as $|-\rangle$ and 0 is encoded as $|+\rangle$. Using a quantum channel, Alice then sends the encoded qubits to Bob. Bob, who does not know what bases Alice has chosen, chooses randomly between measuring using the rectilinear basis or the diagonal basis for each qubit he receives from Alice. Alice and Bob then communicate over a classical channel and share with each other what basis they chose for each qubit for encoding and measuring. They then discard the bits where they chose different bases and only keep the bits where they choose the same basis. Given that there is no eavesdropping, Alice and Bob can now be sure that they share the same key which can later be used for encryption. An example of 10 qubits being transmitted using the BB84 protocol can be seen in Table 1.

Alice's random bit string	0	1	0	0	0	0	1	1	0	1
Choice of encoding basis	+	×	×	+	+	×	+	×	+	×
Sent state by Alice	$\|0\rangle$	$\|-\rangle$	$\|+\rangle$	$\|0\rangle$	$\|0\rangle$	$\|+\rangle$	$\|1\rangle$	$\|-\rangle$	$\|0\rangle$	$\|-\rangle$
Bob's choice of measurement basis	×	×	+	×	+	+	×	×	+	+
Bob's measurements	$\|+\rangle$	$\|-\rangle$	$\|1\rangle$	$\|+\rangle$	$\|0\rangle$	$\|1\rangle$	$\|+\rangle$	$\|-\rangle$	$\|0\rangle$	$\|0\rangle$
Alice and Bob share choice of basis										
Shared secret key		1			0			1	0	

Table 1: Example of BB84 protocol step by step where 10 qubits are being sent from Alice to Bob resulting in a final secret key consisting of 4 bits.

If eavesdropping (or other types of random measurements) occur, there will be errors in the resulting shared key, i.e. the keys will not match entirely. To estimate the error in the shared key Alice and Bob typically share a fraction of their keys over the classical channel in order to approximate the fraction of bits in their shared keys that do not match. This is known as the qubit error rate (QBER). The shared part of the key is then discarded from the final key.

2.6.2 B92

The B92 protocol is a modified version of the BB84 protocol which was suggested by Charles Bennett in 1992 [19]. Just like in the BB84 protocol, Alice has n bits that she wants to encode and send to Bob. To encode the bits Alice chooses a non-orthogonal basis, usually consisting of the vertical $|0\rangle$ state and the diagonal $|+\rangle$ state. For example, she might encode every 0 as $|0\rangle$ and every 1 as $|+\rangle$. Alice then sends her encoded qubits to Bob. Just like in the BB84 protocol, Bob will choose randomly between measuring in the rectilinear basis or the diagonal basis for each qubit he receives from Alice.

Thus, if Alice sends a qubit in the $|0\rangle$ state and Bob chooses the rectilinear basis, he will measure $|0\rangle$ with 100% certainty but can not know if what Alice sent was actually $|0\rangle$ or $|+\rangle$, since the measurement of $|0\rangle$ is possible in both scenarios. However, if he chooses the diagonal basis he will measure either $|+\rangle$ or $|-\rangle$, both with 50% probability. If Bob measures $|-\rangle$, he knows that Alice must have sent the $|0\rangle$ state corresponding to a 0 since Alice only sends either $|+\rangle$ or $|0\rangle$ and it is impossible to measure $|-\rangle$ if she sent $|+\rangle$. Similarly, if Alice sends a qubit in the $|+\rangle$ state and Bob makes a measurement of $|1\rangle$ using the rectilinear basis he knows that Alice must have sent him the $|+\rangle$ state corresponding to a 1. When Bob has finished all of his measurements he then tells Alice over the classical channel for which qubits he measured either $|-\rangle$ or $|1\rangle$. For these measurements Bob knows for sure what state Alice sent and therefore what bit she sent, and these bits are therefore used as the shared key while the remaining qubits are discarded. An example of 10 qubits being transmitted using the B92 protocol can be seen in Table 2.

Alice's random bit string	0	1	1	0	0	1	0	1	0	1
Choice of encoding basis	+	×	×	+	+	×	+	×	+	×
Sent state by Alice	$\|0\rangle$	$\|+\rangle$	$\|+\rangle$	$\|0\rangle$	$\|0\rangle$	$\|+\rangle$	$\|0\rangle$	$\|+\rangle$	$\|0\rangle$	$\|+\rangle$
Bob's choice of measurement basis	×	×	+	×	+	+	×	×	+	+
Bob's measurement	$\|-\rangle$	$\|+\rangle$	$\|0\rangle$	$\|-\rangle$	$\|0\rangle$	$\|1\rangle$	$\|+\rangle$	$\|+\rangle$	$\|0\rangle$	$\|1\rangle$
Alice and Bob share choice of basis										
Shared secret key	1			1		0				0

Table 2: Example of B92 protocol step by step where 10 qubits are being sent from Alice to Bob resulting in a final secret key consisting of 4 bits.

Just like when using the BB84 protocol, a fraction of the acquired key can be shared between Alice and Bob in order to approximate the QBER to detect eavesdropping and measure system errors.

On average, B92 results in a shorter final encryption key than BB84 given a certain amount of sent qubits. This is because of the inherent randomness of the protocol. First, Bob has to choose the right basis to measure in and then he must also make the correct measurement for the result to be useful in producing the shared key. When using the BB84 protocol, Bob only has to choose the right basis which means less qubits have to be discarded. The upside of the B92 protocol is that Alice only has to be able to prepare qubits in two different states and that only one-way communication is required over the classical channel to produce the encryption key.

2.6.3 E91

The E91 protocol is an entanglement based protocol proposed by Artur Eckert in 1991 with the idea that a Bell test could be used in order to verify security [20]. With E91, pairs of entangled qubits in Bell states $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ are created by a quantum source and one of the entangled qubits in each pair is sent to Alice and the other one is sent to Bob. It does not matter by who or where the entangled pairs are created, they could be created by Alice, Bob or by a third party. After receiving their qubits, Alice and Bob both measure them. Alice chooses a random angle of polarisation to perform the measurements in from $\{0, \pi/8, \pi/4\}$ and Bob chooses a random angle of polarisation to perform his measurements in from $\{\pi/8, \pi/4, 3\pi/8\}$. After all qubits have been measured, Alice and Bob communicate over the classical channel and announce what angles they chose to do each of their measurements in. They use the measurements where they chose the same angle to construct the encryption key. The other measurements where Alice and Bob chose different angles are used to perform a CHSH test in order to get a metric of the entanglement fidelity. If the fidelity is poor, Alice and Bob can conclude that there might be an eavesdropper or other types of unwanted measurements that are occurring on the quantum channel meaning that the channel might not be safe. An example of the E91 protocol, when 10 entangled pairs of qubits are being distributed to Alice and Bob can be seen in Table 3.

Alice's measurement angle	0	$\frac{\pi}{4}$	$\frac{\pi}{8}$	0	$\frac{\pi}{4}$	0	$\frac{\pi}{8}$	$\frac{\pi}{4}$	0	$\frac{\pi}{8}$
Alice's measured bit	1	0	0	1	0	1	1	1	0	1
Bob's measurement angle	$\frac{\pi}{8}$	$\frac{\pi}{4}$	$\frac{\pi}{8}$	$\frac{\pi}{4}$	$\frac{3\pi}{8}$	$\frac{\pi}{8}$	$\frac{\pi}{4}$	$\frac{\pi}{8}$	$\frac{3\pi}{8}$	$\frac{\pi}{8}$
Bob's measured bit	0	0	0	1	1	0	1	0	0	1
Alice and Bob share choice of basis										
Shared secret key		0	0							1

Table 3: Example of E91 protocol where 10 pairs of entangled qubits are shared between Alice and Bob, resulting in a final secret key consisting of 4 bits.

Just like with the previously mentioned QKD protocols, a fraction of the encryption key can be shared and compared between Alice and Bob in order to get an estimate of the QBER. Even if the results from the Bell test state that there is no eavesdropper, the key might still contain errors due to other sources of error in the system. Therefore, estimating the QBER can be useful in addition to doing a Bell test [21].

2.6.4 BBM92

The BBM92 protocol was proposed as a possible simplification of E91 where a Bell test is not required in order to guarantee security [22]. Contrary to E91, this protocol was also proven to not be vulnerable to an eavesdropper changing the quantum source to a source from which information can be gained of the communication between Alice and Bob.

BBM92 works in a similiar manner to E91 in that a source produces an entangled Bell state $|\Phi^+\rangle$ where one qubit is sent to Alice and one is sent to Bob. After receiving their qubits, Alice and Bob both measure their qubits by randomly selecting the rectilinear or diagonal basis. They then announce to each other their choice of basis for each qubit and discard all measurement results for which they chose different bases. They then share at least half of all measurements with each other and compare their results. Even without eavesdropping, no system will retain perfect entanglement. Therefore, they calculate the QBER and if this error rate is sufficiently small, the remaining measurements not shared with each other are used as the secret key. An example of the BBM92 protocol, when 10 entangled pairs of qubits are being distributed to Alice and Bob can be seen in Table 4.

Alice's choice of measurement basis	+	×	×	+	+	×	+	×	+	×
Alice's measurements	$\|0\rangle$	$\|+\rangle$	$\|+\rangle$	$\|1\rangle$	$\|0\rangle$	$\|+\rangle$	$\|1\rangle$	$\|-\rangle$	$\|1\rangle$	×
Bob's choice of measurement basis	×	×	+	×	+	+	×	×	+	+
Bob's measurements	$\|+\rangle$	$\|+\rangle$	$\|1\rangle$	$\|-\rangle$	$\|0\rangle$	$\|1\rangle$	$\|+\rangle$	$\|-\rangle$	$\|1\rangle$	$\|0\rangle$
Alice and Bob share choice of basis										
Shared secret key		0			0			1	1	

Table 4: Example of BBM91 protocol where 10 pairs of entangled qubits are shared between Alice and Bob, resulting in a final secret key consisting of 4 bits.

This protocol is essentially an entanglement based version of BB84. It offers the advantages of using entangled pairs for the key distribution, while being simpler to implement than E91. However, BBM92 requires a large fraction of the measurements to be used for security verification which is a disadvantage of the protocol.

3 Method

3.1 Program overview

The desired layout of the program consists of two tabs. One for running a single simulation and one for running multiple simulations. When running a single simulation, the desired workflow is as follows.

First, the user should be able to enter what type of system is being used for the communication by providing relevant parameters that describe the desired setup. Second, the user should be able to choose one of the four QKD protocols and enter parameters specific to the simulation being run relevant for the chosen protocol. Third, the user should be shown the relevant results of the simulation and be able to display a graphical representation of the communication. When running multiple simulations the user should be able to automatically vary a chosen parameter and produce results in the form of a plot.

Each protocol was first implemented ideally as described in section 3.3 with the results calculated as described in section 3.5. Thereafter system behaviors were implemented that simulate real world effects such as losses and disturbances in the communication as described in section 3.4. The ability to run multiple simulations was then implemented as described in section 3.6 and finally the option to view a part of the communication graphically was added as described in section 3.7.

3.2 Qiskit

In order to implement the QKD protocols the Qiskit library was used. Qiskit is an open source library developed for Python used to create and simulate quantum algorithms [23]. A quantum algorithm is run in Qiskit by creating a quantum circuit corresponding to the algorithm and then choosing whether to run the algorithm remotely on a real quantum device or to run a classical simulation of the algorithm using the Qiskit Aer simulator. In the program, the latter option was used. For each simulation of a QKD protocol a quantum circuit is created corresponding to the chosen protocol. Since each protocol was constructed as a quantum circuit the different parts of the QKD protocols were thus represented as quantum gates which could be sequentially added to the quantum circuit representing the protocol. This program structure allowed for a flexible way of constructing the QKD protocols since each part of the protocol, such as encoding or decoding, simply required adding a block of quantum gates to the circuit.

3.3 Implementation of protocols

3.3.1 BB84

In the BB84 protocol, Alice generates a random bit sequence which she sends to Bob, where each bit is randomly encoded in either the rectiliear or diagonal basis. In the implementation this encoding of the qubits corresponds to first randomly adding an X-gate with 50 % probability followed by adding an H-gate with 50 % probability. The effect of these gates on the $|0\rangle$ state, which is always the initial state in any quantum circuit, is summarized below in equation 11.

$$\begin{aligned} X(|0\rangle) &= |1\rangle \\ H(|0\rangle) &= |+\rangle \\ H \circ X(|0\rangle) &= |-\rangle \end{aligned} \tag{11}$$

Thus, the result of adding these gates in the random fashion described above will result in Alice sending $|0\rangle, |1\rangle, |+\rangle$ or $|-\rangle$ all with equal probability. The random options chosen are stored as Alice's version of the shared final key where $|1\rangle$ and $|-\rangle$ represent a 1 and $|0\rangle$ and $|+\rangle$ represent a 0.

When measuring, Bob will choose randomly between doing a measurement in the rectilinear or the diagonal basis. This was implemented by first adding an H-gate with 50 % probability to each qubit followed by a measurement in the rectilinear basis. This sequence works since an H-gate followed by a measurement in the rectilinear basis will yield the same results as doing a measurement in the diagonal basis. The results of these measurements will produce Bob's version of the shared final key.

If eavesdropping occurs an assumption that was made is that Eve would attempt to eavesdrop every qubit transmitted from Alice. It was also assumed that Eve would use the strategy of simply forwarding the qubit state to Bob that she measured in the basis that she chose. Eve will thus also randomly choose between doing a measurement in the rectilinear or diagonal basis. This was implemented as follows. With a 50 % probability nothing was done to an intercepted qubit, since Eve would choose the correct basis with 50 % probability. If Eve chose the wrong basis she would forward a random base state from the wrong basis. This was implemented by adding an H-gate with 25 % probability and adding a sequence of X-H-X-gates with 25 % probability. Note that the effect of this sequence on the base states is given as follows:

$$\begin{aligned} X \circ H \circ X(|0\rangle) &= -|-\rangle \\ X \circ H \circ X(|1\rangle) &= |+\rangle \\ X \circ H \circ X(|+\rangle) &= |1\rangle \\ X \circ H \circ X(|-\rangle) &= |0\rangle\,. \end{aligned} \tag{12}$$

Adding gates in this way will thus simulate the effect of having an eavesdropper in the BB84 protocol. Note that the minus sign that appears in front of the $|-\rangle$ state in equation 12 does not affect the results since the $|\psi\rangle = |-\rangle$ and the $|\psi\rangle = -|-\rangle$ are physically equivalent since one state can be aquired from the other by doing a global phase shift. Figure 4 below shows a flowchart of the described implementation.

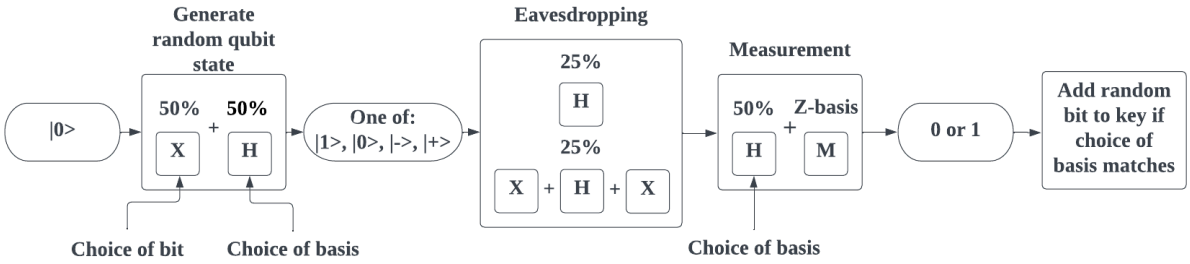

Figure 4: Flowchart of BB84 protocol implementation

3.3.2 B92

In the B92 protocol Alice generates a random bit string and encodes every 0 as $|0\rangle$ and every 1 as $|+\rangle$. Implementing this encoding in the quantum circuit using gates simply means that an H-gate is added with 50 % probability and that nothing is done to the qubit with 50 % probability. Similarly to BB84, the random options chosen are stored as Alice's version of the shared final key.

When performing the measurements, Bob chooses randomly between doing a measurement in the rectilinear or the diagonal basis. This means that the implementation of the measurements are the same as when using the BB84 protocol, i.e. first an H-gate is added with 50 % probability followed by a measurement in the rectilinear basis.

Just like in the BB84 protocol, an assumption is made that if eavesdropping occurs Eve would attempt to eavesdrop every qubit sent from Alice to Bob and she would forward the qubit she measured in the basis that she chose. Simulating this behaviour using quantum gates means doing the exact same thing as was done when simulating eavesdropping for the BB84 protocol. With 50 % probability nothing is done to the qubit, simulating that Eve chose the same basis that Alice chose for encoding. With 25 % probability an H-gate is added and with 25 % probability a sequence of X-H-X gates are added to simulate Eve choosing the wrong basis, and thus forwarding a random base state in the incorrect basis. Figure 5 below shows a flowchart of the described implementation.

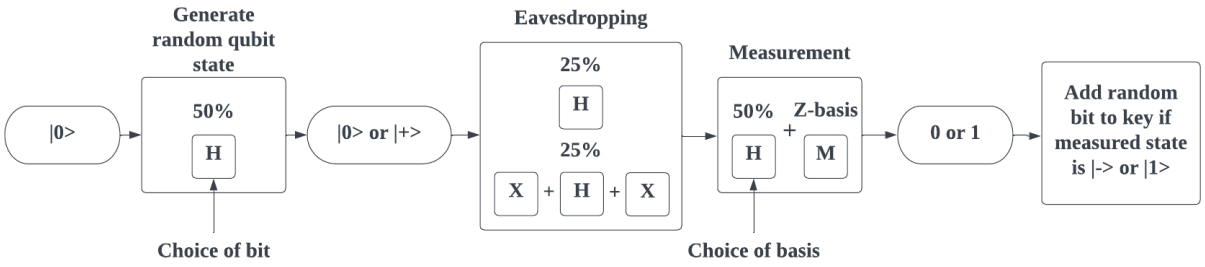

Figure 5: Flowchart of B92 protocol implementation

3.3.3 E91

In order to implement the E91 protocol, initial Bell state pairs $|\Phi^+\rangle$ had to be created. This is achieved using an H-gate followed by a CNOT-gate on a $|00\rangle$ initial double qubit state. In a quantum circuit the encoding of the Bell state thus simply results in the circuit seen below in Figure 6.

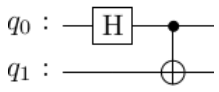

Figure 6: The quantum circuit used to generate a bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}\ (|00\rangle + |11\rangle$.

The encoding part of the E91 protocol therefore means the circuit in Figure 6 is added to every pair of qubits.

During the measurement phase of the protocol Alice chooses a random angle of polarisation to perform the measurement in from $\{0, \pi/8, \pi/4\}$. This can be implemented by adding an RY-gate where the angle θ to rotate is randomly chosen from $\{0, -\pi/4, -\pi/2\}$ followed by a measurement in the rectilinear basis. Note that the angles have to be doubled in the RY-gate, since the angle θ that the RY-gate rotates the polarisation is θ degrees on the Bloch sphere where the angles are twice as large as in Hilbert space. The measurements that Bob perform are implemented in the exact same way, except for the fact that Bob chooses a random angle of polarisation to perform the measurement in from $\{\pi/8, \pi/4, 3\pi/8\}$. To implement this, an RY-gate is added where the angle θ is randomly chosen from $\{-\pi/4, -\pi/2\ -\ 3\pi/4\}$ followed by a measurement in the rectilinear basis. Alice's and Bob's randomly chosen angles are then stored and the measurements for which the choice of angle was the same are used as the shared secret key.

When simulating eavesdropping, the assumption was made that Eve has the possibility to access both qubits being sent from the source. Assuming that the strategy Eve uses is to simply replace the entangled qubits from the source with her own prepared data, this can be simulated by first resetting both qubits to the $|0\rangle$ state and then encoding a random identical binary sequence into the qubits which she then transmits to Alice and Bob. In the quantum circuits this is done by first resetting the qubits to the $|0\rangle$ state followed by randomly adding an X-gate to both qubits with 50 % probability. Figure 7 below shows a flowchart of the described implementation.

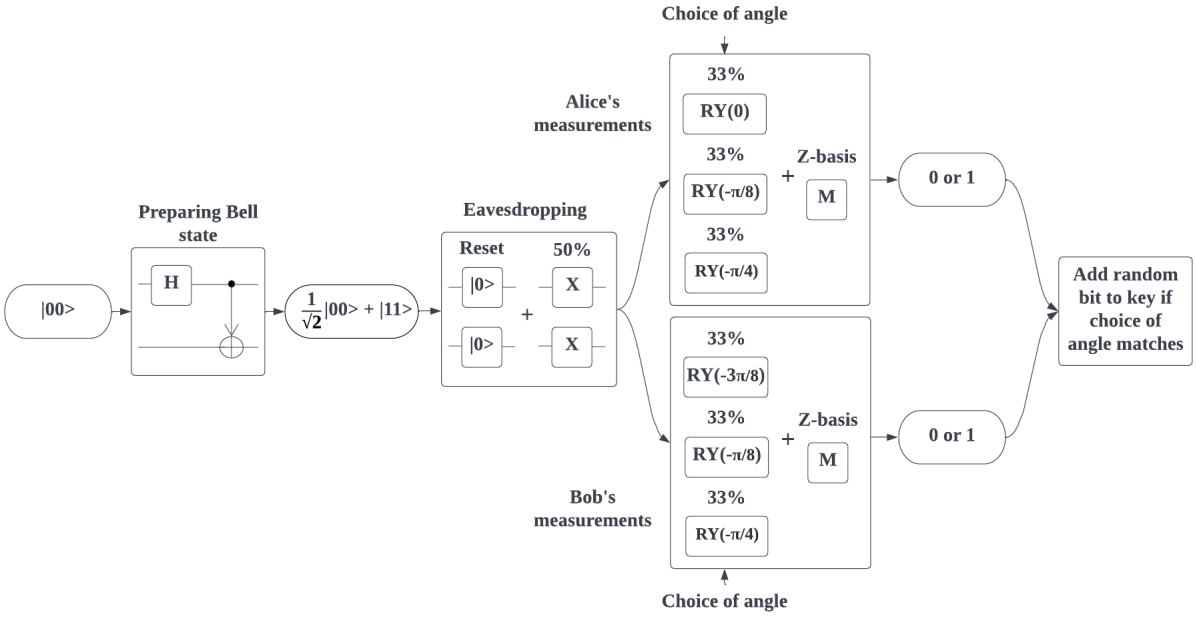

Figure 7: Flowchart of E91 protocol implementation

3.3.4 BBM92

To implement the BBM92 protocol the encoding is exactly the same as described for E91 in the previous section where Bell states are prepared for all pairs of qubits.

When performing the measurements Alice and Bob both choose randomly between measuring in the rectilinear or diagonal basis. This is, like seen previously, simulated by first adding an H-gate with 50 % probability followed by doing a measurement in the rectilinear basis. The random choices are stored and the measurements for which they chose the same basis are used as the shared final key.

Since the encoding is the same in the BBM92 protocol as in the E91 protocol, it was assumed that Eve uses the exact same strategy to eavesdrop as for the E91 protocol. Eve will replace the entangled qubits from the source with her own prepared binary sequence encoded into the qubits that she then forwards to Alice and Bob. The implementation of this is identical to the eavesdropping implementation of the E91 protocol, i.e. first the qubits are reset to the $|0\rangle$ state and then an X-gate is randomly added to both qubits with 50 % probability. Figure 8 below shows a flowchart of the described implementation.

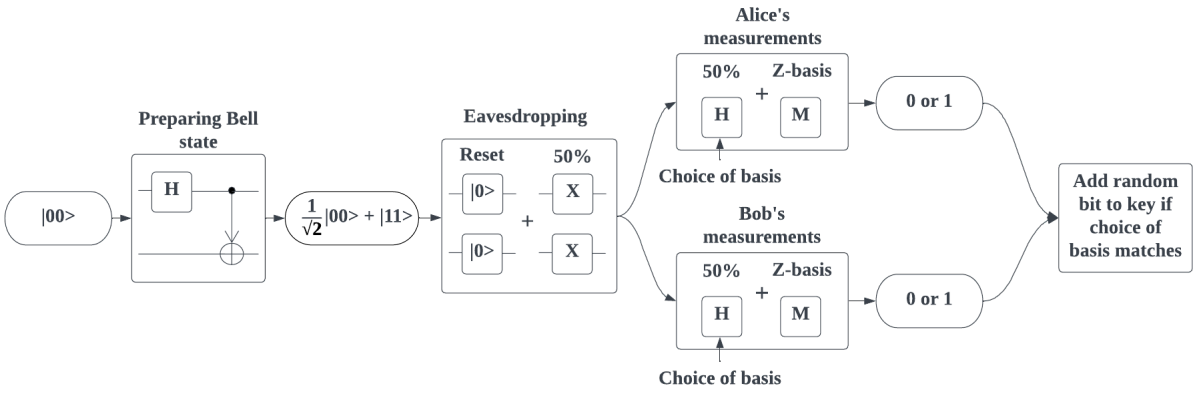

Figure 8: Flowchart of BBM92 protocol implementation

3.4 Program parameters

The implementations of the protocols described in the previous section are the ideal cases without losses or disturbances. In order for the simulations to be more realistic, relevant user-editable parameters need to be implemented that can be used to adapt the simulations to a specific real world use-case. These parameters can be divided into two groups: general system parameters and simulation specific settings.

3.4.1 General system parameters

The general system parameters are parameters that describe a given system and do not depend on the choice of protocol. The relevant parameters that are implemented in the program are listed below.

- Source generation rate
- Source efficiency
- Fiber length
- Fiber loss
- Detector efficiency
- Perturb probability
- State of polarisation (SOP) mean deviation

The source generation rate ν_S is given in megahertz and is the frequency of which the source can generate qubits (or in the case of E91 and BBM92, entangled pairs of qubits). This number is used to determine the key rate, as described in section 3.5.5. The source efficiency η_S is the fraction of these qubits that can actually be used in QKD, due to losses when for example encoding the qubits into the desired states. The frequency of qubits sent by Alice is thus given by the product of the source generation rate and the source efficiency $\nu_S\eta_S$. In general, a number of system parameters will determine the actual generation rate and source efficiency of a quantum source, however in order to keep the tool general only the combined source efficiency is requested from the user.

The fiber length d is given in kilometers and the fiber loss L is given in decibels per kilometer. The total attenuation in the fiber given in decibels is simply calculated as dL. The detector efficiency η_D represents the probability that a received qubit is not registered in the final measurement. In the program, the effects of the source efficiency, the fiber attenuation and the detector efficiency are simulated by removing each generated qubit with a probability of $1 - \eta_S \cdot 10^{-dL/10} \cdot \eta_D$.

The perturb probability is the probability that a photon receives a uniformly random rotation of its polarisation when travelling in the fiber. In reality, almost all photons in the fiber will have their polarisation

slightly altered by random perturbations, which is known as polarisation drift [24]. These deviations in the desired polarisation are corrected for by also sending reference signals in the fiber that can be used to determine the shift of polarisation for each photon and compensating for it in the measurements. However, a small percentage of the photons will not be compensated for correctly and this will result in a portion of all photons having randomly rotated polarisation when measured. The perturb probability is therefore implemented to simulate this behavior by adding RY-gates to randomly chosen qubits with a randomly chosen angle between 0 and π on the Bloch sphere.

Even when corrections are made to compensate for polarisation drift, the recieved qubit states are still not going to be entirely perfect and still contain a certain amount of errors due to imperfections in the compensation of polarisation drift. The SOP mean deviation is the average amount of deviation from the desired SOP in qubits that have been compensated for polarisation drift and is given in radians. To simulate this behaviour, an angle θ is added to each qubit. This angle θ is assumed to be normal distributed with a mean $\mu = 0$ and a standard deviation of $\sigma = \sqrt{\frac{2}{\pi}}\theta$. In Qiskit this is implemented by adding an RY(θ)-gate to each qubit after the initial state preparation but before measurements. Since the mean deviation from the average in a normal distribution is given by $\sqrt{\frac{\pi}{2}}\sigma$ the SOP mean deviation given by the user must be scaled by $\sqrt{\frac{2}{\pi}}$ in order to acquire the correct value of σ.

3.4.2 Simulation specific settings

The simulation specific settings are the parameters given by the user that are specific to a certain simulation and not specific to the system the simulation is run on. These settings are the choice of protocol, the number of qubits generated and the QBER cross check fraction. The user can also enable or disable certain system parameters.

The choice of protocol and the number of qubits sent simply determine which protocol the user wants to use in the simulation and how many qubits should be simulated with that protocol.

The user can also enable and disable losses, perturbations and SOP deviations. This way the user can simulate and identify the effects of each of these system parameters separately. For the same reason, eavesdropping can also be enabled or disabled by the user for each simulation.

To calculate the quantum bit error rate (QBER) in the final key, the QBER cross check fraction is given to decide what fraction of the final key should be used to cross check between Alice and Bob for errors. See section 3.5.1 for a more detailed explanation of how the error rate is calculated.

3.5 Simulation results

After a user has run a simulation with the given parameters and settings, relevant results need to be calculated and presented. The metrics chosen to focus on in this program are described below and were chosen because they are the most common metrics used to measure the behavior and performance of a QKD system.

3.5.1 QBER

The QBER is a common metric used to determine the quality and security of a communication. An approximation of the QBER for a given simulation is calculated by sampling a fraction of the final keys acquired by Alice and Bob and calculating what percentage of the bits match. The size of the fraction used is determined by the QBER cross check fraction parameter that is given by the user. The qubits used to approximate the QBER will then be removed from the final key since their values have to be communicated on the public classical channel. A small QBER cross check fraction will therefore result in a longer final key, with the disadvantages of giving a worse approximation of the QBER, and vice versa.

3.5.2 The S statistic

The S statistic is only provided in the simulation results when the E91 protocol is used, since it is the only protocol that uses the specific entangled states required to perform the CHSH Bell test as described in section 2.3. This can be used to detect eavesdropping and measure system errors.

3.5.3 Combined efficiency

The combined efficiency η_{tot} is the fraction of qubits generated by Alice that are actually measured by Bob. The combined efficiency is thus a metric of the total amount of losses occuring in the system and can be calculated as

$$\eta_{tot} = \eta_S \cdot 10^{-dL/10} \cdot \eta_D \tag{13}$$

where η_S is the source efficiency, η_D is the detection efficiency, d is the fiber length and L is the fiber loss.

3.5.4 Key length

The key length is the number of bits in the final key acquired by both Alice and Bob. Since the losses are implemented by randomly removing each qubit with a certain probability, the key rate will vary slightly between different simulation runs. The expectation value N_K of the key length is however given as

$$N_K = N \cdot \eta_{tot} \cdot (1 - f) \tag{14}$$

where N is the number of generated qubits, η_{tot} is the combined efficiency and f is the QBER cross check fraction.

3.5.5 Key rate

The key rate ν_K is the rate at which the bits in the final key are generated. Note that the key rate is calculated last, after a certain fraction f of qubits have been used in order to approximate the QBER. Given a source generation rate ν_S and a combined efficiency of $\eta_{tot} = \eta_S \cdot 10^{-dL/10} \cdot \eta_D$ the expectation value of the key rate is given by

$$\nu_K = \nu_S \cdot \eta_{tot} \cdot (1 - f). \tag{15}$$

Just like the key length, the key rate will also vary slightly between different simulation runs, due to the implementation of the losses.

3.6 Multiple simulations

After the ability to run single simulations had been implemented, the ability for the user to run multiple simulations automatically was implemented as well in a separate tab in the GUI. The user can chose either a system parameter or the QBER cross check fraction to be varied in a specified interval and the results are shown in a plot. The user can chose which simulation result to be displayed on the y-axis and how many data points in the specified interval that should be simulated. For the remaining parameters, the configuration in the tab for single simulations is used.

3.7 Viewing a sample of the circuit

Lastly, a feature was added to the GUI where a sample of 10 qubits is combined into a single quantum circuit which is then presented graphically to the user. Separate stages in the QKD protocol, such as encoding or decoding, are separated in the summarized circuit using vertical lines known as barriers.

3.8 Evaluating program performance

In order to examine the ability of the program to generate realistic results, comparisons were made to data from three QKD experiments using the BB84 protocol. For each experiment, the required parameters were extracted from the corresponding paper. For the first system [25], the 'C-band source efficiency' and the 'encoder efficiency' as referred to in the paper were combined into the source efficiency used for the simulations. For the second system [26], the 'misalignment probability' as referred to in the paper was converted into the perturb probability used for the simulations. The third system [27] was not fiber based,

so only the key rate was used for comparison. When the papers did not contain information regarding the SOP mean deviation or the polarisation drift correction efficiency, default values were used. According to a recent paper examining novel techniques for polarisation compensation, a SOP mean deviation of 0.13 radians over a 40 km fiber was acquired [28]. Another paper found that the effective compensation of polarisation drift was achieved for about 95 % of received photons in their experiment [29]. These values were thus used as the default values. For each system being simulated, 10 separate simulations were run using 10^7 in order to calculate the mean and standard deviation of the result metrics. A summary of all the system settings used for the different simulations are seen below in Table 5.

System	Source rate [MHz]	Source efficiency [%]	Fiber length [km]	Fiber loss [dB/km]	Detector efficiency [%]	Perturb probability [%]	SOP mean deviation [rad]
ref. [25]	72.6	5.8	18	0.53	6.27	5	0.13
ref. [26]	160.7	1.42	52	0.19	65.25	0.6	0.13
ref. [27]	80	8	9.6*	1*	24	N/A	N/A

Table 5: Parameter values for comparisons to QKD experiments. *System is not fiber based so the values are chosen to represent the total attenuation.

Additionally, simulations were run for all three systems with increasing values of fiber attenuation in order to examine the effect on the key rate. The attenuation was varied from 1 dB to 25 dB and 10^7 qubits were simulated for each attenuation value.

To further examine the performance of the program, additional simulations were run using the BB84 protocol and the E91 protocol with different settings to demonstrate that the program is able to produce qualitatively reasonable results. For both protocols, simulations were run with SOP deviation and perturb probability disabled and enabled, and with eavesdropping disabled and enabled, resulting in four different combinations of simulation settings. The remaining system parameters were set to the values used for ref. [25] as seen in Table 5. For each combination, 10 simulations were run using 10^7 qubits and the mean and standard deviation of the relevant result metrics were then calculated.

Since the values used for the SOP mean deviation and the perturb probability were taken from separate sources, additional simulations were run while varying these simulation parameters to investigate their effect on the results. When using the BB84 protocol, the effect on the QBER was studied while varying the perturb probability between 0 % and 10 % and varying the SOP mean deviation between 0 rad and 0.30 rad. When using the E91 protocol, the effect on the S statistic was studied while varying the perturb probability between 0 % and 20 % and varying the SOP mean deviation between 0 rad and 0.60 rad. As before, the remaining system parameters were the ones used for ref. [25] in Table 5.

4 Results

4.1 Final program

The resulting program consists of two tabs for simulating single or multiple simulations with layouts corresponding to the desired features as described in section 3. Below in Figure 9, 10 and 11 are screenshots of the final program.

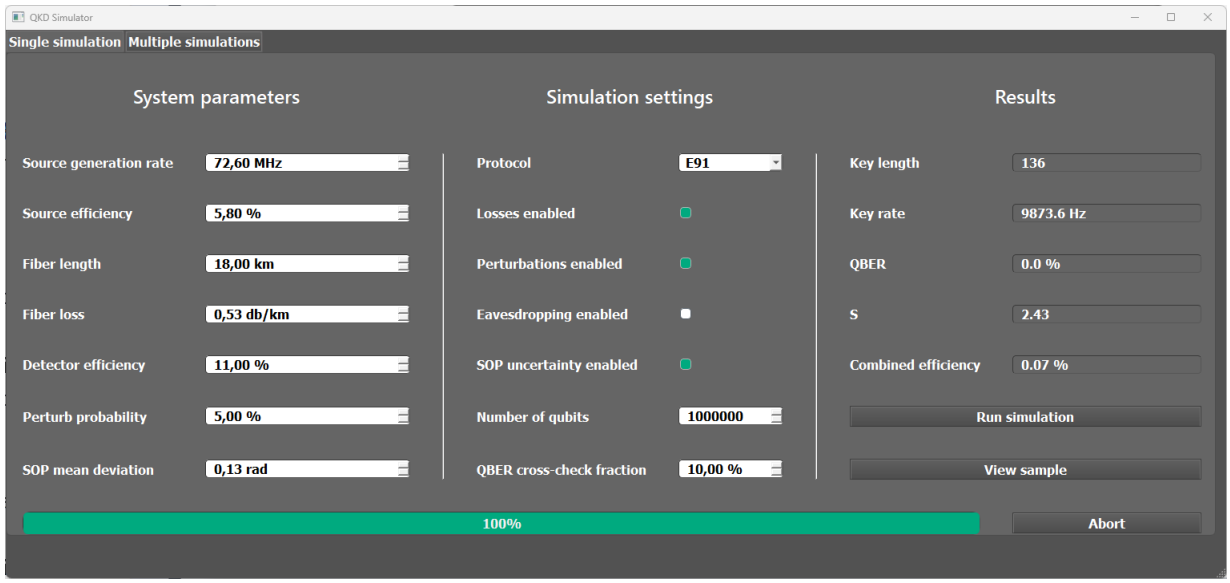

Figure 9: First tab of the GUI, used to run single simulations.

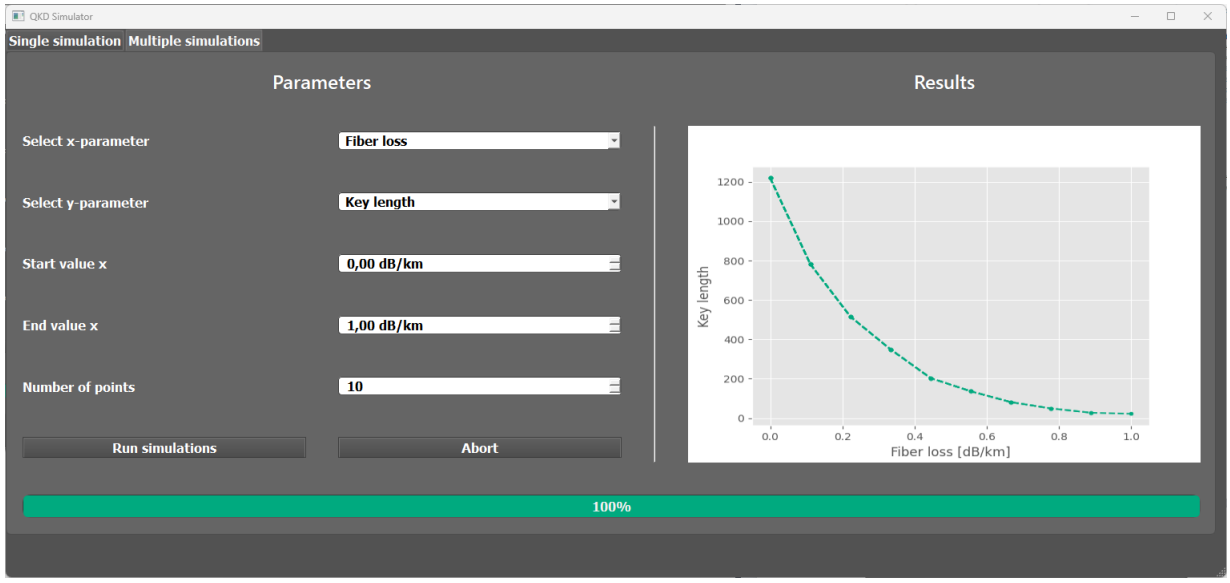

Figure 10: Second tab of the GUI, used to run multiple simulations with varying parameters.

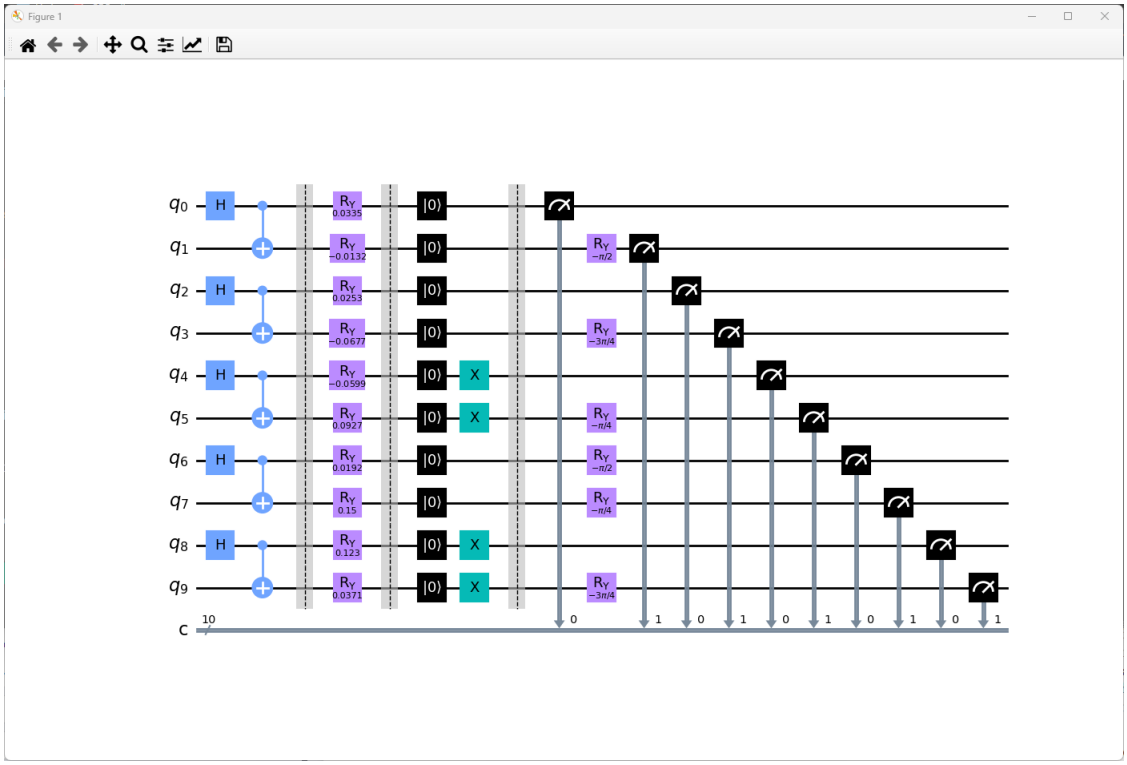

Figure 11: A sample of the final circuit used in order to simulate a protocol. The image above is a sample from a simulation using the E91 protocol with SOP uncertainty, perturbations and eavesdropping enabled.

In the tab for single simulations seen in Figure 9, the user enters the parameters related to the system in the left-most column. The user then enters the desired simulation specific settings in the middle column and presses 'Run simulation'. The progress bar at the bottom shows how far along the simulation has come and when it has finished the results are displayed in the right-most column. The user can abort the simulation by pressing 'Abort' and can view a sample of the quantum circuits used in the simulation by pressing 'View sample' after the simulation has completed. An example of this is shown in Figure 11.

In the tab for multiple simulations seen in Figure 10, the user chooses which parameter to vary and specifies the interval in which it should be varied. The user also specifies how many data points should be simulated in this interval, as well as which result metric that should be plotted on the y-axis. The user then presses 'Run simulations' and the progress bar shows how far along the simulations have come. The option to abort is also present here by pressing 'Abort'. The plot to the right will continuously update with each simulation that finishes. If the user specifies an interval of zero width, e.g. start value and end value being the same, the program will switch to plotting multiple runs of the exact same simulation setup and when it has finished show the mean and standard deviation of the chosen result metric in the plot. This functionality allows the user to study the statistical uncertainty for a given simulation.

4.2 Evaluating program performance

The results from the comparisons to QKD experiments are shown in Table 6 below. For the first system, the resulting QBER was 3.30 ± 1.46 % and the key rate was 13 204 ± 214 Hz whereas the experimental results were a QBER of 3.25 % and a key rate of 13 200 Hz. For the second system, the resulting QBER was 0.74 ± 0.25 % and the key rate was 68 637 ± 529 Hz whereas the experimental results were a QBER of < 2.00 % and a key rate of 94 800 Hz. For the third system, the resulting key rate was 75 646 ± 450 Hz whereas the experimental result was a key rate of 35 000 Hz.

System		SOP deviation	Perturb probability	QBER [%]	Key rate [Hz]
ref. [25]	Simulation Experiment	0.13 rad	5 %	3.30 ± 1.46 3.25	13 204 ± 214 13 200
ref. [26]	Simulation Experiment	0.13 rad	0.6 %	0.74 ± 0.25 < 2.00	68 637 ± 529 94 800
ref. [27]	Simulation Experiment	N/A	N/A	N/A	75 646 ± 450 35 000

Table 6: Results from comparisons to QKD experiments

The effect on the key rate of varying the fiber attenuation is shown in Figure 12 below. The actual fiber attenuations in the experimental setups and the corresponding key rates are inserted for reference.

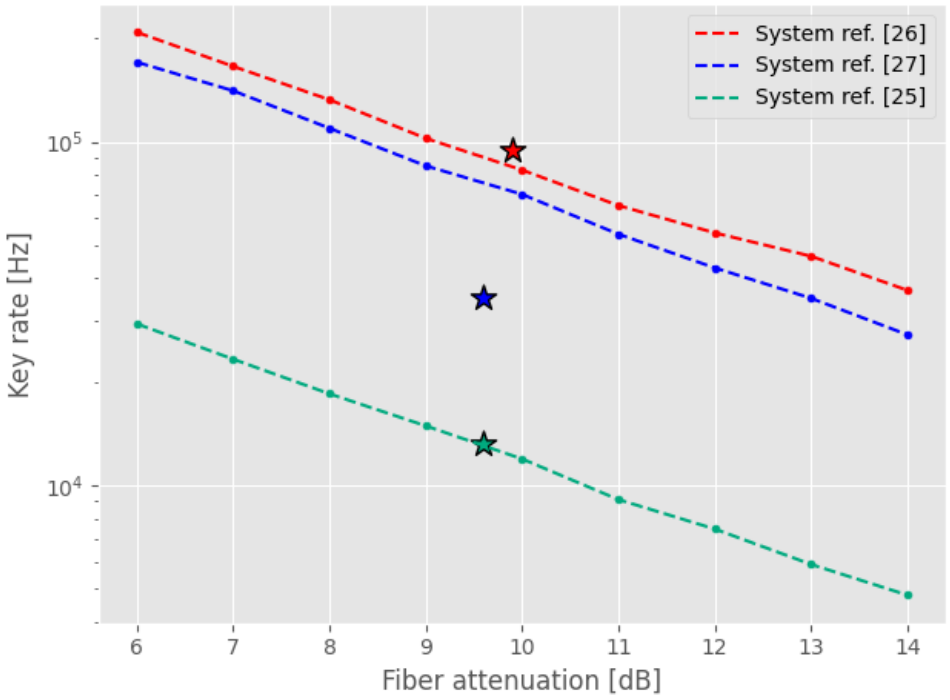

Figure 12: Effect of fiber attenuation on key rate for the three systems used for comparison

When running the example simulations of the BB84 protocol with SOP deviation, perturb probability and eavesdropping all disabled the QBER was 0 %. For the simulations with SOP deviation and perturb probability enabled and eavesdropping disabled, the QBER increased to 3.02 ± 1.18 %. For the simulations with eavesdropping enabled, the QBER increased to approximately 25 %. A summary of the settings and results of the BB84 simulations are seen below in Table 7.

Sim.	SOP deviation	Perturb probability	Eavesdropping	QBER [%]	Key rate [Hz]
1	0.00 rad	0 %	Disabled	0 ± 0	13230 ± 222
2	0.13 rad	5 %	Disabled	3.02 ± 1.18	13242 ± 186
3	0.00 rad	0 %	Enabled	24.41 ± 3.22	13175 ± 188
4	0.13 rad	5 %	Enabled	25.88 ± 2.63	13216 ± 216

Table 7: Results from demo BB84 system with same system parameters as ref [25].

When running the example simulations of the E91 protocol with SOP deviation, perturb probability and eavesdropping all disabled the QBER was 0 % and the average value of the S statistic was 2.83 ± 0.04.

For the simulations with SOP deviation and perturb probability enabled and eavesdropping disabled the QBER increased to 6.24 ± 3.43 % whereas the value of the S statistic decreased to 2.45 ± 0.06. For the simulations with eavesdropping enabled, the QBER increased and the S statistic fell below the classical limit. A detailed summary of the settings and results of the E91 simulations can be seen in Table 8 below.

Sim.	SOP deviation	Perturb probability	Eavesdropping	QBER [%]	S	Key rate [Hz]
1	0.00 rad	0 %	Disabled	0	2.83 ± 0.04	5854 ± 266
2	0.13 rad	5 %	Disabled	6.24 ± 3.43	2.45 ± 0.06	5997 ± 154
3	0.00 rad	0 %	Enabled	37.8 ± 3.6	1.41 ± 0.06	5829 ± 103
4	0.13 rad	5 %	Enabled	39.4 ± 2.59	1.41 ± 0.07	5891 ± 152

Table 8: Results from demo E91 system with same system parameters as ref [25].

The influence on the QBER when varying the perturb probability and SOP mean deviation is shown in Figure 13 below. The influence on the S statistic when varying the perturb probability and SOP mean deviation is shown in Figure 14 below.

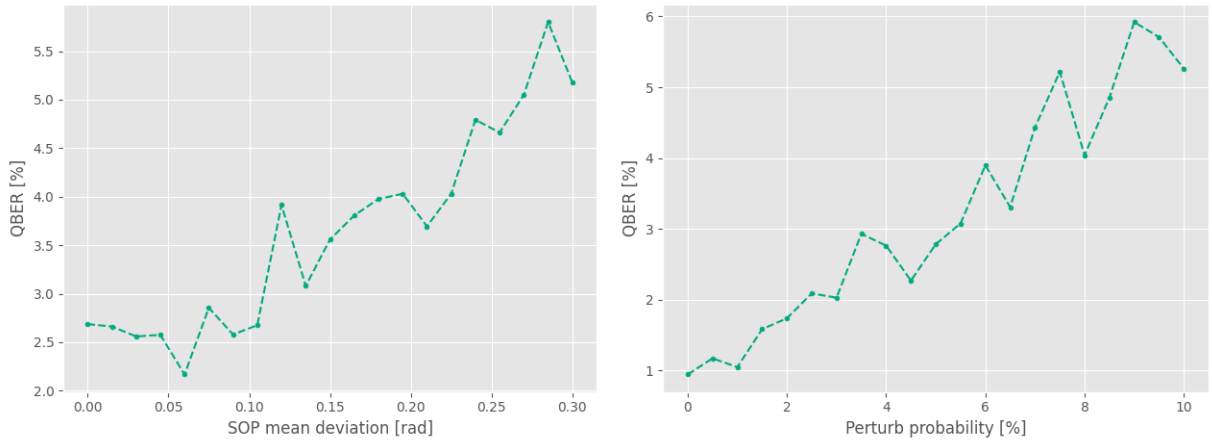

Figure 13: Simulation results showing how the QBER varies when varying the SOP mean deviation (left) and the perturb probability (right).

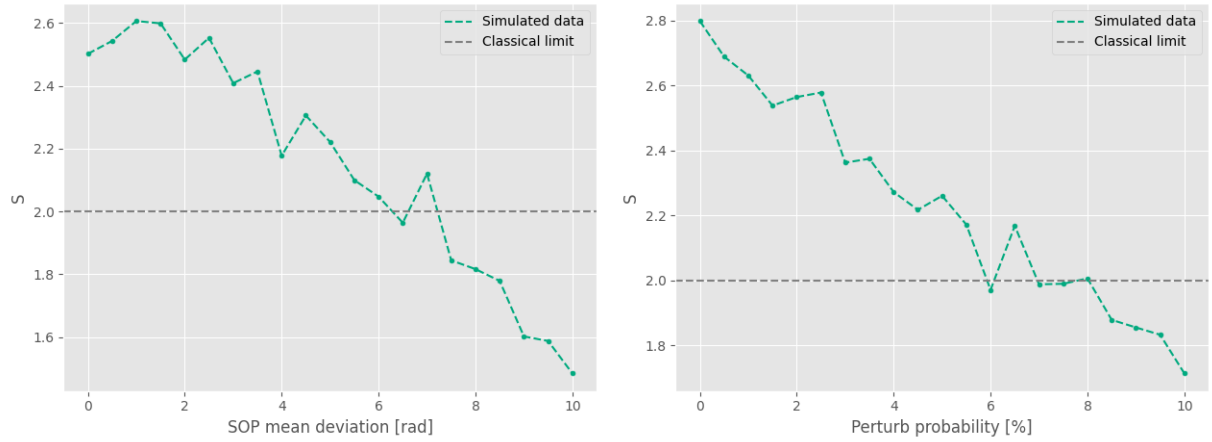

Figure 14: Simulation results showing how the S statistic varies when varying the SOP mean deviation (left) and the perturb probability (right).

5 Discussion

5.1 Evaluating program performance

The comparisons to QKD experiments showed that the program can produce simulation results in the same order of magnitude as the real experimental data.

For the first system [25], the mean QBER of 3.30 % was very close to the QBER of 3.25 % from the experiment. This may indicate that using these settings produces realistic simulation results of the QBER for this system. One must however note that even though the mean QBER matched the experimental data well the QBER fluctuated between different simulation runs resulting in a standard deviation of 1.46 % for the QBER. The simulated key rate was 13 204 ±214 Hz, which was close to the experimental value of 13 200 Hz. A possible explanation for the close agreement to the experimental data for the key rate is that all relevant system parameters were clearly stated in the paper.

For the second system [26], the QBER of 0.74 ± 0.25 % agreed with the experimental QBER result of < 2.00 %. The key rate of 68 637 ± 529 Hz was lower than the experimental key rate of 94 800 Hz. This result is not expected since the program simulates an ideal system which should result in a higher simulated key rate compared to the experimental value. This suggests that the simulated system has larger losses than the experimental system. Despite this, the simulated result is well within the same order of magnitude as the experimental data.

For the third system [27], the only result metric of interest was the key rate, due to the system not being fiber based. The simulated value of the key rate was 75 646 ± 450 Hz while the experimental value of the key rate was 35 000 Hz. This indicates that there might be additional sources of loss in the experimental setup, that are not reflected in the simulation. As previously however, the simulated result is still within the same order of magnitude as the experimental value.

Furthermore, for all three systems it can be seen in Figure 12 that key rate decreases exponentially with the fiber attenuation, which is the expected behavior.

Examining the results from the BB84 and E91 example simulations seen in Table 7 and Table 8, they appear to be qualitatively reasonable. When SOP deviations, perturbations and eavesdropping is disabled the average value of the S statistic is close to $2\sqrt{2}$, which is the expected value from theory. When perturbations, SOP deviation and eavesdropping is enabled, the QBER increases and the S statistic falls below the classical limit, which is also expected from theory. These results support the usefulness of the program as an educational tool. Note however that these results were not compared to real experimental data and so the accuracy of the quantitative results can therefore not be determined.

When examining the effect of the perturb probability and the SOP mean deviation on the QBER and S statistic, they correlate positively with the QBER as seen in Figure 13 and negatively with the S statistic as seen in Figure 14. These results are expected since a higher perturb probability and SOP mean deviation should imply more errors in the final key and worse entanglement fidelity. As before, this could indicate that the program is useful as an educational tool. Note once again that it is only possible to comment the qualitative aspects of these results since they were not compared to experimental data. Comparing Figure 13 to the results from the comparison to the QKD experiments as seen in Table 6, it can be seen that even though the default values chosen for the perturb probability and SOP deviation may not be entirely accurate, the influence of small inaccuracies in these values on the final result is not critical.

5.2 Program limitations

Only a limited amount of protocols have been implemented and to improve the flexibility of the program more protocols would have to be added. Certain protocols, such as the six-state protocol, could relatively easily be implemented using the current framework of the program since the underlying principle of encoding single photons into various polarisation states is very similar to what is done when simulating BB84 and B92. Furthermore, in the comparison to experimental data, only the BB84 protocol was tested. This is partially due to the fact that BB84 is a common benchmark of QKD system performance. Additional comparisons would have to be made in order to determine the accuracy of simulations using the other protocols.

Depending on the losses of the system, a large number of qubits may need to be simulated in order to get reliable results. If too few qubits are being simulated and the resulting key length is short, very few qubits are going to be used in order to estimate the QBER and calculate the S-statistic thus making these metrics statistically unreliable. If the QBER and S-statistic fluctuate heavily between simulation runs, that might indicate that more qubits need to be simulated.

As previously mentioned in section 2.5.2, one of the main challenges of QKD is the high transmission losses in optical fibers, making long-distance QKD difficult. A proposed solution to this problem is to use so called quantum repeaters to increase the range of QKD. A limitation of the program is that simulations with quantum repeaters is not possible. As quantum repeaters is a highly researched topic within QKD, allowing for simulations of quantum repeaters could improve the relevancy of the program.

In real QKD systems the photon detectors have a tendency to make faulty counts of photons even when there has been no incident light. The average rate of these false photon detections is known as dark count [30]. In QKD experiments, the dark count of the detectors is often discussed as a possible source of error since it can have a considerable effect on the performance of the QKD system if faulty detections are made. This program does not allow for simulating dark count, which is a limitation of its ability to produce realistic simulations.

5.3 Conclusion

The final program is able to simulate four of the most essential QKD protocols, and provides tools for the user to examine what effect varying system parameters has on QKD performance. By allowing the user to examine each of the implemented protocols as a quantum circuit, this strengthens the programs usefulness as an educational tool. The comparison of the BB84 simulation results to the experiments are in the correct order of magnitude and indicate that the program is able to produce a useful first approximation for the key rate and QBER, given that all required parameters are available. More comparisons to additional experimental data would have to be made in order to determine the accuracy of the simulation results for the other protocols. Furthermore, the program could be improved by allowing for simulations of more complex QKD systems, involving for example quantum repeaters.

References

[1] Marina von Steinkirch. *Introduction to Quantum Information*. New York: State University of New York at Stony Brook, 2011.

[2] David J. Griffiths and Darrell F. Schroeter. *Introduction to quantum mechanics*. Third edition. Cambridge ; New York, NY: Cambridge University Press, 2018. ISBN: 978-1-107-18963-8.

[3] IBM. *What is quantum computing?* URL: https://www.ibm.com/topics/quantum-computing.

[4] Victor Lovic. "Quantum Key Distribution: Advantages, Challenges and Policy". In: *Cambridge Journal of Sceince Policy* 1 (2 2020). URL: https://www.repository.cam.ac.uk/bitstream/handle/1810/311529/CJSP_Paper_Lovic.pdf?sequence=1.

[5] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010, p. 15. DOI: 10.1017/CBO9780511976667.

[6] Smite-Meister. *Bloch sphere*. License: CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0, via Wikimedia Commons. URL: https://commons.wikimedia.org/wiki/File:Bloch_sphere.svg.

[7] IBM. *Multiple Qubits and Entangled States*. URL: https://qiskit.org/textbook/ch-gates/multiple-qubits-entangled-states.html.

[8] Gregg Jaeger. *Quantum Information: An overview*. Boston: Springer, 2007.

[9] David J. Griffiths and Darrell F. Schroeter. *Introduction to quantum mechanics*. Third edition. Cambridge ; New York, NY: Cambridge University Press, 2018, pp. 449–452. ISBN: 978-1-107-18963-8.

[10] Alain Aspect, Philippe Grangier, and Gérard Roger. "Experimental Realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: A New Violation of Bell's Inequalities". In: 49.2 (July 1982), pp. 91–94. DOI: 10.1103/PhysRevLett.49.91.

[11] George Stamatiou based on png file of C.Thompson. *Schematic of a "two-channel" Bell test*. License: CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0, via Wikimedia Commons. URL: https://commons.wikimedia.org/wiki/File:Two_channel_bell_test.svg.

[12] Huw Fox and Bill Bolton. "8 - Logic gates". In: *Mathematics for Engineers and Technologists*. Ed. by Huw Fox and Bill Bolton. IIE Core Textbooks Series. Oxford: Butterworth-Heinemann, 2002, pp. 270–290. ISBN: 978-0-7506-5544-6. DOI: https://doi.org/10.1016/B978-075065544-6/50009-6. URL: https://www.sciencedirect.com/science/article/pii/B9780750655446500096.

[13] IBM. *Defining Quantum Circuits*. URL: https://qiskit.org/textbook/ch-algorithms/defining-quantum-circuits.html.

[14] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Review* 41.2 (1999), pp. 303–332. DOI: 10.1137/S0036144598347011. URL: https://doi.org/10.1137/S0036144598347011.

[15] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010, p. 586. DOI: 10.1017/CBO9780511976667.

[16] Artur Ekert et al. "Eavesdropping on quantum-cryptographical systems". In: *Physical review. A* 50 (Sept. 1994), pp. 1047–1056. DOI: 10.1103/PhysRevA.50.1047.

[17] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010, p. 532. DOI: 10.1017/CBO9780511976667.

[18] Charles H. Bennett and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Theoretical Computer Science* 560 (Dec. 2014), pp. 7–11. DOI: 10.1016/j.tcs.2014.05.025. URL: https://doi.org/10.1016%5C%2Fj.tcs.2014.05.025.

[19] Raman Research Institute. *B92 Protocol*. URL: https://wwws.rri.res.in/quic/qkdactivities.php.

[20] Artur K. Ekert. "Quantum cryptography based on Bell's theorem". In: *Phys. Rev. Lett.* 67 (6 Aug. 1991), pp. 661–663. DOI: 10.1103/PhysRevLett.67.661. URL: https://link.aps.org/doi/10.1103/PhysRevLett.67.661.

[21] Nikolina Ilic. "The Ekert Protocol". In: *JOURNAL OF PHY334* (2007).

[22] Charles H. Bennett, Gilles Brassard, and N. David Mermin. "Quantum cryptography without Bell's theorem". In: *Phys. Rev. Lett.* 68 (5 Feb. 1992), pp. 557–559. DOI: 10.1103/PhysRevLett.68.557. URL: https://link.aps.org/doi/10.1103/PhysRevLett.68.557.

[23] Qiskit Development Team. *Qiskit 0.42.1 documentation*. URL: https://qiskit.org/documentation/.

[24] J Chen et al. "Stable quantum key distribution with active polarization control based on time-division multiplexing". In: *New Journal of Physics* 11 (June 2009), p. 065004. DOI: 10.1088/1367-2630/11/6/065004.

[25] Mujtaba Zahidy et al. *Quantum Key Distribution using Deterministic Single-Photon Sources over a Field-Installed Fibre Link*. 2023. URL: https://arxiv.org/abs/2301.09399.

[26] Christopher L. Morrison et al. *Single-emitter quantum key distribution over 175 km of fiber with optimised finite key rates*. 2022. arXiv: 2209.03394 [quant-ph].

[27] Ghulam Murtaza et al. "Efficient room-temperature molecular single-photon sources for quantum key distribution". In: *Optics Express* 31.6 (Feb. 2023), p. 9437. DOI: 10.1364/oe.476440. URL: https://doi.org/10.1364%2Foe.476440.

[28] Olinka Bedroya et al. *Resource-Efficient Real-Time Polarization Compensation for MDI-QKD with Rejected Data*. 2022. arXiv: 2209.02707 [quant-ph].

[29] Rende Liu et al. "Analysis of polarization fluctuation in long-distance aerial fiber for QKD system design". In: *Optical Fiber Technology* 48 (2019), pp. 28–33. ISSN: 1068-5200. DOI: https://doi.org/10.1016/j.yofte.2018.12.012. URL: https://www.sciencedirect.com/science/article/pii/S1068520018303183.

[30] RP Photonics Encyclopedia. *Photon Counting*. URL: https://www.rp-photonics.com/photon_counting.html.