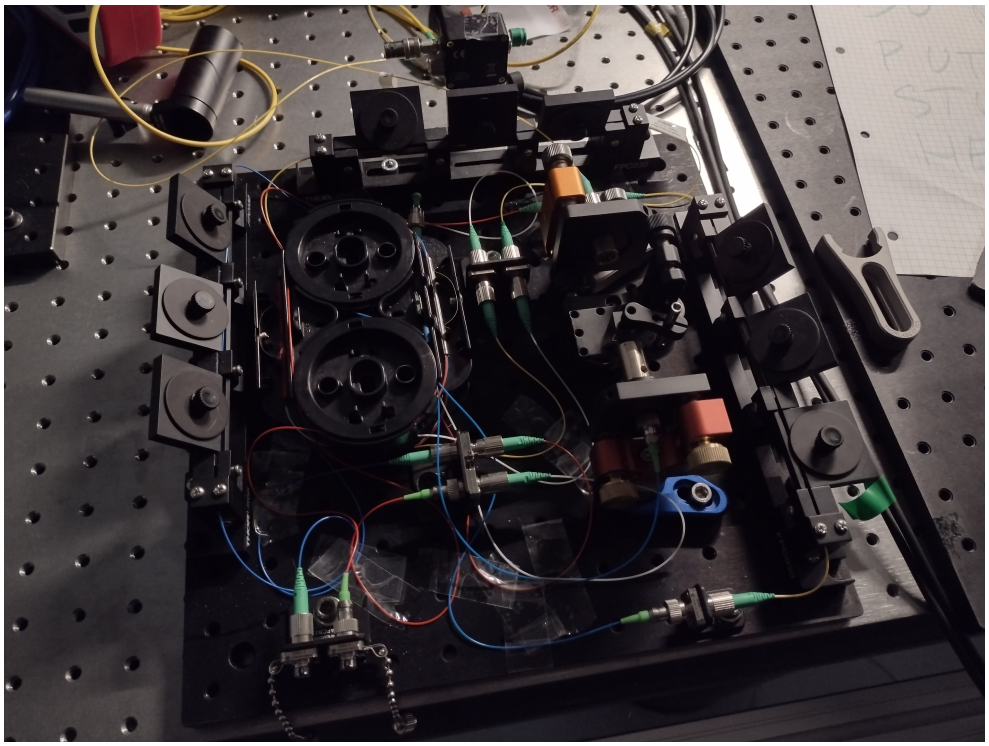
UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE



Master thesis

Single photon sources and applications

Martin A. Krehbiel

Supervisor: Leonardo Midolo, Mikkel Mikkelsen, Beatricie da Lio

Submitted: February 14, 2023

This thesis has been submitted to The Faculty of Science, University of Copenhagen

Contents

Chapter 1

Preamble

Front-page image

The front-page image is of the quadrupler discussed in chapter 3.

Thank yous

I would like to thank my supervisors, Leonardo, Beatricie and Mikkel, for guiding me through this project, I truly could not have done this without them. I would also my parents and my brother for their emotional support and (often unrequested) advice. This thesis really would not exist without these people.

I also owe a special thanks to Ying, Patrik and Eva for helping me out with some technical advice when I needed it most.

Finally a more general thank you to the entire Hy-Q team for inviting me to be a part of their social environment.

The project description

Quantum Key Distribution (QKD) is a method for sharing a cryptographic key between two parties, Alice and Bob, and is the only such method that does not rely on assumptions of a problem being difficult to solve with limited resources but on fundamental principles of physics. Working in the photonics group at NBI, I will develop a pulse quardrulper that will enhance the key rate of such systems while also working on the coding for distilling the raw key produced by the systems into a secure key. The main methods to be used in the project is the creation of numerical models and lab-work with fibre optics.

Chapter 2

Introduction to quantum key distribution

To understand the further sections of this thesis it is necessary to have a basic grasp of quantum key distribution (QKD). Therefore this section will cover the basic task of cryptography and what problem quantum key distribution aims to solve, the origins of security in QKD, an example of a QKD protocol, the practical setup in the laboratory at NBI, and finally the secret key rate and a formula for calculating it.

The problem at hand

In order to understand the problem which Quantum key distribution tries to solve it is necessary to go over the problem which encryption tries to solve, which goes as follows:[25, ch. I.a.1]

There at two parties Alice and Bob who wish to send a secret message from Alice to Bob (or vice versa). The means to share this secret message is an insecure channel. Insecure in this context means that an eavesdropper, often called Eve, can read and edit any messages sent across the channel. Eve can also write her own messages and send them to Alice and Bob. In order to ensure the secrecy of their message Alice and Bob are allowed to agree to a protocol, with which to exchange the message, beforehand. It is assumed that Eve knows this protocol in its totality.[1, ch 2.1]

One simple such protocol would be to replace every letter in the message with the one that follows alphabetically.[1]

Eve is granted her own resources, with which to decrypt the message sent by Alice. Eavesdroppers can categorised according to the resources that they have available to them. In this thesis we will only concern ourselves with an

[1] Since Eve would know the protocol this would be very insecure.

Plain text Message	A	z
Message in ASCII	01000001	01111010
Key	10010011	10101000
Encrypted message	11010010	11010010

Table 2.1: In this example Eve intercepted the message {11010010}. She contemplates two scenarios. Either the original message was a capitalised a and the key is 10010011 or the message was a lowercase z and the key is 10101000. Both plain text messages could result in the the same encrypted message and Eve does not have enough information to tell which. The same could be said for any two characters.

unconstrained eavesdropper. An unconstrained eavesdropper has the following constraints:

1. Eve has no access to the laboratories of Alice and Bob.

2. Eve cannot violate the laws of nature as we currently understand them.

It is worth while to consider the resources that Eve does have access to in this context: unchecked access to the open channel, infinite classical computational power, infinite quantum computational power, infinite quantum memory that never undergo decoherence and the ability to measure any signal between Alice and Bob to any precision allowed by the laws of physics.

It is then surprising to note that Alice and Bob can solve this problem quite trivially without any quantum devices. To do so they simply need some random string of numbers that they both have a copy of, and which Eve does not. This number is called a key. They can then flip every bit in their message if the corresponding bit in the key is 1 and leave it unchanged if the corresponding bit is 0.[1, ch. 2.1.1]

A simple example of why this form of encryption works is given in table 2.1. The key thing to note about that example is that the choice of characters that eve is contemplating is arbitrary. For any plain text message and encrypted message there is a key that encrypts that plain text message as the encrypted message.

The only two problems with this method is that it requires a key as long as the message and that keys cannot be reused.[1, ch. 2.1.1] Reusing an old key risks revelling some information about the key itself. This is a great challenge as it can be proven that Alice and Bob cannot generate a key over an open channel.[19, ch. II.B.1][2] This is the problem that quantum key distribution solves.

[2]More precisely they cannot generate a key with more information unknown to Eve than the key they already share without additional resources, such as a quantum channel.

The advantage of quantum

QKD aims to distribute a key between Alice and Bob, without revealing it to an unconstrained eavesdropper.

Doing QKD requires, in addition to the open classical channel, an open-imperfect quantum channel. A quantum channel is a channel along which qubits can travel from Alice to Bob.[19, ch. I.b.1] Open means that the Eve can read and write messages to the channel. Imperfect means that the channel may randomly alter or loose some of the signals travelling through it. Individual quantum signals, that can take one of two values when measured, are referred to as quantum bits or qubits.

The advantage of using a quantum channel lies in the measurement theorem. To recap, the measurement theorem goes as follows:[18, ch. 1.4]

1. When measuring a quantum state one will always find it in an eigenstate of the measured operator[3].

2. The probability of finding a state $\langle\psi|$ in the eigenstate $|\alpha\rangle$ is given by: $|\langle\psi|\alpha\rangle|^2$.

3. After a state is measured to be in a state $|\alpha\rangle$ it is in that state.

What this boils down to in practise is that if the protocol instructs Alice to send a qubit to Bob as an eigenstate of one of two incompatible observables (i.e. two observables with different eigenstates[18, ch. 1.4]), chosen by Alice at random, then Eve can measure the particle, but unless she knows what basis Alice choose she has to guess the basis (see point 1). If Eve guesses correctly she can gain exact knowledge of the value of the qubit and may forward it to Bob unaltered. Note that she has to forward something or the state will be removed during sifting (see below). However, if Eve guesses wrong she will get a random result and cannot forward the state, sent to her by Alice, to Bob. If the qubit forwarded by Eve is measured by Bob after she performed her measurement, using the wrong basis, then Bob has a chance of getting a result different from what Alice sent.[19, ch. I.b.2] This is true even if he choose the right basis.

Again, even if Alice and Bob have flawless equipment and Bob measures in the very same basis that Alice sent her qubit in and even if no errors occurred while the qubit travelled through the channel; there is still a chance that they get different results if there is an eavesdropper. More bluntly, eavesdropping on a quantum channel creates errors. These errors can be detected if ever Alice and Bob compare their results which they can do at their leisure over the classical channel.

[3]The measured operator simply means the operator of the variable that we are trying to measure

Also Eve cannot make a perfect copy (called a clone) of the qubit due to the no cloning theorem.[27]

However there are still the following unresolved issues before we have a proper protocol for key distribution:

1. In an imperfect channel wouldn't there be lots of errors anyway?
2. If Alice and Bob are comparing their measurements over an unsecured channel aren't they revealing the key to any eavesdropper?
3. Couldn't Eve just impersonate Alice or Bob and convince them that no errors were found?
4. How would Bob know what basis to measure in?
5. What if Eve just get really really lucky?

An example of one way that these issues can be resolved would be the BB84 protocol.

The BB84 protocol

The BB84 protocol is a protocol for QKD that assumes that Alice and Bob have a classical channel and a quantum channel, as described in the preceding sub-chapter, and that they also share an initial key that they intend to grow.

The first step is for Alice to send a series of qubits to Bob. She sends them with a random value in a randomly chosen basis selected from two mutually unbiased basis's that Alice and Bob agreed on as part of the protocol. What measurement outcomes correspond to which value and which basis's are being used are all hardware dependent and do not influence the core protocol. The bits they get out of this is referred to as their raw key.[19, ch. I.b.4] Alice and Bob then post-process their raw key over the classical channel.

This is all rather abstract. In this thesis polarisation encoding was used, so let's use that as an example:

Polarisation encoding means that the qubit is the polarisation of a photon. There are three commonly used bases for the polarisation of a photon: horizontal-vertical (HV or $+$) diagonal-antidiagonal (DA or $\times$) and circular polarisation (which won't be used in this thesis). With $|D\rangle = \frac{1}{\sqrt{2}}(|H\rangle + |V\rangle)$ and $|A\rangle = \frac{1}{\sqrt{2}}(|H\rangle - |V\rangle)$

In the HV basis the binary values are horizontally polarised photons ($|H\rangle$) corresponding to a value of 0 and the vertically polarised photons ($|V\rangle$) which correspond to a value of 1. In the DA basis the values are diagonal polarisation ($|D\rangle$) for 0 and antidiagonal polarisation ($|A\rangle$) for 1.

So when Alice has to chose a random value and a random basis; she flips two coins the first say 1 one side and 0 on the other the second says HV on

one side and DA on the other. If for instance they come up 1 and AD, then Alice sends a photon with anti-diagonal polarisation to bob.

Bob then flips another HV/DA coin and measures the incoming photon in the corresponding basis. Continuing the example above if Alice sends an antidiagonally polarised photon and Bob measures it in the DA basis he will find it to be in the antidiagonal polarisation, because:

$$\begin{aligned} |\langle A|A\rangle|^2 &= 1 \\ |\langle A|D\rangle|^2 &= 0. \end{aligned} \tag{2.1}$$

Alternatively if Bob measures his photon in the HV basis he would get a random result, since:

$$\begin{aligned} |\langle A|H\rangle|^2 &= \frac{1}{2} \\ |\langle A|V\rangle|^2 &= \frac{1}{2}. \end{aligned} \tag{2.2}$$

The second step of BB84 is the so called post-processing. Postprocessing is a classical computational process that involve the following steps:

1. **Sifting** is where Alice and Bob publish what bases they used for what measurements. They then discard all the results for which they used different bases.[1, ch. 5.1.1]

2. They then do **error estimation** where they publish a random subset of their bits in order to count how many errors they got. This has to be a large enough subset to put a tight upper bound on the information Eve could have obtained. The published bits are then discarded.[1, ch. 2.2]

3. **Error correction** is the step where any differences between Alice's copy of the raw key and Bobs copy of the raw key get corrected. Here corrected means that either Alice or Bob flip the bit value of any differing bits or that both Alice and Bob discard the offending bit.[19, ch. III.B.1.a]

4. Finally **privacy amplification** is the step that transforms a long sequence of bits that are partially unknown to Eve into a shorter sequence that is entirely unknown to her.[19, ch. III.B.1.a]

The technical details on how to accomplish theses steps can be found in chapter Postprocessing in practice.

In order to prevent Eve from impersonating either party during post-processing they use their shared key to authenticate the messages that they send over the classical channel.[19, ch. II.B.1] What this means is that every message starts with a little token that is dependent upon both the key and the message. Then when the receiver receives a message they then compute

Alice								
Alice's bit	0	1	0	1	0	0	0	1
Alice's basis	×	×	+	+	×	+	+	×
Alice's photon	D	A	H	V	D	H	H	A
Bob								
Bobs basis	×	+	×	+	×	×	+	×
Bobs measurement	D	?	?	V	D	?	H	A
Bobs bit	0	?	?	1	0	?	0	1
postprocessing								
sifting	0			1	0		0	1
privacy amplification	1			0	0		1	1

Table 2.2: A table showing how the BB84 protocol works. Perfect components has been assumed for clarity. The steps of error estimation and error correction have been left out as there are no errors. ? indicates a random value.

a token using their copy of the key and the message that they received and check that it matches the token that accompanied the message.[1, ch. 2.2.1] Therefore if the message were to be altered by an eavesdropper the two tokens would differ and the receiver would know to discard the forged message. The eaves dropper cannot attach a token of their own to the message because constructing one requires *both* the key and a message, and they do not have the key.

The technical details on how to construct a token such that Eve cannot recover the key is not something that will be covered in this thesis. All that is necessary to know is that such a token can be generated in such a way that it requires only a short key to authenticate a long message.[1, ch. 5.1.1]

An illustration of how the BB84 protocol works is provided in table 2.2

This has answered the issues of the previous protocol that were numbered 2, 3 and 4. Alice and Bob do reveal part of the key under error estimation but they discard those parts of the key, Eve can't impersonate them because the channel is authenticated and Bob doesn't need to know what basis to measure in because Alice and Bob just post select the cases where they happened to use the same basis.

The two remaining issues are "In an imperfect channel wouldn't there be lots of errors anyway?" and "What if Eve is just got really really lucky?". These are answered by the concept of $\varepsilon\delta$-security.

The basic concept of $\varepsilon\delta$-security is that Alice and Bob has some parameter delta in their postprocessing and that for any finite probability ε they can set their variable δ such that the probability of Eve getting the key is at most ε.[19, ch. II.C.2][1, ch. 6.3.2] Accomplishing this against an unrestrained opponent is referred to as information theoretic security.

Note that information theoretic security does not mean absolute security,

but does instead mean the ability to achieve arbitrarily good security. So Eve could just get lucky, but Alice and Bob can make this as unlikely as they see fit.

To solve the problems of there being error anyways it is simply assumed that all errors are caused by Eve. This is likely more paranoia than what is strictly needed,[19, ch. III.B.5] however no one has managed to make a security proof that did not rely on this overly pessimistic assumption. Besides, when dealing with security, paranoia is a virtue not a vice.

QKD in practice

In the real world Alice and Bob are not people, but machines. In this chapter the machines used to make QKD work are described.

Quantum-dots

Before describing the setup it is necessary to give a brief overview of quantum-dots(QD) as these are part of the setup.

In solids electrons can only have certain energies. The energy levels are organised into bands and band gaps. Where the energy band is the range of energy in which there are allowed energy levels and the band gaps is where there are no allowed energy levels.[21, ch. 11.2]

The band below the band gap is referred to as the valance band and the band above the band gap is called the conduction band.[21, ch. 16.1] A structure of bands and band gaps is called a band structure.[11, ch. 3]

In semi conductors, the valance band is full,[21, ch. 16.1] so instead of keeping track of all the electrons we just keep track of the missing electrons, called holes.[21, ch. 17.1] The conduction band is mostly empty so there we just keep track of the electrons. The band structure of GaAs is depicted in subfigure 2.1a.

Since the band structure differs between materials a small island of one material embedded inside another (a QD) will have a small space where electrons (or holes) can exist at energy levels that aren't available in the bulk material[21, ch. 18.1.2] as depicted in subfigure 2.1b. This recreates the square potential well from quantum-mechanics textbooks.

The reason all of this is useful is that an electron can be moved from the valence band in the QD to the conduction band, creating an electron-hole-pair called an exciton[15, ch. II.B] depicted in subfigure 2.1c. Neither the electron nor the hole can move out of the QD, without emitting or absorbing energy. Therefore the exciton cannot move and eventually the electron will go into the hole and release a photon in the process.[15, ch. VI.B][4]

[4]This process does not necessarily emit a photon. Decays that don't emit photons are known as non-radiative decay.

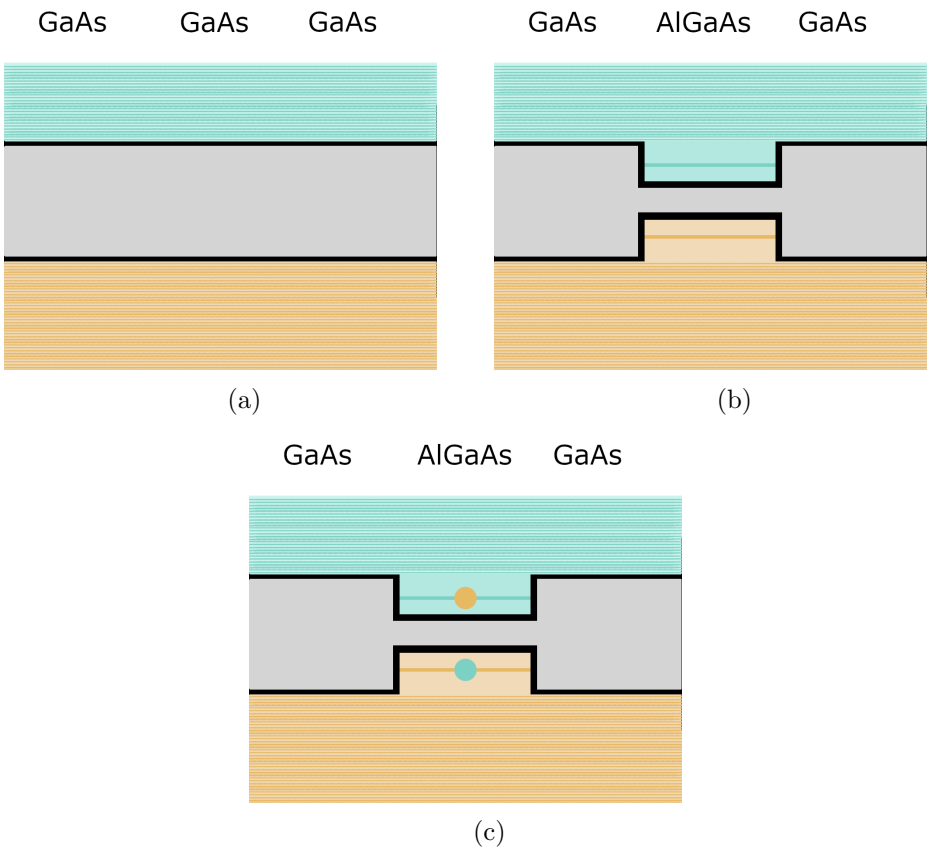


Figure 2.1: These figures show the band-structure of GaAs and AlGaAs. The pale orange area indicate the valance band, the teal area indicates the conduction band and the grey area indicates the band gap. The blue lines in the conduction band indicate states occupied by holes and the orange lines in the valance band indicate states occupied by electrons. Figure 2.1a shows GaAs's band-structure in the absence of a QD. Figure 2.1b shows the band-structure of GaAs with a QD. Figure 2.1c shows the QD from before with the electron on the QD having been excited to the the lowest available energy band. The Blue dot on the orange line represents the hole occupying the state and the orange dot on the blue line represents the electron occupying the state.

By embedding the QD in something called a photonic-crystal-waveguide it is possible to all but ensure that the photon will be emitted into a "guided mode" meaning that the photon is in the waveguide.[15, ch. VI.B] So by stimulating the QD at regular intervals it is possible to create a photons at regular intervals.

In order to excite the QD we use short laser pulses. Such a laser pulse can, if it has the right combination of power and duration exactly excite an electron on the QD. A pulse that does this is called a pi-pulse. So by exciting

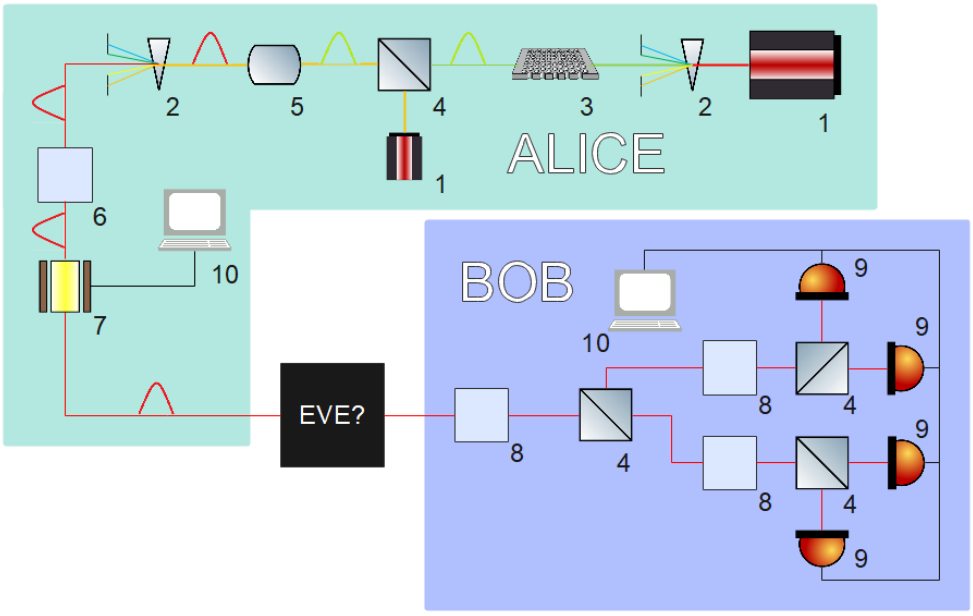


Figure 2.2: The setup of the QKD system. The polarisation controllers were procured by DTU over the summer and were at the start of the project manually adjusted polarisation paddles. 1: Laser; 2:Diffraction grating with a narrow slit; 3: Photonic crystal with a QD; 4: Beamsplitter; 5: Non-linear medium; 6: Polarisation paddles; 7: Polarisation modulator; 8: Polarisation controller; 9: Single photon detectors; 10: Computers.

the QD at a regular interval, using pi-pulses, it is possible to create photons with a regular time-spacing.

Given that this decay only releases a single photon and does so deterministically this setup is called a deterministic single photon source; which is useful for QKD since if Alice releases two photons instead of one Eve can keep one photon and forward the other to Bob, then measure the one she kept after Alice and Bob announce their measurement basis. This would allow her to get complete information about the key.[5][4]

Our setup

The setup in the FIREQ-group is depicted in figure 2.2 and works as follows. First at Alice creates a series of single photons. This is accomplished by exiting a quantum dot (QD) located in a photonic-crystal with a series of laser pulses. A laser pulse which brings the electron from the ground state to the excited state is referred to as a pi-pulse.

[5]This is known as a photon number splitting attack.

The photons emitted by the QD go through a so called shallow-etched grating and are picked up by a fibre. The shallow-etched grating is a device that sends photons out of the plane in which they are moving such that the photons can be picked up by a fibre. It is considered part of the QD for the purposes of figure 2.2. The photons are then sent through a nonlinear medium along with the light from another laser. This converts the light from 930nm wavelength to 1550nm. Because optical fibbers have higher transmission at 1550nm than at 930nm[22] this will reduce losses for all fibre transmission.

The photons then undergo polarisation encoding. The photons are sent into a polarisation controller and then a polarisation modulator. The polarisation controller is adjusted manually to align the polarisation of the incoming light with the optical axis of the polarisation modulator which modulates the the polarisation of the single photons according to an electrical input. Specifically the photons become either horizontally, vertically, diagonally or anti diagonally polarised. We note that horizontal and vertical base states form one orthonormal-basis and that the diagonal and anti-diagonal states form another. This is in principle where Alice ends and the channel connecting Alice and Bob begin.

The channel depends upon the exact setup. In the field trial the photons entered a series of fibres taking it to the Danish Technical University(DTU) whereas in many of the test leading up to that the photons just entered fibre coil. This is useful as a coil can contain many kilometres of fibre in a small volume. Either way the photons are then released into Bob.

The first components in Bob are a polarisation controller and a 50/50 beamsplitter. The polarisation controller adjusts for the polarisation drift that has occurred in the channel between Alice and Bob. The output of the beamsplitter then leads to a polarisation controller and a polarising beamsplitter(PBS). The PBS sorts the photons into two different fibres according to their polarisation. The polarisation controller transform the polarisation state of the photons from one basis to another. This is better understood with an example:

> Lets say that Charlie has a PBS that send vertically polarised light into one fibre and horizontally polarised light into another fibre. However what Charlie wants is one that sorts into the diagonal/anti-diagonal basis. Charlie can do this by placing a polarisation controller in front of his beamsplitter. The polarisation controller then does a unitary transform given by:
>
> $$\begin{aligned} |\nwarrow\rangle &\Rightarrow |\uparrow\rangle \\ |\nearrow\rangle &\Rightarrow |\rightarrow\rangle \end{aligned} \tag{2.3}$$
>
> This means that the system of both components sorts according photons into the diagonal/anti-diagonal basis.

One of the polarisation controllers in Bobs setup perform exactly this transformations, while the other does the identity transformation. This then places the photons received from Alice on four different fibbers according to their polarisation and a passive basis choice[6]. Finally a single-photon-detector(SPD) at the end of each path detects which path the photon is on, and thereby what its original polarisation was.

A computer then gathers all the clicks and turns it into a file upon which post-processing can then be performed.

The upper bound for secret key rate

The final conceptual notion for understanding the rest of this thesis is the distinction between the secret key rate and the raw key rate. The raw key rate is the rate of key generation before postprocessing and secret key rate is the number of bits after postprocessing.[19, ch. II.B.4] Both are usually measured in bits per second.

The secret key rate can be derived by the following formula:[19, ch. IV.B.2]

$$K = R(1 - I_E - leak_{EC}(Q)) \tag{2.4}$$

where K is the secret key rate, R is the raw key rate, I_E is the upper bound on Eves knowledge of the raw key, Q is the quantum bit error rate(QBER) and $leak_{EC}(Q)$ is the information leaked during error correction. So to put the equation into words: the secret key rate is the raw key rate minus all that Eve knows about it and all that Eve can learns about it during error-correction.

Theorem 1. *[19, ch. 4] The upper bound on Eves information on the raw key can be given as:*

$$I_E = 1 - Y_1(1 - h(Q/Y_1)) \tag{2.5}$$

Where Y_1 is the single photon rate, Q is the QBER and $h()$ is the function for the binary entropy function. The single photon rate is the number of pulses that Alice emits that has exactly 1 photon. In real systems it is common to have Alice emitting some 0-photon pulses as well as multi photon pulses. The error rate is given as the fraction of signals where Bob gets the wrong result at measurement. The binary entropy function is given by $h(x) = -x \cdot Log_2(x) - (1 - x) \cdot log_2(1 - x)$.

Proof. To prove this we examine what Eves optimal strategy is given the information available to her. Eve can, without disturbing the value of the qubit, measure the number of photons in a pulse.[19, ch. IV.A.1] Measuring this information is always part of the optimal strategy as she can elect not to make use of it.

[6] A passive basis choice means that Bob does not choose in which basis he would measure any specific photon. In a sense the BS decides in which basis the photon will be measured in.

If there are zero photons Eve can choose whether or not to send a qubit to Bob and while she could send a qubit to Bob this doesn't help her gain information on Alice's key.[7] So the information gained is $I_{E,0} = 0$.[12]

If there is more than one photon then Eve can gain full information since she can perform photon number splitting attacks. So $I_{E,\geq 2} = 0$

Finally if there are exactly one photon then Eve can gain some information, but at the cost of introducing some error according to $I_{E,1} = h(\varepsilon_1)$, where $I_{E,1}$ is the information gained by Eve, on the raw key, if Alice sends 1 photon to Bob and ε_1 is the error rate, between Alice and Bob, when Alice sends exactly one photon.

So Eves best strategy would be to maximise the expectation value of her information, which can be organised into the following formula:[19, ch. IV.B.1]

$$\begin{aligned} I_E &= \max_{Eve} \{Y_0 \cdot I_{E,0} + Y_1 \cdot I_{E,1} + (1 - Y_0 - Y_1) \cdot I_{E,\geq 2}\} \\ &= \max_{Eve} \{Y_0 \cdot 0 + Y_1 \cdot h(\varepsilon_1) + (1 - Y_0 - Y_1) \cdot 1\} \\ &= 1 + \max_{Eve} \{-Y_0 - Y_1 \cdot (1 - h(\varepsilon_1))\} \\ &= 1 - \min_{Eve} \{Y_0 + Y_1 \cdot (1 - h(\varepsilon_1))\} \end{aligned} \tag{2.6}$$

where the "Eve" under max indicates that max is a maximisation over the parameters that Eve controls. $I_{E,n}$ is the information Eve gets on the secret key when Alice sends n photons to Bob and Y_n is the probability, expressed as a fraction, that Bob detects a signal when Alice emitted n photons in that signal. Alice and Bob have no way of knowing ε_1, however they can put an upper bound on ε_1 by assuming that all of their errors result from Eves attempt at eavesdropping. This yield the following bound on ε_1: $\varepsilon_1 \leq \frac{Q}{Y_1}$. Eve can however minimise Y_0 without consequence.

So even if Eve optimises her strategy she gains no more information than $I_E = 1 - Y_1 \cdot (1 - h(\varepsilon_1))$. □

Using equation 2.4 and equation 2.5 we now see that:

$$\begin{aligned} K &= R\left(1 - \left(1 - Y_1\left(1 - h\left(\frac{Q}{Y_1}\right)\right)\right) - \text{leak}(Q_{EC})\right) \\ &= R\left(Y_1\left(1 - h\left(\frac{Q}{Y_1}\right)\right) - \text{leak}(Q_{EC})\right) \end{aligned} \tag{2.7}$$

This explains the advantage of having a deterministic source of single photons for QKD. While it is possible to make QKD work with a source of single photons that sometimes produce more photons or no photons, but only the single photons count toward the final key rate, while multi-photon states still count toward a grater error rate.

[7] It is often assumed in QKD, as part of security proofs, that during error correction Bob correct his key to make it match that of Alice.[12]

Summary

In this chapter we covered the following:

- The problem that QKD tries to solve is to generate a string shared by two parties, Alice and Bob, and that having a method for creating such a string enables secure communication through the one-time-pad.
- That the advantage of having a quantum channel is that any attempt at reading the message therein results in detectable errors.
- The BB84 protocol which uses a quantum channel, a classical channel and an initial key to make a key larger than the initial key.
- The distinction between raw and final key rate, and the amount of secret key that can be distilled from the raw key given the parameters that can be obtained using the BB84 protocol.

Chapter 3

Quadrupler

A SUB-PROJECT that started early on in the project was making a quadrupler which is a device that takes in a series of laser pulses with a given repetition rate and emit a series of laser pulses with four times that repetition rate. The repetition rate of a laser pulse is the number of pulses emitted by a laser over a time interval, usually expressed in Hz. This project did not start from scratch, but is an ongoing project in the group that had been put on hold.

The reason that such a device would be useful for QKD is that such laser pulses are used for stimulating a quantum dot as part of the deterministic generation of single photons. Recall that in BB84 the key rate is dependent on the rate of single photon emissions, so being able to make four times as many single photons would mean getting four times the key rate; all else being equal.

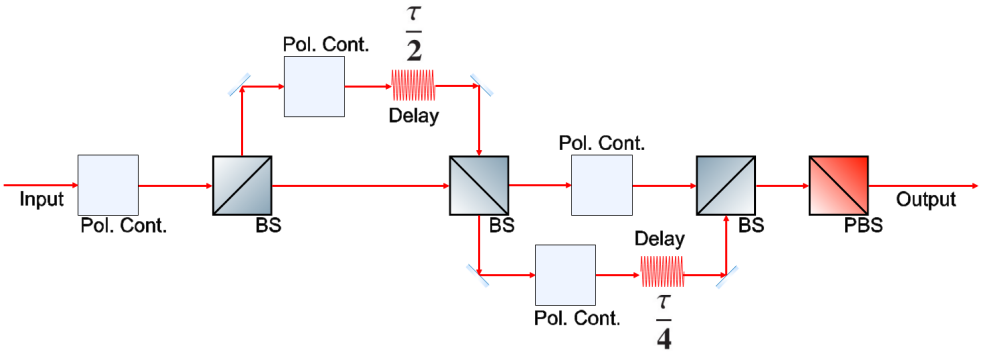
The laser in the lab is called MIRA and has a repetition rate of 72.6MHz.

The goal of this project was to improve the existing quadrupler. Preferably without purchasing new components.

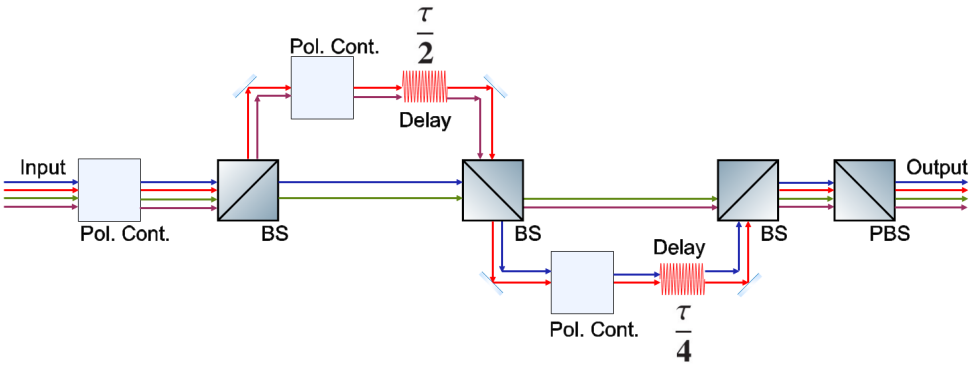
Initial concept

The quadrupler is depicted in figure 3.1a. It works as follows:

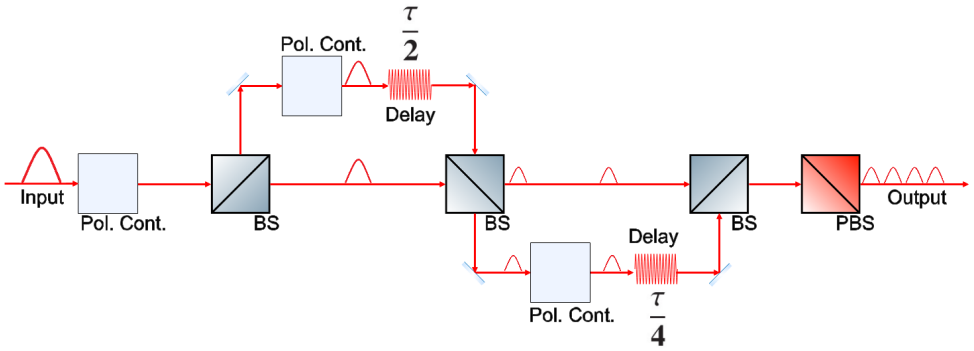
The light reflects of the first beamsplitter and half of it gets a delay equivalent to half the time between pulses, also referred to as the temporal spacing. The other half passes to the next beamsplitter without delay. The two pulses then hit the second beamsplitter, at different times due to the delay, and both get divided in two such that half the pulses get a delay of one quarter of the original temporal spacing. The last beamsplitter is there to bring all the pulses into a single fibre. This last step necessitates a 50% loss in power for all pulses. The purposes of the PBS and the polarisation controllers is covered later in this chapter.



(a) A schematic of the quadrupler.



(b) A schematic for a hypothetical scenario with 4 linearly independent beamsplitters.



(c) A schematic of the quadrupler with some pulses added to illustrate how it works.

Figure 3.1: Different schematic for the quadrupler. While the icons are reminiscent of a free space setup only the PBS is actually in free space, the rest is in fibre. BS: 50/50 beamsplitter; PBS: polarising beamsplitter; Pol. Cont.: polarisation controller; Delay: A delay with a polarisation controller. The fraction next to the delay indicate the amount of delay, with τ being the inverse of the repetition rate of the laser also known as the temporal spacing.

Another way to explain the setup is that there are four paths through the system as depicted in figure 3.1b. One where the light goes straight through the quadruple without any delay, one where the pulse goes through one delay where it gets delayed by $\frac{\tau \cdot 2}{4}$, one where the pulse goes through the other delay and gets delayed by $\frac{\tau}{4}$ and finally the path that goes through both delays and gets delayed by $\frac{\tau \cdot 3}{4}$. With tau being the inverse of the repetition rate of the input laser.

There is a problem with implementing the above description in the real world. Namely that real world components have losses and that these losses vary between different components and different ports of the same component. Add to this that 50/50 beamsplitters do not split light exactly 50/50 and it is easy to see that the pulses emitted by the quadrupler would not be of equal amplitude.

This is a problem for achieving higher key rates as it makes it impossible to have all the pulses be pi-pulses simultaneously. That is all the pulses will not excite the quantum dot completely. There are a few approaches to solving this problem. First there is the possibility of simply reducing the losses, secondly the losses could be distributed evenly among the different paths. Failing that one must introduce artificial losses on the paths with smaller losses.

To introduce artificial losses on the paths with smaller intrinsic losses both of the delays and the input port have a polarisation controller. The polarisation controllers enable the polarisation of a pulse to be adjusted according to what path the pulse took through the system. The PBS at the end then sends through only the fraction of the polarisation that aligns with its polarisation axis. This then enables path dependent losses and by extension losses on the path with the smaller losses.

Since there is a total of three polarisation controllers we have three degrees of freedom which is enough to ensure that all the pulses have the same amplitude.

To see this let's first imagine that we had four polarisation controllers in the quadrupler and that all were linearly independent of each other, as depicted in figure 3.1a. If this was the case we could effectively adjust the polarisation of each path separately. Therefore we could introduce separate losses on each path. This would enable the attenuation of all but the weakest pulse such that all pulses would have equal amplitude *and* have that amplitude be as great as possible using only reductions in the amplitudes.

However the quadrupler setup has only 3 independent variables so we cannot set all amplitudes independently with this method, so we do not have complete control of the outcome. Since we lack one degree of freedom and since the desired output can be characterised by a single number, the pulse amplitude after the quadrupler, this must be the number that we cannot control.

Quadrupled

A term that comes up later is the term quadrupled. This term is the referrers to an object having been affected by a quadrupler. A quadrupled laser pulse is one that has passed through a quadrupler, and a quadrupled QD is a QD that is being stimulated by a quadrupled laser beam and so on.

The first part of the project

The main problem when I took over the project was that there were great losses and the pulses coming out of the quadrupler were quite uneven in amplitude. The first problem is a product of both the faults in the components and the artificial losses that have been introduced to mitigate this problem. The components already available were of high quality, and while it is always possible to procure products of slightly higher quality at ever higher costs it is not clear that the benefit would justify the costs.

It was decided that the best approach to remedy this was to make a digital model of the quadrupler such that it would be possible to simulate all permutations of the various components. Then the components could be arranged such that the losses were distributed evenly among the different paths. That rearranging the components would decrease the variations in amplitude can be explained with the following example:

Imagine one had three beamsplitters that all came from the same manufacturer and were manufactured according to the same specifications but differ in their losses and imperfections. These components could be arranged into a quadrupler such that a single path had all the beamsplitter ports with highest losses and both of the imperfect delays and a different path had all the least lossy outputs and potentially none of the delays. This would obviously mean that one of the pulses would leave the quadrupler with much greater amplitude than the other. One could then imagine rearranging the components such that the paths through the different path had equal losses, or as close to equal as could be achieved with the components available.

To see how likely success is it useful to calculate the number of unique permutations that can be made from the components. There are 3 beamsplitters and three positions for a beamsplitter in the quadrupler which which leaves $3 \cdot 2 \cdot 1 = 6$ permutations of the beamsplitters. Furthermore we can flip the outputs of the beamsplitter, as in changing which one goes to the delay. The same can be done for the inputs. This gives $2^2 = 4$ permutations per beamsplitter. Lastly we can interchange the 2 delays. This gives a total of $6 \cdot 4^3 \cdot 2 = 768$ outputs. So the chance that there is as something to be gained by this, assuming that the components had been assembled at random and that there is only one optimal solution, is $\frac{767}{768} \approx 1$.

The code

The code to make and simulate all of these different permutation is organised as follows:

- Since we are concerned with pulses of light in the real quadrupler, it makes sense to quantify the light as pulses. The **pulse class** has a time, an amplitude and some functions for performing arithmetic on these variables. The time is how much delay the pulse has undergone, and is useful as a sanity check.

- The **pulse list class** is a location in the circuit such as an output of a beamsplitter. It has some functions for applying arithmetic functions on all pulses on the list.

- The **positionlists class** is to pulselists what pulselists are to pulses. A positionlist can take pulselists and gather them in one place and enables arithmetic operations on all pulses in the pulselists in the postionlist. This is very useful for keeping track of multiple outputs of a component such as a beamsplitter.

- All components inherit the **component class** which gives them some functions that are necessary for all components. An example of such a function would be the flush function which sends light pulses in the output to the input of a different component.

- Every **specific component** has a run function that describes how the component handles input pulses.

- The **circuit class** keeps track of the components and has a function that identifies the component with the earliest pulse, i.e. the one with the smallest time (see above), and triggers the run function of that component. It has another function that calls that function until there are no more pulses in the input of any component, after which the simulation has ended.

This is not a complete description of the code but merely forms a basic understanding of how it is structured. The code is a general framework that can, with some minor edits, simulate any optical setup that uses pulsed light.

To simulate all permutations of our setup it is necessary to be able to cycle through them. This is taken care of by the simulate all function which is a function that simulates all the permutations of the setup and is depicted in figure 3.2. The simulate all function first defines the components with their transmission coefficients, which was measured as explained below. It then runs the function allcombis which recursively finds all ways to combine the lists of components given to it. The simulateall function then runs a for loop

```
DEFINE SimulateAll()
    |
    |   BS1 = Beamsplitter("BS1",kwargs)
    |   BS2 = Beamsplitter("BS2",kwargs)
    |   BS3 = Beamsplitter("BS3",kwargs)
    |   D1 = Delay("D1",kwargs)
    |   D2 = Delay("D2",kwargs)
    |
    |   combis=allcombis([BS1,BS2,BS3][D1,D2])
    |
    |   FOR c IN combis
    |       |
    |       |   FOR n IN range (0 to 63)
    |       |       |
    |       |       |   flipper(n)
    |       |       |   Circuit = make_circuit(c)
    |       |       |   score = Circuit.run()
    |       |       |   add score to score_list
    |       |       |
    |       |   END FOR LOOP
    |       |
    |   END FOR LOOP
    |
    |   score_list.sort()
    |   RETURN score_list
    |
END DEFINE
```

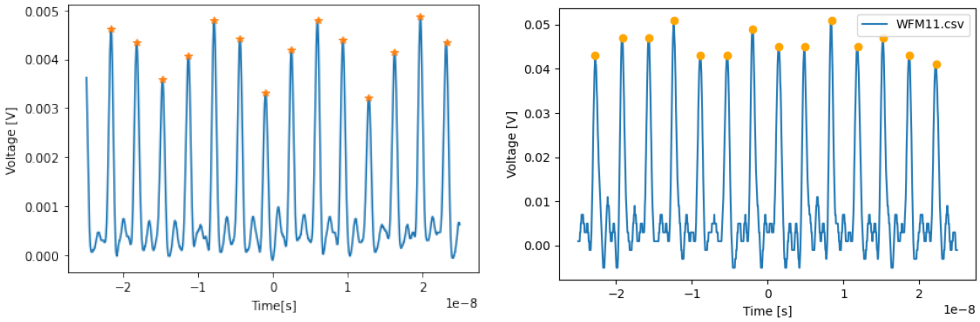
Figure 3.2: Pseudo-code for the simulate all function. The terms in purple are intrinsic command like if or while. The functions are blue and orange depending for user-defined functions and not user-defined functions.

over that list. In that for loop there is another for loop over the integers from, and including, 0 to, and including, 63. This number is called n. For each number n a function called flipper is called, which flips the input and output of the beamsplitters depending on the number n.[1] The function then makes a circuit based on the current place in both for loops and calculates a score with the following formula $I_{max} - I_{min}$. Once all simulation is complete the simulate all functions orders them by their score before returning them to the function call.

Getting the data for the simulation

In order to simulate the setup it is first necessary to characterise the components. The measurements were made by sending a continuous wave (CW) laser into the quadrupler, and then measure the power at the output of each

[1] This is done in such a way as to systematically go over every permutation of flipping and not flipping the inputs and outputs of the beamsplitters.



(a) (b)

Figure 3.3: a) The output of the quadrupler before I joined the project. b) The output of the quadrupler after the simulation and optimisation. The important difference between the figures is that the pulses could be much more even in height after the adjustments.

component. This necessitated disconnecting part of the system such that only one path through the system was open at any given point in time, otherwise any two beamsplitters would form a Mach-Zender interferometer.

During the measurements a lot of care was taken in order to ensure that no dust was on the fibres during measurements. This was done because the improvements are intended to be permanent and it is would be nearly impossible to record where any dust was on a fibre and even harder to get it back into that same position if it ever got moved.

The results of the simulation

It was revealed by the simulation that there was a more optimal way to arrange the components of the simulation. Specifically by changing the positions of two of the beamsplitters, and flipping their outputs.

These changes were then implemented.

The first tests

The quadrupler was then tested. In order to see how great the improvements were the tests were compared to some preexisting data on the quadrupler.

The test consisted of sending pulsed light through the entire system and measuring the outgoing light with a fast photo detector. The photo detector was connected to an oscilloscope and thus the individual pulses of light could be recorded and stored digitally on a USB. The data can be seen in figure 3.3. The pulses are more even in the newer figure. This indicates that it is easier to adjust the quadrupler such that the amplitude of the pulses are even.

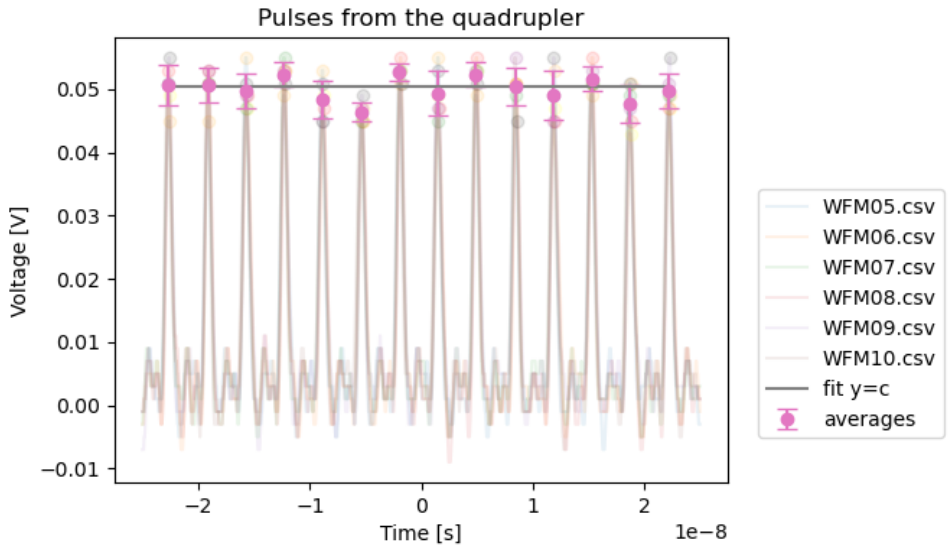


Figure 3.4: The waveforms of multiple waveforms layered on top of each other. The error bars are one $\sigma = 68\%$ confidence intervals and have been made assuming a Gaussian distribution.

Immediately after the first data set was saved five other were also saved without adjusting any of the polarisation paddles for a total of six data sets. This enable us to estimate the uncertainty of the amplitude measurement by finding the standard deviation. To gain useful information it is necessary to ensure that only pulses that have taken the same path through the quadrupler are compared since the pulses that travel through different paths might have a systematic error that we are interested in finding. To accomplish this the oscilloscope was set to "trigger" on a non-quadrupled pulse from MIRA. MIRA being the laser that the quadrupler was quadrupling in this experiment.

The resulting graph is shown in figure 3.4. The fact that the fit to a constant value lies within all but a third of the 68% confidence intervals indicates that we cannot be rule out the possibility that all of the pulses have the same amplitude. This is further indicated by the fact that there is no consistent pattern for pulses that have taken the same path relative to all others.

EXAMPLE: The first, fifth, ninth and thirteenth pulse all took the same path through the system. This is obvious from the fact that every fourth pulse must have taken the same path through the quadrupler. The first pulse in figure 3.4 is on the fitted line, the fifth pulse is below the line, the ninth pulse is above the line and the thirteenth pulse is below the line. This is of course less rigorous than the confidence intervals from before but it does provide a sanity check for the idea that the pulses are of equal amplitude.

A similar sanity check can be done for any other set of pulses that took the same path through the quadrupler.

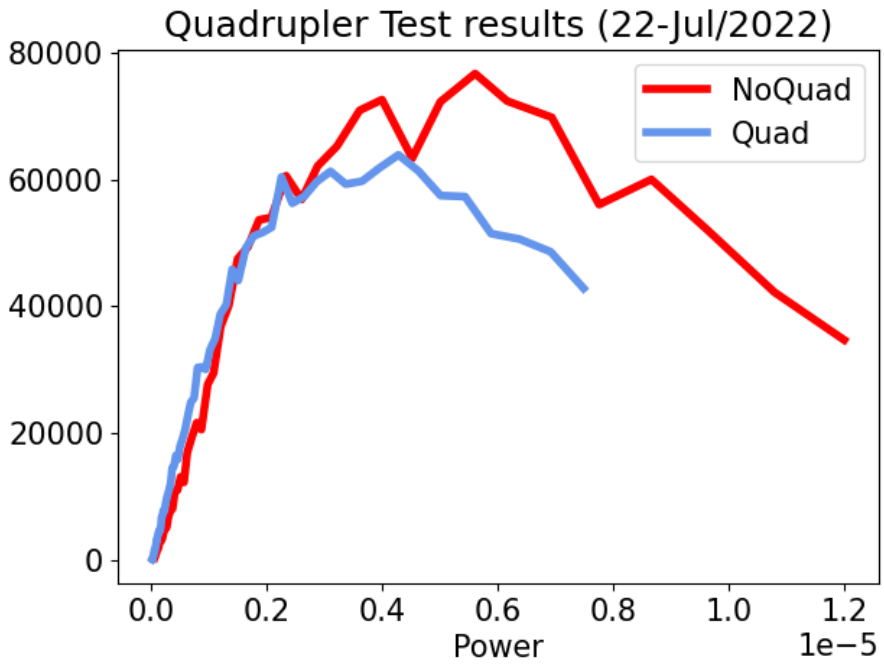


Figure 3.5: This figure depict the number of photons emitted by a QD, which is being pumped by a pulsed laser, as measured by an SNSPD. The line labelled "NoQuad" depicts the case where the pulsed light directly excites the QD, where as the line labelled "Quad" depicts the scenario where the pulsed light is passed through the quadrupler before being sent to the QD. The Power for the plot titled "Quad" was divided by four before plotting the figure in order to account for the fact that power was measured before the light entered the quadrupler. The unit of power is watts and the y-axis is in arbitrary units.

Subsequent tests

The next step was to see if the quadrupler actually enables us to excite a quantum dot four times as often. To do this the output of the quadrupler was connected to a polarisation maintaining fibre. This fibre then guided the light to a quantum dot. The emmited photons were collected and sent to a superconducting-nanowire-single-photon-detector (SNSPD). A computer then recorded the photon count rate from the output of the detector.

This was done multiple times with many variations. The data was plottet as photon counts as a function of power. Figure 3.5 shows the photon counts on the SNSPDs as a function of laser power before the quadrupler. What is notable is that the quadrupler actually seems to *decrease* the number of photons generated by the single photon source. There may be a couple of explanations for this:

1. **The deadtime** of the QD is the time after its excitation when it cannot be re-excited because it has not decayed into the groundstate. This results in subsequent pulses being unable to reexcite the quantum dot if they arrive within this time window.

2. **Alignment drift** is the tendency of the laser to move. The setup requires that the laser is pointed toward the QD and this adjustment has to be made manually. Tt was necessary to readjust the laser throughout the measurement. If the laser was worse aligned in subsequent measurements this could obviously explain some of the gap between the two measurements.

3. **Detector darktime** is the time after a measurement when the detector is unable to detect another photon. This obviously can lower the photon count rate when we have the photons arrive with shorter time separation.

4. **The uneven amplitudes** of the pulses exiting the quadrupler will also result in lower detection rates. To see how this is the case one could imagine one of the amplitudes being perfectly tuned for exciting the quantum dot. If the pulses don't have the same amplitude the other three pulses must necessarily be slightly off pi-pulse, otherwise they *would* have the same amplitude. Therefore one cannot excite the QD fully with all the laser pulses, if they are of unequal amplitude. This would result in less than a four fold increase in single photon rate. The unevenness of amplitude is a result of both the instability of the polarisation of the Quadrupler (see below) and also the fact that the polarisation was adjusted by hand.

Neither the first nor the last two reasons could ever individually explain the quadrupler being less effective than no quadrupler. However in combination they can.

Further an intensity interferometry measurement, also called a Hanbury-Brown-Twiss experiment, was performed. The setup is depicted in figure 3.7. The way an intensity interferometry measurement works is that the light from the light source being examined enters into a 50/50 beamsplitter. The two outputs of the beamsplitter then sent to two detectors. A computer notes the timing of the detection events and correlate them. Correlating them in this case means that for every pair of detections, with one detection from each detector, the difference in the timing of the events $\Delta t = t_1 - t_2$ is recorded. Then a histogram is made showing the number of detection pairs with a given Δt. This data one can find the correlation at $\Delta t = 0$ called $g_2(0)$ or simply g_2.

This measurement resulted in fig. 3.6. From this plot it is very clear that the quadrupler does succeed at shortening the time between pulses. However it is also clear from the graph that the quadrupled quantum dot does not

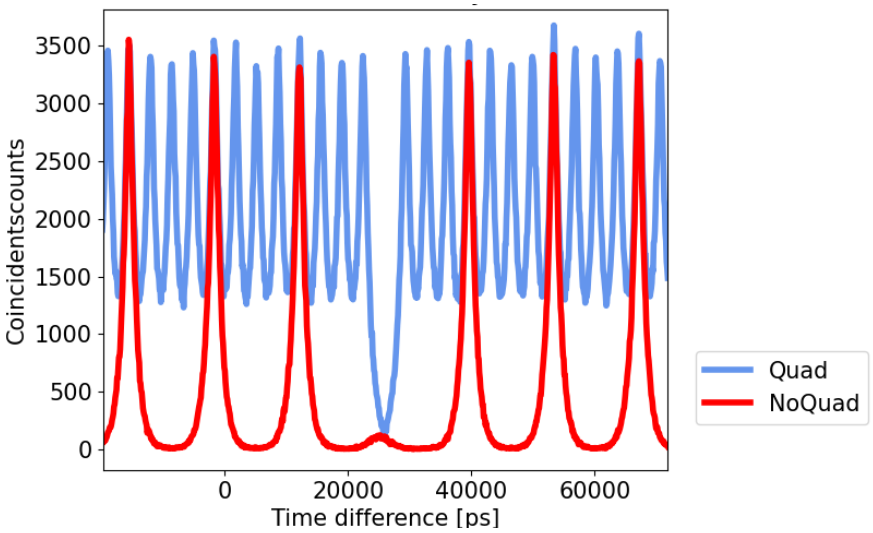


Figure 3.6: The g_2 measurement for the single photon source, with and without a quadrupler labelled Quad and NoQuad respectively. The plots do not have their minima at 0; this is explained by the two arms of the Hanbury-Brow-Twiss setup being of unequal length.

manage to undergo a full decay. To see this one need only remember that the rate of photon emission is proportional to the probability of finding the QD in the excited state. This indicates that the finite lifetime of the excited states of the quantum dot is a limiting factor in the setup; as expected from the deadtime explanation. Furthermore the peak height and $g_2(0)$ of the two measurements are in agreement with each other which is also expected from the deadtime explanation. So it is likely a main contributor to the issue.

Remaining problems with the quadrupler

Whenever a piece of equipment get tested for the first time there is bound to be some practical issues to rear their heads. Here are the ones encountered with the quadrupler.

Polarisation stability

It was noted during the tests that the polarisation of the quadrupler was very unstable and there was a notable drift over the span of 15 minutes. A smaller contribution to this is probably that the setup is not covered in any way and therefore some convection is to be expected. The greater part is due to thermal heating since the quadrupler never reached thermal equilibrium as the laser only went through the quadrupler at sporadic intervals. This was necessary to get data on the effects of exciting the QD directly with the laser. Reaching

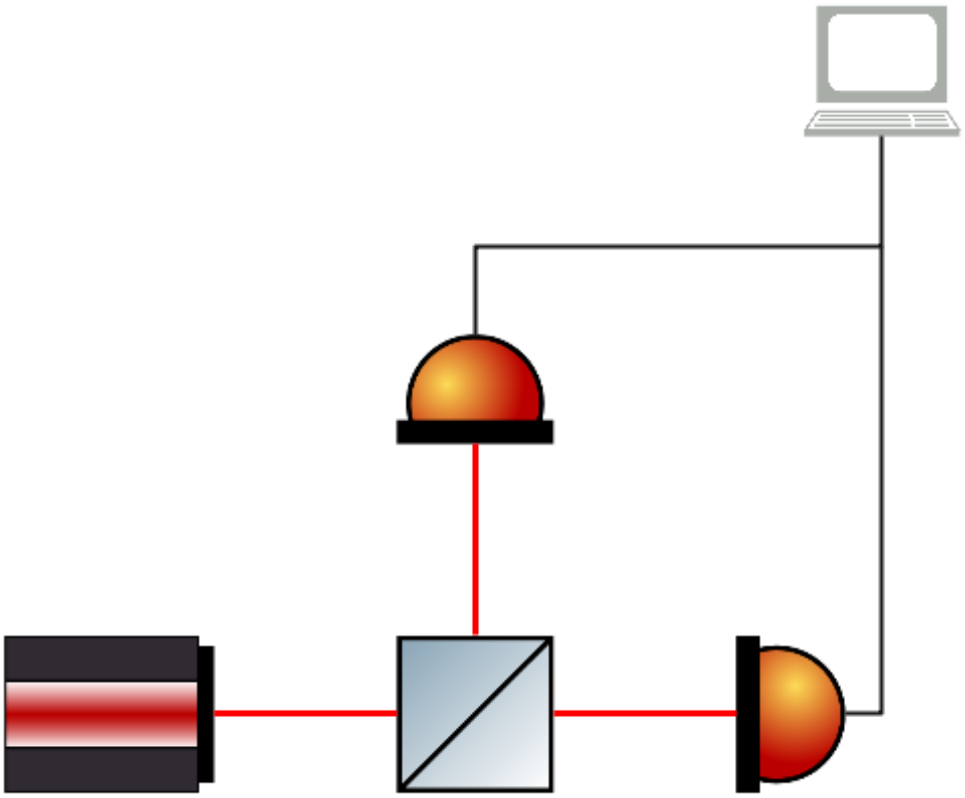


Figure 3.7: A depiction of an intensity interferometry measurement.

thermal equilibrium is important since the polarisation of the light is sensitive to thermal effects.

Another potential source of instability is that the fibre that carried the light from the laser to the quadrupler, or directly to the dot, was physically moved. This almost certainly caused slight twisting of the fibre. Since the fibre was a single-mode fibre this has almost certainly changed the input polarisation and as consequence the output polarisation. However even when the fibre was not moved the polarisation still drifted; so this is not the only effect at play.

This issue could possibly be resolved by adding some kind of thermal isolation and letting the system reach thermal equilibrium.

Variations in efficiency

The energy efficiency of the quadrupler is of paramount concern as the goal of making the quadrupler is to repeatedly stimulate a QD in a deterministic manner. With high losses this would be impossible to accomplish since the input power is limited by the maximum power tolerance of the input fibre of the quadrupler.

The long term average power of the light coming out of the quadrupler never broke 20% of the input power and never got below 10%. During the test it was at times necessary to inject power into the quadrupler beyond what they could maintain in the long term according to their specifications. Theoretically one could only reach 50% with the current design as the last 50/50 beamsplitter sends half of the light to a termination (see figure 3.1) but even that is likely far beyond what is practically possible.

To improve the efficiency an additional polarisation controller could be inserted as described in the section titled initial concept.

Summary

In this chapter we covered the following:

- A quadrupler takes in a stream of laser-pulses and outputs one with four fold the repetition rate. It works by using three beamsplitters and two delays.
- By rearranging the components we can improve the quadrupler. A code was made to find the optimal setup.
- Additionally the code can, with minor edits, be repurposed to simulate any optical system with pulsed lasers.
- The system can excite a quantum dot four times as fast, but this does not result in four times the single photons.
- Questions of polarisation stability and efficiency still remains to be solved. These could possibly be resolved by adding thermal isolation anf an additional polarisation controller.

Chapter 4

The field trial

This section will be somewhat different from the other chapters in this work. This is mainly due to the fact that I did not make a lot of decisions as I joined late in the project. This resulted in me mostly doing tasks that do not involve a lot of decision making. An example would be constructing Bob, which I did at DTU. This is a task that you want done correctly and therefore by someone who understands the setup, but the task does not involve making decisions about the setup as all the components were already purchased.

Therefore I cannot spend this chapter discussing any real choices that I made, but I can spend time describing the work and justify the choices made by others.

Purpose of the work

The purpose of the work was to make a field trial of BB84 QKD with a single photon source. This was done over an already existing fibre in the metropolitan area of Copenhagen. This is an important milestone for QKD with single photon sources as it proves that we can make polarisation encoding work even over long distances in environments with a lot of potential noise sources. The fibre stretched 18km from NBI (Alice) to DTU (Bob) resulting in an attenuation of 9.6dB. Another fibre sent a signal to keep the two setups synchronised.

As always the source was emitting photons at a rate of 72.6 Mhz and underwent frequency conversion. The frequency conversion was to 1545 nm instead of 1550 as is usually the case. This was because there were some cross talk with some other fibre due to the fibre being preexisting. This might initially seem like it would detract from the experiment but really the success shows that single photon QKD is fairly robust.

The prep-work

We build a box around Alice. The box was there to prevent convection from being too much of a problem.

We had to set up this very long fibre which lead from the lab where the photons were encoded to an old telephone central in the basement of NBI. We encountered some problems. There was a lot of noise on the channel from some other source so we moved the wavelength.

Results

We achieved stable operation in terms of 2 kbits/s.

Chapter 5

Measurement device independent QKD

There are several so-called security loopholes that plague QKD.[19, ch. III.B.4] A security loophole is a way that Eve can get more information that expected, usually by breaking restriction 1, that Eve cannot peer in to the laboratories of Alice and Bob

The one most commonly referenced is the Trojan Horse attack. The attack is performed by shining a laser into Alice's setup. The light then gets reflected internally in the system and returns, carrying information about the polarisation Alice is currently encoding. This extra information can then be used by Eve when choosing her measurement basis.[19, ch. III.B.4]

These security loopholes are generally easy to address individually for but can be both difficult to find and the corrections potentially lead to new security loopholes. So it is useful to have a method that would close an entire category of such loophole.[19, ch. III.B.4]

Measurement Device Independent Quantum Key Distribution (MDIQKD) is such a method. This is not an add-on to the BB84 protocol, but is instead a replacement that closes all detector side loopholes.[14]

It was decided that the group should do research in this direction and so I was assigned to procure components needed to build an MDIQKD setup.

The MDIQKD protocol

In the MDIQKD protocol Alice and Bob independently prepare a qubit in one of two orthogonal basses; they then send them to an untrusted third party, called Charlie. An "untrusted third party" means that Alice and Bob may suspect that Charlie is in league with Eve. In fact, in security proofs it is assumed that Charlie is Eve.[6][26] Charlie, after receiving these qubits from Alice and Bob performs a bell-state measurement and publishes the result publicly. This then reveals the correlation between Alice's and Bobs bit. Bob

then corrects his bit value. Alice and Bob then announce what basis they were using and discard the bits for which they used different basses.[14] They then perform postprocessing almost as in BB84. The only difference is that the maximal secret key rate is no longer given by eq. 2.7 but as:[14]

$$K = Y_{HV}^{1,1}(1 - h(e_{DA}^{1,1})) - \text{leak}(Q_{HV}) \tag{5.1}$$

where $Y_{\alpha}^{a,b}$ is the single photon rate when Alice and Bob send a and b photons in the basis α and $\text{leak}(Q_{HV})$ is the information leaked in correcting the errors in the HV basis. $e_{DA}^{1,1}$ is the probability of error when Alice and Bob send one photon each and choose the DA basis. Given that Alice and Bob have no way of knowing what errors occurred when they send 1 photon as opposed to two photons it will have to be estimated as $e_{DA}^{1,1} \leq \frac{Q_{DA}}{Y_{DA}^{1,1}}$. The reason for the asymmetry between the HV and DA bases is that HV is used as the raw key and DA is used only for estimating the amount of Eves dropping.[12]

The advantage of MDIQKD is that even when Eve knows all about the correlation between Alice and Bobs bit she does not know their values. Say the $|\psi^-\rangle$ was measured by Eve and that Alice and Bob both announce that they were using the diagonal basis. Then Eve would know that Alice and Bob sent the same bit value (see below), but she has no means of finding out what that bit value is.

The correlations

Before seeing how the bell-state measurements reveal the correlations it is worth noting that only the $|\psi^+\rangle$ and $|\psi^-\rangle$ states can be distinguished using only nonlinear components.[14] Since single photon non-linearity's are still a matter of cutting edge research[24] two bell-states have to do. Also, since we will still be working with polarisation encoding of the qubits we might as well use that notation immediately.

The two bell-states that we can measure are:

$$|\psi^+\rangle = \frac{1}{\sqrt{2}}(|VH\rangle + |HV\rangle) \tag{5.2}$$

$$|\psi^-\rangle = \frac{1}{\sqrt{2}}(|VH\rangle - |HV\rangle) \tag{5.3}$$

Where V is vertical polarisation and H is horizontal polarisation.

If Alice and Bob both send $|V\rangle$ then neither of the bell-states will be measured, since

$$\langle\psi^+|VV\rangle = \langle\psi^-|VV\rangle = \langle\psi^+|HH\rangle = \langle\psi^-|HH\rangle = 0. \tag{5.4}$$

Same result if Alice and Bob both send $|H\rangle$ states. If on the other hand they send different bit values they will measure one of the bell-states as:

$$\langle\psi^+|VH\rangle = -\langle\psi^-|VH\rangle = \frac{1}{\sqrt{2}}, \tag{5.5}$$

meaning that there is a 50% chance of measuring either state. In either case the original bits are anti-correlated so Bob must flip his bit for the protocol to work.[14]

If Alice and Bob both choose to send their bits in the diagonal basis the projection onto the bell-states is:

$$\begin{aligned}\langle DD|\psi^+\rangle &= \frac{1}{2}\left(\langle V| + \langle H|\right)\left(\langle V| + \langle H|\right)|\psi^+\rangle \\ &= \frac{1}{2}\left(\langle VV| + \langle VH| + \langle HV| + \langle HH|\right)|\psi^+\rangle \\ &= \frac{1}{2}\cdot\left(\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}\right) \\ &= \frac{1}{\sqrt{2}}\end{aligned} \tag{5.6}$$

$$\begin{aligned}\langle DD|\psi^-\rangle &= \frac{1}{2}\left(\langle V| + \langle H|\right)\left(\langle V| + \langle H|\right)|\psi^-\rangle \\ &= \frac{1}{2}\left(\langle VV| + \langle VH| + \langle HV| + \langle HH|\right)|\psi^-\rangle \\ &= \frac{1}{2}\cdot\left(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right) \\ &= 0\end{aligned} \tag{5.7}$$

$$\begin{aligned}\langle DA|\psi^+\rangle &= \frac{1}{2}\left(\langle V| + \langle H|\right)\left(\langle V| - \langle H|\right)|\psi^+\rangle \\ &= \frac{1}{2}\left(\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right) = 0 \\ \langle DA|\psi^-\rangle &= \frac{1}{2}\left(\langle V| + \langle H|\right)\left(\langle V| - \langle H|\right)|\psi^-\rangle \\ &= \frac{1}{2}\left(\frac{1}{\sqrt{2}} - \frac{-1}{\sqrt{2}}\right) = \frac{1}{\sqrt{2}}\end{aligned} \tag{5.8}$$

$$\begin{aligned}\langle AA|\psi^+\rangle &= \frac{1}{2}\left(\langle V| - \langle H|\right)\left(\langle V| - \langle H|\right)|\psi^+\rangle \\ &= \frac{1}{2}\left(-\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right) = -\frac{1}{\sqrt{2}} \\ \langle AA|\psi^-\rangle &= \frac{1}{2}\left(\langle V| - \langle H|\right)\left(\langle V| - \langle H|\right)|\psi^-\rangle \\ &= \frac{1}{2}\left(-\frac{1}{\sqrt{2}} - \frac{-1}{\sqrt{2}}\right) = 0.\end{aligned} \tag{5.9}$$

With the $|AD\rangle$ and $|DA\rangle$ cases are equivalent.

From this it can be seen that, when Charlie measures a $|\psi^+\rangle$ and both Alice and Bob sent photons in the diagonal basis that state Bob should not

flip his bit. Also, when the $|\psi^-\rangle$ state is measured Bob should flip his bit.[14] Either way neither Eve nor Charlie can gain any information on the state of Alice or Bob.

Experimental bell-state measurement

The setup for a bell-state measurement is given in figure 5.1 and is able to tell a $|\psi^+\rangle$ state from a $|\psi^-\rangle$. The device has multiple detectors therefore the term "event" no longer referrers to a single photon being detected, but to any number of photons being detected when we would expect a photon from Alice and from Bob.

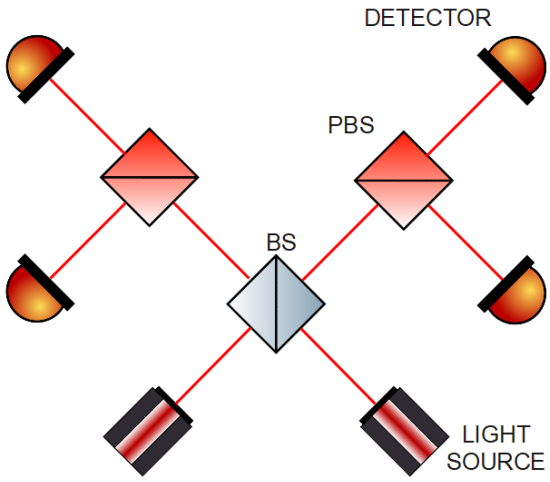


Figure 5.1: The setup for a bell-state measurement. PBS: Polarising BeamSplitter; BS: BeamSplitter; DETECTOR: single photon detector; Light source: An arbitrary source of light

If the photons form a $|\psi^+\rangle$ state the Hong-Ou-Mandel effect ensures that the photons incident on the beamsplitter go into the same path and the PBS ensures that two different detectors go off. If a $|\psi^-\rangle$ state is incident on the beamsplitter the photons go to different paths of the detector, but still to opposite detectors. So one can tell the states apart on whether the detectors went off on the same arm or on different arms.

To see this we first note that a beamsplitter transforms the creation operators according to the following linear matrix $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}$ while also changing basis from the $\begin{pmatrix} a \\ b \end{pmatrix}$ to the $\begin{pmatrix} c \\ d \end{pmatrix}$ basis where a and b denotes the input ports and c and d denotes the output ports.[10] From this it follows that the state transform according to:

$$\begin{aligned}
|\psi^\pm\rangle =& \frac{1}{\sqrt{2}}(a_V^\dagger b_H^\dagger \pm a_H^\dagger b_V^\dagger)\,|\varnothing\rangle \\
\rightarrow& \frac{1}{2\sqrt{2}}\left((c_V^\dagger + id_V^\dagger)(ic_H^\dagger + d_H^\dagger) \pm (c_H^\dagger + id_H^\dagger)(ic_V^\dagger + d_V^\dagger)\right)|\varnothing\rangle \\
=& \frac{1}{2\sqrt{2}}\left((ic_V^\dagger c_H^\dagger + id_V^\dagger d_H^\dagger - c_H^\dagger d_V^\dagger + c_V^\dagger d_H^\dagger) \pm (ic_H^\dagger c_V^\dagger + ic_H^\dagger c_V^\dagger - c_V^\dagger d_H^\dagger + c_H^\dagger d_V^\dagger)\right)|\varnothing\rangle
\end{aligned} \tag{5.10}$$

Alice								
Alice's bit	0	1	0	1	0	0	0	1
Alice's basis	×	×	+	+	×	+	+	×
Alice's photon	D	A	H	V	D	H	H	A
Bob								
Bobs bit	0	0	1	0	1	0	0	1
Bobs basis	×	+	×	+	×	×	+	×
Bobs photon	D	H	A	H	A	D	H	A
Charlie								
Charlies possible measurement	$\psi^+/\emptyset$	$\psi^\pm/\emptyset$	$\psi^\pm/\emptyset$	$\psi^\pm/\emptyset$	$\psi^-/\emptyset$	$\psi^\pm/\emptyset$	$\emptyset$	$\psi^+/\emptyset$
Charlies actual measurement	ψ^+	$\emptyset$	ψ^+	ψ^-	ψ^-	$\emptyset$	$\emptyset$	ψ^+
postprocessing								
Alice's key after sifting	0			1	0			1
Bobs key after sifting	0			0	1			1
Should Bob correct this bit?	No			Yes	Yes			No
Bobs key after corrections	0			1	0			1
Alice's key after privacy amplification	1			0	0			1
Bobs key after privacy amplification	1			0	0			1

Table 5.1: A table showing how the MDIQKD protocol works with perfect components. The steps of error estimation and error correction have been left out as there are no errors. In the "Charlies possible measurement" row is indicated all of Charlies possible measurement outcomes with $\psi^\pm$ indicating the possibility of both the psi^+ and psi^- states. $\emptyset$ indicates presence of the $\phi^\pm$ states which we cannot detect.

where $|\emptyset\rangle$ is the vacuum state and $i_J^\dagger$ is the creation operator for a photon in channel i and with polarisation J. Following from eq. 5.10 the states tranform according to:

$$|\psi^+\rangle \rightarrow \frac{1}{2\sqrt{2}}(2ic_V^\dagger c_H^\dagger + 2id_V^\dagger d_H^\dagger)\,|\emptyset\rangle = \frac{i}{\sqrt{2}}(c_V^\dagger c_H^\dagger + d_V^\dagger d_H^\dagger)\,|\emptyset\rangle \tag{5.11}$$

$$|\psi^-\rangle \rightarrow \frac{1}{2\sqrt{2}}(2c_V^\dagger d_H^\dagger - 2c_H^\dagger d_V^\dagger)\,|\emptyset\rangle = \frac{1}{\sqrt{2}}(c_V^\dagger d_H^\dagger + c_H^\dagger d_V^\dagger)\,|\emptyset\rangle \tag{5.12}$$

from which it is clear to see that the photons of the ψ^+ are in the same path, but will be split apart by the PBS and that the ψ^- are already separated.

The device cannot tell the $|\phi^\pm\rangle$ states apart. In both cases it is just a single detector going off. In fact it is impossible to tell the difference between a $|\phi^\pm\rangle$ state and a single dark count. Therefore we will discard all $|\phi^\pm\rangle$ events. This does not compromise security as no detected state reveal any information to Eve.[14]

There is no way for two detectors on the same side to go off in a single event without a dark count or some other noise, so if such an event is detected it is discarded.

Simulating a MDIQKD setup

The assignment was to build a MDIQKD setup. This was an entirely new setup so the first thing to do was to procure the necessary components and so it was important to justify the choice of components. This was done by simulating how well the possible setups with the various components would work.

So to do that I made numerical model. The model follows the method laid out in [25]. The paper will be referred to as "the simulation paper" in this thesis.

The simulation paper, in short, uses a series conditional probabilities to calculate the probability of all possible events and then use those probabilities to calculate the terms in equation 5.1.

Since this project started as budget estimates I ended up making the entire model in Google Sheets of all things. Had I but known it would go this way, I would have rather written the budget in python. The model is depicted in figure 5.2.

The *blue area* takes in a list of components and outputs the price and performance in terms of losses and chance of the circuit flipping the polarisation. The data was taken from the specifications given by the various providers.

The *green area* calculates the chance of various numbers of photons making it to the detectors given the losses. It also calculates the odds of getting certain events given that certain numbers of photons get detected.

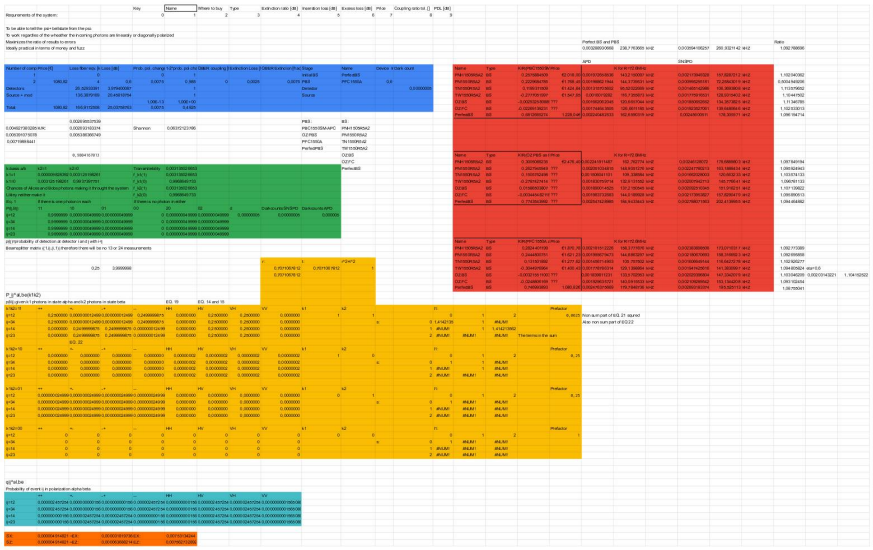


Figure 5.2: The numerical model. The purpose of the various coloured blocks are explained in the main text. The only thing that isn't in the figure is a lookup table for component specifications that are located far below this figure.

The *yellow area* calculates the odds of a certain output of the beamsplitter and multiplies it by the probability of an event given that output of the beamsplitter, calculated above. This results in the odds of an event given the input of the beamsplitter.

The *cyan area* determines the probabilities of events given the chosen polarisation's of Alice and Bob. This is done by summing over the many tables in the yellow area.

The *orange area* uses the cyan area to calculate the gain and error rates of the two polarisation bases. There is a single field in between the blue and green areas that calculate the key rate per qubit from these numbers.

In the *red area* are all the results given the choice of beamsplitter, polarising beamsplitter and choice of detector. Results in this context refers to both the performance of the device and the price of the components.

The maths of the model

Of the numbers in the component-specifications not all are useful. The insertion losses, for instance, are irrelevant as they are the ratio of the intensity output at one port of the beamsplitter and the intensity at the input port.

Effects	Description	Consequence
Excess losses	The losses in the component.[8]	lower key rate[25]
Polarisation dependent losses	Increases in extinction ratio for one polarisation.[8]	lower key rate[13]
Extinction ratio	The fraction of the light that changes polarisation in the component.[8]	higher QBER[7]
Coupling ratio tolerance	The maximum or expected error in the coupling ratio.[8]	higher QBER

Table 5.2: An overview of the effects of imperfections. QBER is the quantum bit error rate (see chapter 2). It isn't obvious that the polarisation dependent losses only result in lower gain. Technically they require a new security proof but that new security proof only results in a lower gain. An * means that this is extrapolation: We do not get a lower gain, because no photons are lost, but obviously this can change what event is measured.

This is useless as most of these "losses" are the light being split between two ports. The figure of relevance is the excess losses; that is the ratio between the input power and total output power.[8]

By examining specifications from multiple manufactures four figures of merit emerged, they are presented in table 5.2.

The different errors were dealt with as follows:

Excess losses are counted accounted for in the green area when calculating the chance of the photons making it to the detector. The simulation paper also deals with losses by counting them at the beginning justified by the linearity of all components.

In the paper discussing **polarisation dependent losses**[13] the lossier polarisation sets the ceiling for the gain. So the PDL is effectively applied to all polarisation's.

The extinction ratio applies a change in polarisation called misalignment errors. To account for these the simulation paper adds a term $E_d(1-2\tilde{E})$ to the error rate, where E_d is the misalignment and $\tilde{E}$ is the error without the alignment errors.

Coupling error tolerance is not accounted for in the simulation paper with no justification. Below I argue that the coupling error results in a QBER of approximately $\sigma_Q = 2^{\frac{2}{3}}\sigma_R^2$ where σ_Q is the additional QBER and σ_R is the coupling error. Given that some components have a coupling error tolerance of 5%[23] the resulting QBER would be $\sigma_Q = 2^{\frac{2}{3}} \cdot 0.05^2 = 0.7\%$. In our BB84 field trial the total QBER was 5%, so this is hardly negligible. With single photon sources this is easy to account

for by altering the equations in the yellow area to turn a factors of $\frac{1}{\sqrt{2}}$ into r and t as appropriate and calculating $r = \frac{1}{\sqrt{2}} + \sigma_r$ and $t = \sqrt{1 - r^2}$ with σ_r being the coupling error tolerance.

Coupling error

Taking into account the coupling error is the main way this model deviates from the outline in the simulation paper. To justify adding this in we first estimate the effect by considering a perfect beamsplitter and then perturbatively introduce a minor imperfection. We will then propagate that error through the system using the same technique as in error propagation.

To begin this we consider sending a $|\psi^{\pm}\rangle$ state in to a beamsplitter and we get the following output:

$$a_{\pm} |C_H D_V\rangle + b_{\pm} |C_V D_H\rangle + c_{\pm} |C_H C_V\rangle + d_{\pm} |D_H D_V\rangle , \tag{5.13}$$

where $a_{\pm}$,$b_{\pm}$,$c_{\pm}$and$d_{\pm}$ are the coefficients of the output and a state $|\alpha_i \beta_j\rangle$ is a state with one photon with polarisation i in state α and one photon with polarisation j in state β. c and d are the two outputs of the beamsplitter. A beamsplitter is described by the following transformation of the creation operators:[10]

$$\begin{pmatrix} a^\dagger \\ b^\dagger \end{pmatrix} \rightarrow \begin{pmatrix} t & ir \\ ir & t \end{pmatrix} \begin{pmatrix} c^\dagger \\ d^\dagger \end{pmatrix} \tag{5.14}$$

We calculate the coefficients after interaction with a beamsplitter:

$$\begin{aligned} |\psi^{\pm}\rangle &= \frac{1}{\sqrt{2}}(a_H^\dagger b_V^\dagger \pm a_V^\dagger b_H^\dagger) |\varnothing\rangle \\ &= \frac{1}{\sqrt{2}}\Big(t^2 c_H^\dagger d_H^\dagger - r^2 d_H^\dagger c_V^\dagger + irt(c_H^\dagger c_V^\dagger + d_H^\dagger d_V^\dagger) \\ &\quad \pm \Big(t^2 c_V^\dagger d_H^\dagger - r^2 d_V^\dagger c_H^\dagger + irt(d_V^\dagger d_H^\dagger + c_V^\dagger c_H^\dagger) \Big) \Big) |\varnothing\rangle \\ &= \Big((t^2 \mp r^2) c_H^\dagger d_V^\dagger - (r^2 \mp t^2) c_V^\dagger d_H^\dagger + (irt \pm irt)(c_H^\dagger c_V^\dagger + d_H^\dagger d_V^\dagger) \Big) |\varnothing\rangle \end{aligned} \tag{5.15}$$

where $|\varnothing\rangle$ is the vacuumstate. Comparing this with 5.13 yields:

$$\begin{aligned} a_{\pm} &= \frac{1}{\sqrt{2}}(t^2 \mp r^2) \\ b_{\pm} &= -\frac{1}{\sqrt{2}}(t^2 \mp r^2) \\ c_+ &= d_+ = 2irt \\ d_- &= c_- = irt - irt = 0 \end{aligned} \tag{5.16}$$

We then calculate the qubit error rate or QBER for the $|\psi^{\pm}\rangle$ state, denoted $Q^{\pm}$. The QBER is given by the zero terms in eq. 5.16, so:

$$\begin{aligned} Q^{+} &= |a_{+}|^2 + |b_{+}|^2 = (1-2r^2)^2 = 1 - 2r^2 + 4r^4 \\ Q^{-} &= |c_{-}|^2 + |d_{-}|^2 = 0 \end{aligned} \tag{5.17}$$

where the rule that $t^2 = 1 - r^2$ has been used and is derived from the unitarity of the beamsplitter matrix.[1]

$Q^{\pm}$ is obviously zero when $r = \frac{1}{\sqrt{2}}$. We now see what happens when we increase r by a little and propagate the change using the same technique used for propagating errors. For this we will need the derivatives of the QBER:

$$\begin{aligned} Q^{+\prime} &= (1 - 4r^2 + 4r^4)' = 4 \cdot 4r^3 - 4 \cdot 2r \\ Q^{-\prime} &= 0 \end{aligned} \tag{5.18}$$

The problem is that these are both zero for perfect 50:50 beamsplitters. So for first order error-propagation yields nothing: $\sigma^2_{Q_{\pm}} = \left(\frac{dQ^{\pm}}{dr}\big|_{r=\frac{1}{\sqrt{2}}} \right)^2 \sigma_r^2 = 0$.

So to get any estimate of the QBER caused by a perturbation to r we must use second order error propagation. This requires the second derivative of the QBER:[17]

$$\begin{aligned} Q^{+\prime\prime} &= (1 - 4r^2 + 4r^4)'' = 4 \cdot 4 \cdot 3r^2 - 4 \cdot 2 \\ Q^{-\prime\prime} &= 0. \end{aligned} \tag{5.19}$$

Second order error propagation also requires a distribution of the error. Assuming a Gaussian distribution we get a skew of zero and a kurtosis of three; written as $\gamma = 0$ and $\kappa = 3$. Using second order perturbation theory we obtain:[17]

$$\begin{aligned} \sigma^2_{Q_+} &= \left(\frac{dQ^{\pm}}{dr}\big|_{r=\frac{1}{\sqrt{2}}} \right)^2 \sigma_r^2 + \gamma \frac{dQ^{\pm}}{dr} \frac{d^2Q^{\pm}}{dr^2}\big|_{r=\frac{1}{\sqrt{2}}} \sigma_r^3 + \frac{\kappa - 1}{4} \left(\frac{d^2Q^{\pm}}{dr^2}\big|_{r=\frac{1}{\sqrt{2}}} \right)^2 \sigma_r^4 \\ &= \frac{1}{2} \left(\frac{16 \cdot 3}{2} - 8 \right)^2 \sigma_r^4 \\ &= \frac{1}{2} (16)^2 \sigma_r^4 \\ &= 16 \cdot 8 \cdot \sigma_r^4. \\ \sigma^2_{Q_-} &= 0 \end{aligned} \tag{5.20}$$

Unfortunately suppliers don't announce the imperfections of their beamsplitters in terms of erros in r but in terms of errors in $R = r^2$, for this however we can restrict ourselves to firstorder perturbation theory. So:

[1] The beamsplitter is unitary since all losses are accounted for elsewhere in the model.

$$\sigma_r^2 = \left(\frac{dr}{dR}|_{R=\frac{1}{2}}\right)^2 \sigma_R^2 = \left(\frac{d\sqrt{R}}{dR}\right)^2 \sigma_R^2 = \left(\frac{1}{2}\frac{1}{\sqrt{\frac{1}{2}}}\right)^2 \sigma_R^2 = \frac{1}{2}\sigma_R^2. \tag{5.21}$$

We get the QBER from coupling ratio error as:

$$\begin{aligned} \sigma_{Q_+}^2 &= 8 \cdot 16\sigma_r^4 = \frac{8}{4} \cdot 16\sigma_R^4 = 32\sigma_R^4 \Rightarrow \sigma_{Q^+} = \sqrt{2} \cdot 4\sigma_R^2 \\ \sigma_{Q^-} &= 0 \end{aligned} \tag{5.22}$$

σ_{Q^-} is always zero since the derivatives of $i(rt - rt)$ are always zero. Because we expect to get both of the states $|\psi^+\rangle$ and $|\psi^-\rangle$ equally often the expectation value of the QBER from coupling ratio error is:

$$\sigma_Q = \frac{Q^+}{2} + \frac{Q^-}{2} = 2^{\frac{3}{2}}\sigma_R^2 \tag{5.23}$$

Limitations of the model

The model has it's limitations. In particular it does not account for two photon emission which are a significant fraction of emissions. This was not done for a few reasons: firstly the fraction of two photon states depend upon the individual source and it was not clear what source would be used; including 2 photon emissions would entail a great expansion of the model, that being the downside of making a model in a spreadsheet; thirdly the two photon emission is a ever shrinking fraction of total emissions as technology improves; fourth and finally only so much accuracy is needed for choosing a components for an experiment: let's not over do this.

So including two photon emissions would be a lot of trouble, for a small gain, to include an unknown factor, that shrinks with time, for an analysis that doesn't even need it in the first place. Working on two photon emissions would have been focusing on the code while losing sight of the goal.

Resolution of the project

An assessment revealed that the best performing device would contain a PNH1505R5A2 beamsplitter from Thorlabs and two polarising beamsplitters from OZ Optics. These components would cost 2.476€ and give a final key rate of 0.67kBits/s. The final key rate is conditional on using the SNSPDs already owned by the group.

This is a fairly realistic estimate.

As Beatrice and I were preparing to propose this project Beatrice got a new job. My new supervisor, Mikkel, said he had no interest in continuing this project.

Study	**keyrate [bits/s]**	**keyrate [bits/pulse]**
Guang-Zhao Tang Et Al. (2016)	†	$4.4 \cdot 10^{-6}$
Zhiyuan Tang Et Al. (2016)	†	$2.48 \cdot 10^{-6}$
Hui Liu Et Al. (2019)	343	†
Guang-Zhao Tang Et Al. (2021)	†	$5.44 \cdot 10^{-7}$
My simulation (2022)	672	$9.25 \cdot 10^{-6}$

Table 5.3: The key rate of published experimental papers. † means that this value was not disclosed in the paper. As we can see the setup would have been world class in terms of both key rate and bits per pulse.

Chapter summary

- MDI-QKD can eliminate certain weaknesses in QKD because letting Eve know what the result of the detectors gives her no information on the state of Alice and Bobs bits.
- Doing MDI QKD requires a bell-state measurement.
- There are four kinds of component imperfection that all result in either a lower gain or a lower QBER.
- Some sources of error were not taken into account by the simulation paper but I have.

Chapter 6

Postprocessing in practice

TOWARDS the end of the project it was decided that it would be good if the Hy-Q group had its own "in house code" for QKD postprocessing. This will require us to go into more details with the theory behind the postprocessing.

To this end an application was written in python, see figure 6.1.

The idea was making a small app that could send messages between each other as demanded by postprocessing. If for instance two parties are to post process a key and they are in separate locations they would need some means of communication. In order to communicate over the internet they need to know each others IP address's. The app helps in this regard by displaying your IP address.

To establish contact or perform any other function the user has to type in a message starting with a command phrase from the command library. The command library is a python dictionary that translates the first space separated string of characters into a function that takes the rest of the message as an argument.

For instance if "con " is written at the start of a message then what follows is interpreted as an IP address.

The party that starts the session is labelled "Alice" internally, but they need not be the Alice from the physical protocol.[1, ch. 6.0]

As Bobs system responds that it is ready, which it does after Bob gives an affirmative response to dialog box, Alice's system starts checking files named "TestData(*).npz" with the * denotes a wildcard. Once it finds it the script starts doing post selection. The idea is that any raw key can be saved under such a name to be automatically processed.

Postprocessing recap

We recall from the introduction that the postprocessing of the key can be translated into the following parts:

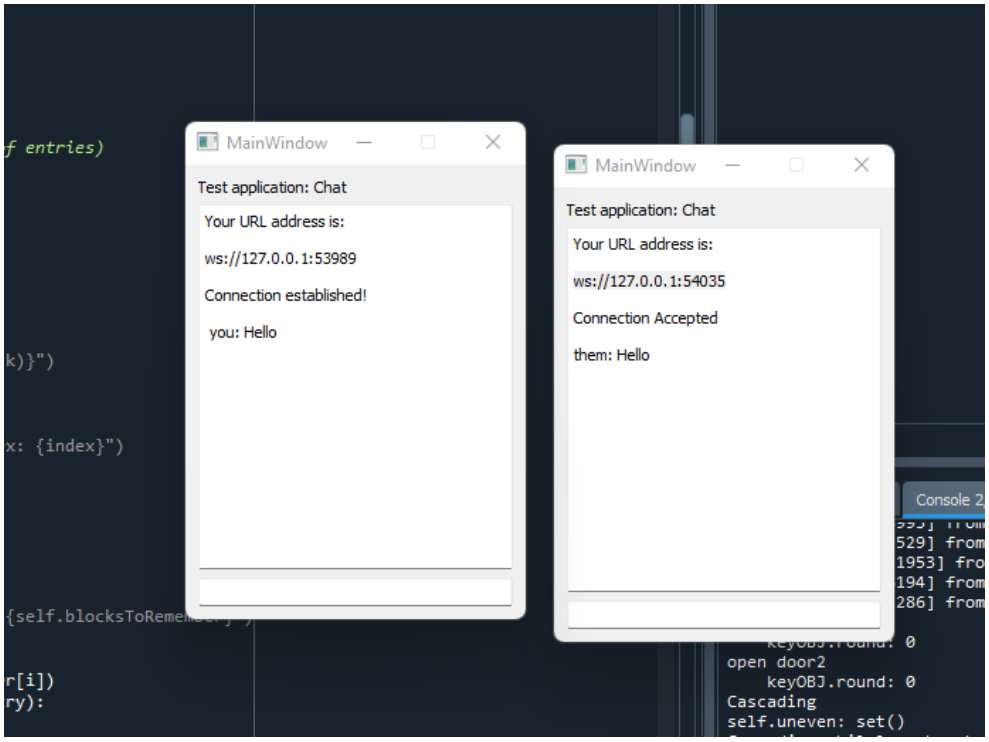


Figure 6.1: The application described in this section. Here a connection was made between the two instances of the app using the IP provided on screen. Also the message hello was sent (not encrypted).

1. Post selection (also referred to as sifting) where Alice and Bob see which photons they generated/measured using the same basis.

2. Estimation of the error rate where they find the QBER.

3. Error correction in which Alice and Bob change their key as to make them exactly alike.

4. Privacy amplification where Alice and Bob use their partially secret raw key to make a shorter fully secure secret key.

5. Authentication of the messages is necessary to ensure that Eve is not impersonating Alice or Bob.

All but the last were programmed. Though privacy amplification was never implemented.

Phrase	**Usage**
Command library:	
con	Establishes a connection with the IP address in the rest of the message.
Communications library:	
red	Bob sends this message to Alice to inform her that he has accepted her request to open the session.
Mes*	Meant as a way for the systems to denote a message to the users. For now the system treats any message not starting with a command phrase as a message for the users.
PBV*	Publishes the value of a single bit. Was left redundant by CPB.
PBQ	This phrase is followed by an array of bits from which the other party is meant to estimate the QBER of a raw key.
EQB	This phrase is followed by the QBER of a block is used to communicate this information to the other party.
CBP	This is a request for the other party to return the parity of a subset of blocks.
CPB	This message is followed by the parity of a block about which information was requested.
CLM*	This was meant to indicate that this message contained a series of sub messages that were all being processed and authenticated together. Many messages are short and it would be a waste of key rate to authenticate them all separately.
PPS	A phrase followed by arrays containing the time and basis of measurements. Used for post selection of measurements.

Table 6.1: A list of commands in the command and communications libraries. The command library is used for the users of Alice and Bob to control their systems and the communications library is for the systems Alice and Bob to communicate with each other. An * after a phrase indicates that it has not been implemented yet.

A note on communication

The way communication works is similar to how interpreting from the user works. The scripts send text messages to each other and upon arrival the script reads the first word and looks it up in a dictionary called comLibrary. The output of the dictionary is a function which is then called.

The disparate parts of a message (key ID, blockIndex or a number) are separated by "//" so the receiver can separate the values.

All keys have a key ID. This is a string that uniquely distinguish that key. When Alice and Bob exchange information the ID of the block is included with that message. This enables multiple keys to be processed in parallel.

Post selection

The first thing to happen is that Alice sends a list of times when she thinks she sent a photon and another list of the corresponding bases. Bob receives this message and remove all data-points where the time does not match up.[1] He then sends the time and basis choice of his remaining key to Alice such that she too can remove bits created with the "wrong" basis. She then tells Bob when she is done and then Bob starts the error correction.

Estimating the error rate

The estimation of the error rate consists of Bob sending a small fraction of the key for post selection. Instead of sending the index of all the blocks only a seed is sent. A seed is a number that a pseudo-random number generator can use as a starting point to generate other random number in a deterministic way; so same seed, same random number. In this case the (pseudo) random number is used by Bob to divide key into chunks. Then Bob sends them to Alice along with the seed. Alice uses the seed to divide her own key in the same way as Bob did before comparing, such that she compares like to like without needing a long set of indices to be sent. For a real-world system the seed should be chosen by dedicated random number generation hardware but for test cases using standard library functions is fine.

The advantage of using a seed is that it does not require a lot of data to send a single number. Alice then counts the fraction of signals where Bob and her got different results, i.e. the QBER.[19, ch. IV.A.1] This value is used for setting parameters for Cascade and would be used for determining how much privacy amplification is needed or if it can be accomplished at all.

[1] There is currently no way to take into account a time delay between Alice sending a qubit and Bob measuring it.

Error correction

How can Alice and Bob correct their errors without handing Eve the key?

The first such algorithm[16] called Binary works by dividing the key into blocks. Then Alice and Bob both announce the parity of the block i.e. the sum of bit values modulo 2. The blocks with a parity mismatch are further divided into sub-blocks for which the process is repeated until they reach a block-size of one bit which Bob can then correct.[3] This algorithm has to be applied multiple times in order to fix all errors.

It was noted that each run of Binary after the first reveals what errors were missed by preceding runs of Binary. Correcting these errors then reveal what errors were missed all other runs of Binary that were performed. To take advantage of this information the currently most common method for secret key reconciliation Cascade was created.[16]

Cascade works by running Binary over a string a set number of times. After each run the blocks from other runs are checked for errors. If any are found then the corrections of these might reveal more errors. This can result in a Cascade, hence the name.[2]

Implementation

When starting error correction a few things are done to make later steps easier. First of all the Numpy index over what blocks have already been corrected is saved.

The way the protocol works is by Alice requesting blocks from Bob. Alice only needs to keep track of how many rounds they have already been through, the blocks from those rounds and whether or not they have an even or uneven number of errors. When she gets the parity of a set of blocks back she picks the ones where her parity mismatches Bobs and corrects the bits she can. Afterwards if none of her blocks are need further parity checks she can check if there are any blocks from previous rounds with an uneven number of errors. If there is she fixes them. If not then she needs to request new blocks from Bob to make further progress: so this round is done.

Exact details such as the size of the blocks and how many rounds to do. [16] has a list of optimised Cascade functions but to start with I only used their suggested values for the original Cascade algorithm. Which is four passes the first of which has a block size of $k_1 = \frac{0.73}{QBER}$ and the following have a size of $k_i = 2k_{i-1}$ and random shuffling between passes.

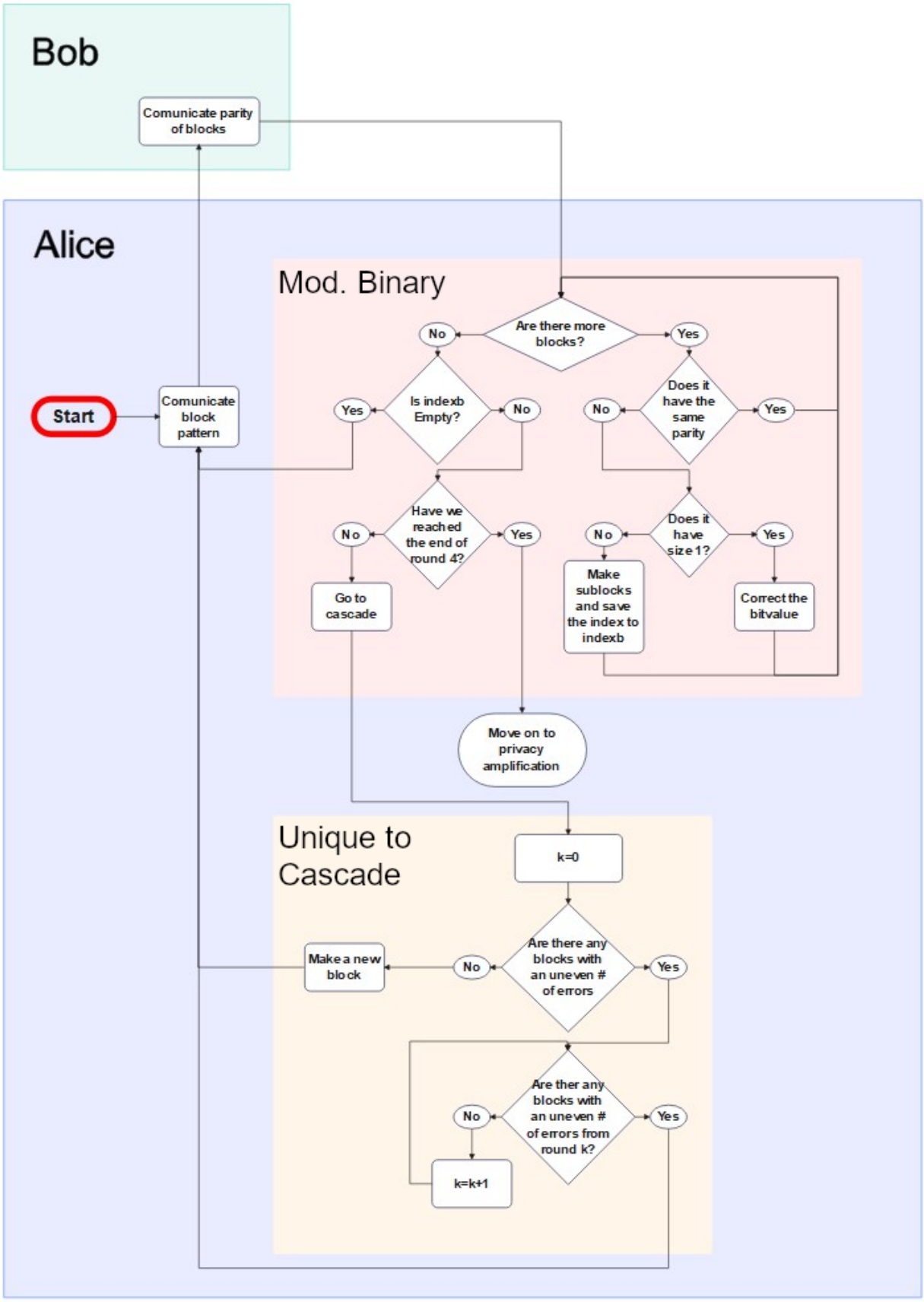


Figure 6.2: A flowchart of the implementation of Cascade. The red box denotes a modified Binary algorithm and the yellow box denotes parts unique to Cascade.

Privacy amplification

The basic framework

Let's start with a toy model of eavesdropping followed by privacy amplification.

Eve knows the value of n bits of a raw key of length i with $i > n$. If Alice and Bob knew which bits Eve has access to, then Alice and Bob could discard those bits and thereby have a secure key.[2]

In a more realistic case however they do not know what bits Eve has measured but do know the value of n. What they can do is scrambling their key first using a deterministic function such that changing a single bit in the input completely scrambles the output. Then if Alice and Bob cut away $n+1$ bits, leaving them with $i-n-1$ pseudo random bits, then Eve has only little information on the remaining bits. To see that this would work lets examine the following strategy:

Eve finds all the possible secret keys by keeping the bits she knows constant and systematically go through every possible combination for the remaining $i-n$ bits. This way you can at least eliminate some of the options.

This leaves Eve with 2^{i-n} possibilities, but there are only 2^{i-n-1} possibilities for what the secret key could be. So Eve still has twice as many random numbers as there are possible secret keys. Chances are Eve has only eliminated very few potential keys. So this strategy still results in very low information on the state of the final key.

To ensure information theoretic security we need more mathematical rigour.

Shannon entropy

Let us start with a definition of what it means to have a secret key. The basic notion is that Eve has no lower entropy on the key than she someone who guessed randomly. That is $H(K|Z) = H(K)$ where K is the secret key, Z the information Eve has gained by eaves dropping and $H()$ is the Shannon entropy function defined as:[1, ch. 3.1]

$$H(P) = -\sum_i p_i \cdot log_2(p_i) \tag{6.1}$$

The Shannon entropy of a message, such as a series of bits, is the lower limit on the number of bits that can convey the message.[1, ch. 3.1] This gets easier to understand with an example. Let us say that George is making a simple game. In the game the player can open a door and see what is inside before moving on to the next door. Behind each door the player finds either nothing (50% probability), pile of gold (37.5% probability), a treasure chest (12.5% probability) or a goat (0% probability). The goat was added as a prank hence the low probability.

In order to gather data on his players George makes the game such that his server decides whats behind the door and communicates this choice to the players.

George wonders how small he can make the message. For instance he could send the entire picture to be shown on the users screen; however this would require sending a the same three pictures many times. Alternatively he could send all the pictures once, and then just send the strings 'Nothing', 'Pile', 'Chest' and 'Goat'. Given that a picture can fill kilobytes and ASCII characters are one byte each this is already a thousand times more efficient. George further realise that he can get even more efficient by using the much shorter strings 'N', 'P', 'T' and 'G'; and better than that by sending only the binary strings '00', '01', '10' and '11' for an average of 2 bits per symbol. Can he get them shorter?

The limit to how small you can make a message is, as mentioned before, given by the Shannon entropy. In this case the Shannon entropy is:

$$\begin{aligned} H(G) &= -\frac{1}{2} \cdot log_2(\frac{1}{2}) - \frac{3}{8} \cdot log_2(\frac{3}{8}) - \frac{1}{8} \cdot log_2(\frac{1}{8}) - 0 \cdot log_2(0) \\ &= -\frac{1}{2} \cdot (-1) - \frac{3}{8} \cdot (-1.415) - \frac{1}{8} \cdot (-3) - 0 \\ &= 1.406 \end{aligned} \tag{6.2}$$

. So we need an average of 1.406 bits per symbol; our symbols being nothing, a pile of gold, a treasure chest and a goat. To get a better result than 2 bits George could denote an empty room by '1', a pile of gold by '01', a treasure chest by '001' and a goat by '000' would require an average of:

$$\frac{1}{2} \cdot 1 + \frac{3}{8} \cdot 2 + \frac{1}{8} \cdot 3 + 0 \cdot 3 = 1.625 \tag{6.3}$$

bits per symbol.

There are established methods such as Hoffman encoding that can get him arbitrarily close to an average cost of 1.406 bits per symbol but there is no algorithm that can beat that.[1, ch. 3.1.2] Therefore the Shannon entropy is the minimum number of bits needed to communicate a message. So the Shannon entropy is the information a message contains measured in bits.

Definition of security

A key can be said to be perfectly secret if the Shannon entropy of the key given Eves information is equivalent to Shannon entropy of the key without her information.[20, ch. II.10] That is if the shortest message communicating the key to Eve would be equivalent to sending her the entire key. In short the key is safe if all she needs to have the key is the key itself.

This is however strictly impossible as Eve could always randomly guess the right basis every time and there by have a small chance of getting the key.

So we say that Alice and Bob have to be able to make their key arbitrarily close to that threshold definition given above. Specifically they have to be able to set it to:[2]

$$H(S) - H(S|E) = 2^{-T} \tag{6.4}$$

where $H(S)$ is the Shannon entropy of the key, $H(S|E)$ is the Shannon entropy on the key given Eves information and T is a parameter that Alice and Bob can set arbitrarily high.

Proof of security

We define the Renyi entropy as:[2]

$$R(P) = -\sum_i log_2(p_i^2) \tag{6.5}$$

The Renyi entropy is upper bounded by the Shannon entropy.[1, ch. 6.3.1] $H(x|y) > R(x|y)$

We also define a family of functions G to be universal if for every $g \in G$ less than $\frac{|H|}{|B|}$ of inputs $x_1 \neq x_2$ that $g(x_2) = g_2(x_2)$. Where H is the size of the family and B is the size of the co-domain of the functions in the family respectively.[2]

If we have a key W of length n and we have a family of universal functions G then the Renyi entropy of $g(W)$ is upper bounded by:

$$H(W|GV) > R(W|GV) > r - log_2(1 + 2^{r-R(x)}) > r - \frac{2^{r-R(x)}}{ln(2)} \tag{6.6}$$

where r is the length of the key. The first step is justified by the Renyi entropy being upper bounded by the Shannon entropy and the last step by the inequality $log_2(1 + y) \leq \frac{y}{ln(2)}$.

Justifying the second inequality requires a bit more work. First we note that we can write the Renyi entropy as:

$$R(P) = \sum_i -log_2(p_i^2) = -log_2(p_c) \tag{6.7}$$

where $p_c = \sum_i p_i^2$ is the collision probability, which is the likelihood that two draws from the random distribution will give the same result. So for instance flipping a coin has a collision probability of 50% since the chance of getting heads or tails twice in a row is 25% each and there are only those two options for getting the same result twice.

With this way of rewriting the Renyi entropy we can write:

$$\begin{aligned} R(G(x)|G) &= \sum_g p_g \cdot R(G(x)|G=g) \\ &= \sum_g p_g \cdot (-log_2(P_c(G(x)|G=g))) \\ &\geq -log_2\left(\sum_g P_G(g) P_c(G(x)|G=g)\right) \end{aligned} \tag{6.8}$$

where the last step is justified by Jensens inequality.[2]

From here we just need to rewrite the term inside the logarithm of equation 6.8 as:

$$\begin{aligned} \sum_g P_G(g) P_c(G(X)|G=g) &= P(G(X_1)=G(X_2)) \\ &= P(X_1=X_2) + P(X_1 \neq X_2) \cdot P(G(X_1)=G(X_2)|X_1 \neq X_2) \\ &\geq P_c + (1-P_c)\cdot 2^{-r} \end{aligned} \tag{6.9}$$

justified by G being a family of universal hashfunctions and therefore, by definition, having a probability of $g(x_1) = g(x_2)$ of at most $\frac{1}{|B|}$. So if the function has outputs of r bits then $\frac{1}{|B|} = \frac{1}{2^r} = 2^{-r}$.

Continuing eq. 6.9 we find that:

$$\begin{aligned} &\geq P_c + (1-P_c)\cdot 2^{-r} \\ &= P_c \cdot (1-2^{-r}) + 2^{-r} \\ &> P_c + 2^{-r} \\ &= 2^{log_2(P_c)} + 2^{-r} \\ &= 2^{-R(X)} + 2^{-r} \\ &= 2^{-r}\left(1 + 2^{r-R(X)}\right) \end{aligned} \tag{6.10}$$

We now insert eq. 6.9 & 6.10 into eq. 6.8 and get:

$$\begin{aligned} R(G(x)|G) &\geq -log_2(\sum_g P_G(g) P_c(G(x)|G=g)) \\ &> -log_2\left(2^{-r}\left[1 + 2^{r-R(X)}\right]\right) \end{aligned} \tag{6.11}$$

Applying the rules that the Renyi entropy is upper bounded by the Shannon entropy and the inequality $log_2(1+y) \leq \frac{y}{ln(2)}$ we get 6.6.[2]

[2] If you, like I, are confused by the change from $\geq$ to $\leq$ remember that eq6.8 has a negative logarithm, thus the direction of the inequality is flipped.

However it is very difficult to calculate the Renyi entropy due to phenomena like "spoiling knowledge", the fenomenon that giving Eve slightly more information may *increase* her Renyi entropy[1, ch. 6.3.1], and calculating the key rate seems impossible. However with BB84 there are some shortcuts that allow for calculating a key rate. The following is taken from [2]

First we note that if Eve is only allowed a string $e(W)$ where e is a function $e : 0,1^n \to 0,1^t$, where $0,1^l$ is the set of all binary stings of length l, n is the length of the error corrected key shared by Alice and Bob, t is an arbitrary integer and e is an arbitrary function.

Let's call the shared reconciled key W, the output of Eves eaves dropping function T and label the output of $g(W)$ as K. If we say that t is smaller than n some inputs of e will result in the same output. The number of different values for W that result in the same output for V can be labelled as c_v. If all values of W occur with the same probability then the probability of any value for W given V is $P(W|V=v) = \frac{1}{c_v}$ and so the collision probability for W given Eves information v is given as $P_c = \frac{1}{c_v}$ making the Renyi entropy $R(W|V=v) = log_2(P_c) = log_2(\frac{1}{c_v})$.

Using equation 6.11 we find:

$$\begin{aligned} H(G(W)|G,V=v) &\geq R(G(W)|G,V=v) \geq r - \frac{2^{r-log_2(\frac{1}{c_v})}}{ln(2)} \\ &= r - \frac{2^r}{ln(2)\cdot c_v}. \end{aligned} \tag{6.12}$$

Before averaging over all values for v we note that $P(V) = c_v \cdot 2^{-n}$ and also we define a security factor s as $s = n - t - r$. We get:

$$\begin{aligned} H(W|GV) &= \sum_v P(V)\cdot H(W|G,V=v) \\ &> \sum_v c_v \cdot 2^{-n} \cdot \left(r - \frac{2^r}{ln(2)\cdot c_v}\right) \\ &= r - \frac{2^{r-n+t}}{ln(2)} \\ &= r - \frac{2^{-s}}{ln(2)} \end{aligned} \tag{6.13}$$

meaning that in order for Eve to get the key she needs a message at least as long as the key except for a factor that Alice and Bob can decrease exponentially by increasing the number of bits sacrificed.

Note that this is given as an average. This is necessary because an unconstrained eavesdropper can choose the function $e()$ to give a unique value for $2^t - 1$ values of W and then give 1 value for all other values of W. This means that Eve would have a chance at getting the correct value, but the probabillity

is $\frac{2^t}{2^n} = 2^{-r-s}$ so this can be decreased exponentially as demanded by the definition of security given above demands. Still this is worth noting as it means that succesfull eavesdropping can only ever be made statistically improbable.

There are many things worth noting about this result. To start with let's point out the similarities to eq. 2.4. Since s is given as $s = n - t - r$, the length of the final key r is given as:

$$r = n - t - s \tag{6.14}$$

n is the length of the raw key and t is eves information. This leaves three key differences:

1. Eq. 6.14 is given in terms of key length were as eq. 2.4 is given in terms of key rate. This is because eq. 6.14 was derived using blocks of static size.

2. Another key difference is that s is a security parameter absent from eq. 2.4. This is because the number of sacrificed bits trends to zero as the block size increases and eq. 2.4 works in the asymptotic limit.[19, ch. II.B.4]

3. Lastly eq. 6.14 does not include a term for the error correction. This is because the proof assumed that the key already has been corrected already.[2]

Only the last of these really matter for the security of QKD, the other two being a matter of convention. However it can be proven that the key remains secure with probability 2^{-s_m} if during privacy amplification an additional $M + 2s_m + 2$ bits are sacrificed, where M is the number of bits published during error correction and s_m is the security factor.[1, ch. 6.3.2]

It should also be noted that giving Eve access to only a deterministic function is an inaccurate model of how quantum eavesdropping works since quantum eavesdropping is probabilistic in nature. However not only are there more elaborate proofs that give the same key rate there is also the De Finetti theorem which states that the asymptotic keyrate cannot exceed the bounds given here.[19, ch. III.B.2.c] Note that this is only true for BB84 and *some* of the other protocols out there and is not true in general.

Lastly we may want to use a family of hash function that is not exactly universal. Recall that a universal family of hash functions H has the property that for all functions $g \in H$ and with valid parameters X the set of inputs $x_1, x_2 \in X$ such that $g(x_1) = g(x_2)$ is at most of size $\frac{|B|}{|H|}$ where B is the set of all valid outputs.

An almost universal family of hashfunctions H_ε has the property that for all functions $g \in H_\varepsilon$ and with valid parameters X the set of inputs $x_1, x_2 \in X$ such that $g(x_1) = g(x_2)$ is at most of size $\frac{\epsilon|B|}{|H_\varepsilon|}$ where ε is a parameter

determining the degree of universality. If $\varepsilon = 1$ the family is universal and if $\varepsilon = 0$ the family is not universal.[1, ch. 6.3.1]

We can use almost universal hash functions for privacy amplification at the cost of having to sacrifice an additional $2log_2(\varepsilon)$ additional bits. So the final number of bits to that can be extracted is:[1, ch. 6.3.1]

$$r = n - t - s - M - 2s_m - 2 - 2 \cdot log_2(\varepsilon) \tag{6.15}$$

Most of these terms are small however so it is still possible to have a substantial positive keyrate.

The number theoretic transform as an almost universal hash function

The number theoretic transform (NTT) is a transformation not unlike the Fourier transform.[1, ch. 7.3.2] The NTT takes in a vector in modulo n and outputs another vector modulo n with the same length. The NTT of a vector $\hat{r}$ is written as $\hat{R}$:[1, ch. 7.3.2]

$$R_j = F(\hat{r})_j = \sum_{i=0}^{L-1} r_i \cdot \omega^{ij} \tag{6.16}$$

with R_j being the jth index of $\hat{R}$, r_i being the ith index of $\hat{r}$, L being the size of the vector $\hat{r}$, F being the NTT and ω being the Lth root of unity. "The Lth root of unity" means that $\omega^L = 1$. An additional constraint placed upon ω is that $\omega^{L'} \neq 1$ for all $0 < L' < L$. Lastly all arithmetic operations are in mod m, meaning that $nm + k = k$.

Example of modular arithmetic: Say you want to add 7 and 18 in modulo 12. Well $7 + 18 = 25 = 2 \cdot 12 + 1 = 1$. Note that $18 = 12 + 6 = 6$ so we could equally have written $7 + 6 = 13 = 1 \cdot 12 + 1 = 1$. A physical example of mod 12 arithmetic is a clock. First one notices that $12 = 1 \cdot 12 + 0 = 0$ same as 12:00pm 12/April is the same time as 00:00am 13/April. Also if the clock says 7:00 and you wait 18 hours (the 7+18 example from before) it will be 1:00.

It is already worth noting the similarities with the Discrete Fourier Transform:[9, ch. 12.1.1]

$$y_j = \sum_{i=0}^{L-1} x_i \cdot \omega_L^{ij} \tag{6.17}$$

where y_j is the jth index of the Fourier transform, x_i is the ith element of the vector being transformed and $\omega_L = e^{-i\frac{2\pi}{L}}$. Note that $\omega_L^L = 1$. We will use this later to compute the NTT using a similar method as for the Fast Fourier Transform.

The NTT is useful as it allows for fast multiplication of polynomials mod x^{L-1}. To see this first we must realise that a vector $\hat{r}$ can be written as the polynomial in x $P_{\hat{r}} = \sum_{i=0}^{L-1} r_i \cdot x^i$ and vice versa. If we have two such vectors/polynomials $\hat{r}$ and $\hat{s}$ and we wish to multiply them to get $\hat{r}\hat{s} = \hat{t}$ with $t_j = \sum_{i=0}^{L-1} r_i \cdot s_{i-j}$ where the subscript is mod L and the multiplication is done in mod p then we could multiply element by elements using $\mathcal{O}(L^2)$ operations. We could also take the NTT of both $\hat{r}$ and $\hat{s}$ multiply their product and then take the inverse NTT.[1, ch. 7.3.2] This can be seen from the following expression:

$$t_j = \sum_{i=0}^{L-1} r_i \cdot s_{i-j} \tag{6.18}$$

$$T_i = \sum_{j=0}^{L-1} t_j \cdot \omega_L^{ij} = \sum_{j=0}^{L-1}\sum_{i=0}^{L-1} r_i \cdot s_{i-j} \cdot \omega_L^{ij} \tag{6.19}$$

$$\begin{aligned} R_i \cdot S_i &= \sum_{j=0}^{L-1}\sum_{l=0}^{L-1} r_j \cdot \omega^{ij} \cdot s_l \cdot \omega^{li} \\ &= \sum_{l=0}^{L-1}\sum_{i=0}^{L-1} r_j \cdot s_l \cdot \omega^{ij+il} \\ &= \sum_{k=-j}^{L-j-1}\sum_{i=0}^{L-1} r_j \cdot s_{j-k} \cdot \omega^{ij+i(k-j)} \\ &= \sum_{k=-j}^{L-j-1}\sum_{i=0}^{L-1} r_j \cdot s_{j-k} \cdot \omega^{i(j+k-j)} \\ &= \sum_{k=-j}^{L-j-1}\sum_{i=0}^{L-1} r_j \cdot s_{j-k} \cdot \omega^{ik} \end{aligned} \tag{6.20}$$

with l=k-j. Eq. 6.19 and 6.20 are equal because the sufixes are mod L, $\omega^L = 1$ and the commutative property of addition.

We now define the following family of hash functions:

Definition 1 ($H_{NTT,p,L,\beta}$)**.** *$H_{NTT,p,L,\beta}$ is a almost universal family of hash-functions $h_C(r) = F^{-1}(C \cdot F(r))_{[0;\beta]}$ where F is the NTT, F^{-1} is the inverse NTT and the subsript $[0;\beta-1]$ means that only picking the first β elements. The NTT is done with L elements and mod p where p is a prime number larger than L. The function has a universality of $\frac{p^\beta}{(p-1)^\beta}\frac{|B|}{|H|}$.*

If this hashfunction is used with $p = 110503$ and $\beta = 10000$ then $\epsilon = \frac{110503^{10000}}{110502^{10000}} = 1.0947$ which means reducing the output by $2log_2(\epsilon) = 0.26$ bits. Not a big sacrifice.

Timecomplexity and its importance

To understand why the NTT was used it is necessary to understand the concept of time complexity. Time complexity is a rough measure of how the time it takes to calculate something scales with time.[9, Notation] For instance finding the the product of all elements of a vector is done in $\mathcal{O} = n$ time where n is the number of elements in the vector this is because each time you add an element to the list you have to do one more computation. Alternatively having a square matrix act on the same vector is of order $\mathcal{O} = n^2$[1, ch. 7.2.1] because you have to multiply all n values of the vector with a row of n matrix values for all n rows of the matrix.

An important facet of time complexity is that only the fastest growing term remains.[9, Notation] Example: let's say that in addition to calculating the product of all elements in the vector noted before the program also had to locate the vector in memory. If the time it takes to locate the vector is independent of n then it is constant with regards to n meaning that the program as a whole is of time complexity $\mathcal{O}(n + c) = \mathcal{O}(n)$.

The reason for this and the reason for time complexity in general is that we want to what algorithm to use when dealing with a large data set to process. For small data set it really doesn't matter if we are processing them with a slow algorithm as modern processors make fast work of them anyways. For large data sets however only the scaling of the algorithm is important since an algorithm that solves a problem with $\mathcal{O}(c_1 log(n) + c_2)$ time scaling will eventually be faster than an algorithm that solves the same problem with scaling of $\mathcal{O} = c_3 e^n$ no matter the value of c_1, c_2 and c_3. For the same reasons any constant factor multiplication is also ignored. Note that getting a faster computer is ever only equivalent to changing the values of those variables c_i unless it gives access to different operations.

The reason for using the NTT instead of just multiplying two polynomials is that the NTT can be performed in $\mathcal{O}(nlog(n))$ time where as multiplying two polynomials via a shift and add algorithm is in $\mathcal{O}(n^2)$.[1, ch. 7.3.1] Given that we have to process gigabytes worth of raw key to get megabytes worth of secret key the speed and efficiency of the hashing algorithm becomes very important.

The implementation

The question now is how to implement the NTT with time complexity $\mathcal{O}(nlog(n))$. We will go by the explaination given in the original Cooley-tukey paper.[5] The NTT looks like this (see equation 6.16):

$$R_j = F(\hat{r})_j = \sum_{i=0}^{L-1} r_i \cdot \omega^{ij} \tag{6.21}$$

where the subscript denotes an index and ω is the Lth root of unity. If $L = k_1 \cdot k_2$ we can write $j = j_1 k_1 + j_0$, $i = i_1 k_2 + i_0$ and using the rule that $\omega_L^{k_1 K_2} = 1$ rewrite eq. 6.21 as:

$$\begin{aligned} R_{j_1 k_1 + j_0} &= \sum_{i_0=0}^{k_2-1} \sum_{i_1=0}^{k_1-1} r_{i_1 k_2 + i_0} \cdot \omega^{(j_1 k_1 + j_0)(i_1 k_2 + i_0)} \\ &= \sum_{i_0=0}^{k_2-1} \sum_{i_1=0}^{k_1-1} r_{i_1 k_2 + i_0} \cdot \omega^{(j_0 i_1 k_2)} \omega^{(j_1 k_1 + j_0) i_0} \end{aligned} \tag{6.22}$$

which we can then split into the following equations:

$$A_{i_0 j_0} = \sum_{i_1=0}^{k_1-1} r_{i_1 k_2 + i_0} \cdot \omega^{(j_0 i_1 k_2)} \tag{6.23}$$

$$R_{j_1 k_1 + j_0} = \sum_{i_0=0}^{k_2-1} A_{i_0 j_0} \omega^{(j_1 k_1 + j_0) i_0} \tag{6.24}$$

By declaring a single multiplication and addition a single operation and by calculating it the way implied by eq. 6.23 and 6.24 we need to perform k_1 operations to calculate any particular $A_{i_0 j_0}$. There are $k_1 \cdot k_2 = L$ possible values for $A_{i_0 j_0}$ leading to a total of $L k_1$ operations to calculate all A values.

Once we have those we can calculate the values of R_j. To calculate one value of R_j requires k_2 operations and there are L values of j for a total of $L + k_2$ operations to calculate all values of $\hat{R}$.

Adding these together we get $T = L k_1 + L k_2 = L(k_1 + k_2)$ operations. However both eq. 6.23 and eq. 6.24 are of the same form as eq.6.16 meaning that if k_1 or k_2 can written as the product of two numbers we can apply this method recursively.

To see how efficient this is lets examine the case where $L = p^n$. In this case the number of operations that it takes to calculate $\hat{R}$ is $T = L(p+p+p+...+p)$ where the ... implies that we are adding p n times. This can be rewritten as $T = L(p + p + p + ... + p) = Lnp = pLlog_p(L)$ so $\mathcal{O}(Llog(L))$. Note that this scaling does require that we can keep factorising L.

Implementation

Privacy amplification using the NTT was never implemented.

The NTT was set programmed in a separate script to make sure it functioned correctly. The plan was to write the code in a clean script that can run faster and where there would be fewer lines of code to examine when trying

to find bugs. This worked, but due to time constraints the code was never implemented in the base script which is therefore left unfinished.

The implementation of the NTT was based of the recursive algorithm found in [9, ch. 12.2] which implements the FTT recursively. As stated earlier the NTT and FTT are the same except for the exact expression for the nth root of unity. The way the recursive version of the FTT works is by splitting the vector being transformed into even and odd entries performing an FTT on both then concatenating the results by them selves, thereby getting two vectors of full length, and then multiplying all elements of the FTT of the uneven indices by ω^i where i is their index post concatenation. The concatenated vectors are then added element-wise.

How this is equivalent to the FTT is admittedly not intuitively obvious. To see it first we write with reference to eq. 6.22 $k_2 = 2$. Then eq. 6.23 reads:

$$A_{i_0 j_0} = \sum_{i_1=0}^{2^n-1} r_{2\cdot i_1 + i_0} \omega^{2 j_0 i_1} \tag{6.25}$$

Which is the FTT of every even or uneven element of $\hat{r}$. Then in eq. 6.24 the even elements are left unaltered because $\omega^{(j_1 k_1 + j_0)\cdot 0} = 1$ and the result of the FTT of the uneven elements is multiplied by ω to the power of the index.

The NTT is implemented like that but all operations are done in modular arithmetic. Additionally a function was made to convert between basis's. The idea was to convert the raw key into base 256 and then convert that into base 786433 to perform the NTT on a vector of length $2^18 = 262144$ with $\omega = 786433$. This would have meant the ability to process $262144 \cdot log_2(786433) bits \approx 626 kbyte$ of raw key for every two calls of the NTT. Given that running the NTT twice as described above takes 11.97 seconds on my laptop my laptop could process $\approx 80 kbit$ of raw key a second. Given that the field trial required processing 50kbit per second it means that this form of postprocessing would have lax hardware requirements.

Chapter summary

- There are a lot of stages to postprocessing, most of which have been implemented.
- Error correction can be done by the Cascade algorithm which uses previously corrected blocks to find new errors. Thereby cascading through multiple blocks.
- Using the information theoretic concepts of Shannon- and Renyi-entropy it is possible to enable Alice and Bob to ensure that Eves information on the final key is arbitrarily small. This results in her assigning roughly equal probability to all possible states of the key. This is called privacy amplification.

- To do privacy amplification Alice and Bob needs to agree to a universal or almost universal family of functions. The NTT can be used to make such a function.
- The NTT is fast enough to work with current key rates without any strict hardware requirements.

Chapter 7

Conclusion

Quantum key distribution is a technology that can ensure a private key is shared. This insurance is provided not by assumptions of an eavesdroppers technology and methods to solve a mathematical riddle but instead on limits provided by laws of science. The technology to make this work exist but still faces hurdles of practical application.

I have in this report documented my work over the last year in the field of quantum key distribution. The work can broadly be classified into the categories of improving the key rate using a quadrupler, helping out with a field trial, the creation of a model for analysing the effectiveness of a MDIQKD setup and the creation of a practical in-house system for post-processing the raw keys made by a QKD setup.

Bibliography

[1] Gilles van Assche. *Quantum Cryptography and Secret-Key Distillation*. Online-Ausg. Cambridge: Cambridge University Press, 2012. ISBN: 978-0-511-61774-4.

[2] C.H. Bennett et al. "Generalized privacy amplification". In: *IEEE Transactions on Information Theory* 41.6 (Nov. 1995), pp. 1915–1923. ISSN: 00189448. DOI: 10.1109/18.476316. URL: http://ieeexplore.ieee.org/document/476316/ (visited on 02/13/2023).

[3] Gilles Brassard and Louis Salvail. "Secret-Key Reconciliation by Public Discussion". In: *Advances in Cryptology — EUROCRYPT '93*. Ed. by Tor Helleseth. Vol. 765. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 410–423. ISBN: 978-3-540-57600-6. DOI: 10.1007/3-540-48285-7_35. URL: http://link.springer.com/10.1007/3-540-48285-7_35 (visited on 02/13/2023).

[4] Gilles Brassard et al. "Limitations on Practical Quantum Cryptography". In: *Physical Review Letters* 85.6 (Aug. 7, 2000), pp. 1330–1333. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.85.1330. URL: https://link.aps.org/doi/10.1103/PhysRevLett.85.1330 (visited on 02/03/2023).

[5] James W. Cooley and John W. Tukey. "An algorithm for the machine calculation of complex Fourier series". In: *Mathematics of Computation* 19.90 (1965), pp. 297–301. ISSN: 0025-5718, 1088-6842. DOI: 10.1090/S0025-5718-1965-0178586-1. URL: https://www.ams.org/mcom/1965-19-090/S0025-5718-1965-0178586-1/ (visited on 01/30/2023).

[6] Hua-Jian Ding et al. "Measurement-device-independent quantum key distribution with insecure sources". In: *Opt. Lett.* 47.3 (Feb. 2022), pp. 665–668. DOI: 10.1364/OL.447234. URL: https://opg.optica.org/ol/abstract.cfm?URI=ol-47-3-665.

[7] Guan-Jie Fan-Yuan et al. "Modeling Alignment Error in Quantum Key Distribution Based on a Weak Coherent Source". In: *Phys. Rev. Applied* 12.6 (Dec. 2019), p. 064044. DOI: 10.1103/PhysRevApplied.12.064044.

URL: https://link.aps.org/doi/10.1103/PhysRevApplied.12.064044.

[8] *Fiber Coupler Tutorials*. URL: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=10758 (visited on 10/28/2022).

[9] Michael T. Heath. *Scientific computing: an introductory survey*. Revised second edition, SIAM edition. Classics in applied mathematics 80. Philadelphia: Society for Industrial and Applied Mathematics, 2018. 567 pp. ISBN: 978-1-61197-557-4.

[10] John C. Howel. *Beam Splitter Input-Output Relations*. URL: https://www.pas.rochester.edu/~howell/mysite2/Tutorials/Beamsplitter2.pdf (visited on 11/10/2022).

[11] Thomas Ihn. *Semiconductor nanostructures: quantum states and electronic transport*. Oxford ; New York: Oxford University Press, 2010. 552 pp. ISBN: 978-0-19-953443-2.

[12] Masato Koashi. *Efficient quantum key distribution with practical sources and detectors*. Issue: arXiv:quant-ph/0609180. Sept. 23, 2006. DOI: 10.48550/arXiv.quant-ph/0609180. arXiv: quant-ph/0609180. URL: http://arxiv.org/abs/quant-ph/0609180 (visited on 10/24/2022).

[13] Chenyang Li et al. "Secure quantum communication in the presence of phase- and polarization-dependent loss". In: *Phys. Rev. A* 98.4 (Oct. 2018), p. 042324. DOI: 10.1103/PhysRevA.98.042324. URL: https://link.aps.org/doi/10.1103/PhysRevA.98.042324.

[14] Hoi-Kwong Lo, Marcos Curty, and Bing Qi. "Measurement-Device-Independent Quantum Key Distribution". In: *Phys. Rev. Lett.* 108.13 (Mar. 2012), p. 130503. DOI: 10.1103/PhysRevLett.108.130503. URL: https://link.aps.org/doi/10.1103/PhysRevLett.108.130503.

[15] Peter Lodahl, Sahand Mahmoodian, and Søren Strobe. "Interfacing single photons and single quantum dots with photonic nanostructures". In: *American Physical Society* 87.2 (May 11, 2015), p. 50. DOI: https://doi.org/10.1103/RevModPhys.87.347. URL: https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.87.347.

[16] Jesus Martinez-Mateo et al. "Demystifying the Information Reconciliation Protocol Cascade". In: (2014). DOI: 10.48550/ARXIV.1407.3257. URL: https://arxiv.org/abs/1407.3257 (visited on 11/16/2022).

[17] S. Mekid and D. Vaja. "Propagation of uncertainty: Expressions of second and third order uncertainty with third and fourth moments". In: *Measurement* 41.6 (2008), pp. 600–609. ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2007.07.004. URL: https://www.sciencedirect.com/science/article/pii/S0263224107000681.

[18] Jun John Sakurai and Jim Napolitano. *Modern quantum mechanics*. 2nd ed. Cambridge: Cambridge university press, 2017. ISBN: 978-1-108-42241-3.

[19] Valerio Scarani et al. "The security of practical quantum key distribution". In: *American Physical Society* 81.3 (Sept. 29, 2009), p. 45. DOI: https://doi.org/10.1103/RevModPhys.81.1301. URL: https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.81.1301.

[20] C. E. Shannon. "Communication Theory of Secrecy Systems*". In: *Bell System Technical Journal* 28.4 (Oct. 1949), pp. 656–715. ISSN: 00058580. DOI: 10.1002/j.1538-7305.1949.tb00928.x. URL: https://ieeexplore.ieee.org/document/6769090 (visited on 02/13/2023).

[21] Steven H. Simon. *The Oxford solid state basics*. 1st ed. Oxford: Oxford University Press, 2013. 290 pp. ISBN: 978-0-19-968077-1.

[22] *Single Mode FC/APC Fiber Optic Patch Cables*. URL: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=334 (visited on 10/24/2022).

[23] *Thorlabs - TW1550R5A2 2x2 Wideband Fiber Optic Coupler, 1550 ± 100 nm, 50:50 Split, FC/APC Connectors*. URL: https://www.thorlabs.com/thorproduct.cfm?partnumber=TW1550R5A2 (visited on 11/03/2022).

[24] Ravitej Uppu et al. "Quantum-dot-based deterministic photon–emitter interfaces for scalable photonic quantum technology". In: *Nature Nanotechnology* 16.12 (Dec. 2021), pp. 1308–1317. ISSN: 1748-3395. DOI: 10.1038/s41565-021-00965-6. URL: https://www.nature.com/articles/s41565-021-00965-6 (visited on 10/26/2022).

[25] Qin Wang and Xiang-Bin Wang. "Simulating of the measurement-device independent quantum key distribution with phase randomized general sources". In: *Nature* 4 (Apr. 14, 2014), p. 6. DOI: https://doi.org/10.1038/srep04612. URL: https://www.nature.com/articles/srep04612.

[26] Weilong Wang, Kiyoshi Tamaki, and Marcos Curty. "Measurement-device-independent quantum key distribution with leaky sources". In: *Scientific Reports* 11.1 (Dec. 2021), p. 1678. ISSN: 2045-2322. DOI: 10.1038/s41598-021-81003-2. URL: http://www.nature.com/articles/s41598-021-81003-2 (visited on 11/08/2022).

[27] W. K. Wootters and W. H. Zurek. "A single quantum cannot be cloned". In: *Nature* 299.5886 (Oct. 1982), pp. 802–803. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/299802a0. URL: http://www.nature.com/articles/299802a0 (visited on 11/04/2022).