

An introduction to a novel diagrammatic notation for quantum mechanics and its applications to some quantum cryptographic protocols

December 27, 2019

Abstract

Dirac notation is the most commonly used formalism for the description of quantum cryptographic protocols. Although experienced quantum physicists have no problem understanding these protocols in this notation, it fails to expose many of the features of quantum theory clearly and intuitively, especially those features that involve the interactions of multiple systems across time and space. Coecke and Kissinger recently proposed [3] a new - strictly diagrammatic - formalism for quantum theory in an aim to address this problem. Herein we build upon their work and apply it to the field of quantum cryptography. To this end, we work out four protocols (quantum key distribution, quantum key recycling, quantum teleportation and the quantum one time pad) and some of their respective equivalences diagrammatically. We also look into the security of each of these protocols and develop a fully fledged security proof for quantum key recycling without noise. Since the only assumed knowledge is undergraduate level quantum mechanics, this work also provides a handbook style introduction to the diagrammatic notation. Altogether, this results in a standalone work that both familiarizes the reader with the diagrammatic notation and subsequently provides them with diagrammatic representations of some recent findings and protocols in the field of quantum cryptography.

1 Introduction

The field of Quantum cryptography is concerned with the exploitation of the laws of quantum mechanics in order to derive secure methods of communication. In this field, classical cryptographic primitives are interwoven with quantum physics in order to make protocols for tasks such as sending a message securely in the presence of an adversary. Perhaps the first person to realize that there was something to be gained from using quantum mechanics for cryptography was Stephen Wiesner, at the time a professor at Colombia University in New York. His seminal paper, titled "Conjugate Coding", was originally rejected for publication by the IEEE Information Theory. According to some historical accounts it was too far ahead of its time [11]. After all, it was the first paper that described the encoding of information into quantum states and even provided two examples of its potential applications. It wasn't until a decade later that this paper was published by SIGACT news [16]. In hindsight, IEEE Information Theory likely regretted their choice for not publishing this article; quantum cryptography quickly grew into a large field, both in- and outside of academia. Within academia, this has resulted in a zoo of cryptographic protocols and schemes, many of them

with large bodies of research towards their optimization and security proofs. Outside of academia, large international companies such as ABN AMRO and Atos have picked up on some of these protocols, and are working on their practical implementations [1, 2].

The protocol that the Dutch bank is aiming to realize is Quantum Key Distribution (QKD). The aim of this protocol is a secure key exchange between two parties, Alice and Bob. This is achieved when, after the protocol, Alice and Bob are the only parties to share some secret key k . The security of this protocol relies on the fact that the measurement of a quantum state disturbs it [9], and that any eavesdropper will therefore always introduce some anomaly in the communicated quantum state. Bennett and Brassard were the first to develop a secure QKD scheme in 1984. A couple of years earlier, the same authors together with Breidbart had already developed another quantum cryptographic protocol, Quantum Key Recycling (QKR). QKR is also a protocol with the aim of achieving secure key exchange between two parties. From a quantum mechanical perspective, it relies on the same principles as QKD. The main differences are in the classical domain. In QKR, Alice and Bob can achieve communication in less steps compared to QKD. In recent work [8], Leermakers and Škorić even reduce classical communication between Bob and Alice to just one bit. In that same article, they also provide a security proof for their version of QKR. For this, they first transform it to a protocol that allows for a more straightforward security proof in a series of security wise-equivalent steps. They were not the first to do this, Shor and Preskill used protocol equivalences in this manner already in a publication in 2000 [15]. In fact, after Shor and Preskill's publication equivalences between protocols have often been exploited in the field of quantum cryptography.

Quantum cryptographic protocols and their respective equivalences are usually given in the formalism of Hilbert spaces. Although experienced quantum physicists will have no problem understanding protocols in this formalism, it fails to expose many of the features of quantum theory in a clear and intuitive way, especially those features that involve the interaction of multiple systems across time and space. Of course, in particle physics Feynman diagrams were introduced to show the interactions of elementary particles across time and space. In the field of quantum computing a potential solution to this problem is provided by quantum circuits, which are a model for visualizing a set of qubits as registers to which a sequence of quantum gates can be applied. Although this method may be satisfactory for quantum computing, it is rather limited in its capabilities and does not provide a good tool for the visualization of quantum mechanical processes in other fields such as quantum cryptography. Coecke and Kissinger [3] propose a new - strictly diagrammatic - formalism for quantum theory in general with the aim of addressing this problem. Their diagrammatic notation includes all of the functionality of quantum circuits, and is moreover well suited for visualizing a much wider range of quantum mechanical processes, among which quantum cryptographic protocols.

As their book was released only recently, their diagrammatic notation has yet to be appreciated by many quantum physicists. Given this fact, and the rise in complexity and popularity of quantum cryptographic methods, we propose to build upon the work of Coecke and Kissinger, and apply it to the field of quantum cryptography. The aim is to work out the recently introduced protocols diagrammatically together with various equivalences, among which those given in [8]. Since the only assumed knowledge is undergraduate level quantum mechanics, we also provide a handbook style introduction to the diagrammatic notation. If successful, this work could not only serve as an opportunity for researchers to view their research from a new, potentially refreshing, perspective, but also provide an intuitive, purely diagrammatic, security proof for QKR.

The structure of this work is such that it can be divided up into two parts. Chapter 2 is a fully

fledged introduction to the diagrammatic method for a reader who is familiar with quantum mechanics and Dirac notation. The rest of this report ventures into uncharted terrain. Building upon the introduced notation, we will translate findings and protocols from recent papers into diagrams. In chapter 3 we introduce QKD, QKR, and two more protocols, and build their diagrammatic representations. In chapter 4 we will look at equivalences between protocols. To conclude this report we also take a critical look at both the notation and conducted research in chapter 5, and summarize the main points in chapter 6. This work differs in structure from an ordinary scientific work since it also includes some points of self-reflection at the end, in chapter 7.

2 Diagrammatic Preliminaries

This section gives a concise introduction to the diagrammatic method for a reader familiar with undergraduate level quantum mechanics. Note that [3] provides a more in depth introduction into this diagrammatic method, but does not derive most concepts back to Dirac notation.

Appendix A contains all of the concepts discussed in this section as keywords with hyperrefs to the respective subsections.

2.1 From Dirac to Diagrams

2.1.1 States, Effects, and Hermitian Operations

The **ket** is defined as a triangle with its sharp edge down in diagrammatic notation. It can be interpreted as the preparation of a state, in this case ψ . It is referred to as **state** throughout the thesis.

$$|\psi\rangle \equiv \begin{array}{c} | \\ \psi \\ \triangle \end{array} \quad (1)$$

The **bra** in diagrammatic notation is the flipped state, and is referred to as **effect**.

$$\langle\phi| \equiv \begin{array}{c} \triangle \\ \phi \\ | \end{array} \quad (2)$$

Triangles are the smallest building blocks in the diagrammatic notation. Most diagrams can be reduced to just triangles, and subsequently to Dirac notation by equations (1) and (2). This makes them a powerful tool for translating complicated diagrams to Dirac notation and vice versa.

From the fact that the **Hermitian adjoint** of a bra gives a ket and reversedly it follows that the operation of flipping a diagram around its horizontal axis corresponds to taking the Hermitian adjoint diagrammatically. Flipping a diagram around its vertical axis is also a legal operation and this corresponds to taking the **Hermitian conjugate**. Both of these operations applied together takes the **transpose**. All of these diagram operations can be summarized as follows.

(3)

Equations 1 and 2 show the most general method of writing a bra and a ket in diagrammatic notation. However, the notation also allows for us to take these states and effects further apart. For the state this goes as follows.

$$\begin{array}{|}\psi\end{array} \equiv \begin{pmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_n \end{pmatrix} = \sum_{i=0}^n \psi_i \begin{array}{|}i\end{array} \stackrel{(1)}{=} \sum_{i=0}^n \psi_i |i\rangle \quad (4)$$

The effect is then simply the adjoint of equation 4. Note that the triangle in this equation is not on its side. This is because it is a basis vector, which is independent of conjugate transformations. At various points throughout the thesis we will also refer to the white state that is not on its side as representing a classical variable since it has similar properties.

2.1.2 Tensor Product

So far we have seen diagrams with no more than one diagrammatic component but this is soon to change. Therefore, we have to think about what it means for two components to be in one and the same diagram. Having multiple diagrammatic components in the same diagram that are not connected to one another is defined as the **tensor product** over those components. For example, vertically composing a state and effect translates as follows to Dirac notation.

$$\begin{array}{|}\psi \\ \phi\end{array} \equiv |\psi\rangle \otimes \langle\phi| = |\psi\rangle \langle\phi| \quad (5)$$

Horizontally composing two states or effects thus gives the tensor product over two kets and bras respectively.

$$\begin{array}{|}\psi\end{array} \begin{array}{|}\psi\end{array} \equiv |\psi\rangle \otimes |\psi\rangle \quad \begin{array}{|}\phi\end{array} \begin{array}{|}\phi\end{array} \equiv \langle\phi| \otimes \langle\phi| \quad (6)$$

2.1.3 Wires

The **identity** map in the diagrammatic notation is given by the following diagram, subsequently referred to as **wire**.

$$| \quad (7)$$

A wire can be reduced to triangles (and to a ket and a bra) as follows.

$$| \equiv \sum_i \begin{array}{c} \downarrow \\ \triangle_i \\ \uparrow \end{array} \stackrel{(1,2)}{=} \sum_i |i\rangle \langle i| \quad (8)$$

Note that we omit the limits in the summation symbol on purpose. Since we are working with two state systems for the entirety of this report the summation symbol without limits can be interpreted as the sum over 0 and 1 unless explicitly stated otherwise.

To exemplify the wire's behavior, we compose test in- and outputs.

$$\begin{array}{c} \triangle_\phi \\ | \\ \triangle_\psi \end{array} \stackrel{8}{=} \sum_i \begin{array}{c} \triangle_\phi \\ | \\ \triangle_i \\ | \\ \triangle_i \\ | \\ \triangle_\psi \end{array} \equiv \sum_i \langle \phi|i\rangle \langle i|\psi\rangle = \langle \phi|\psi\rangle \quad (9)$$

Furthermore, we can also bend around wires.

$$\text{bend} \equiv \sum_i \begin{array}{c} \triangle_i \\ | \\ \triangle_i \end{array} \quad (10)$$

Composing test inputs.

$$\begin{array}{c} \text{bend} \\ \triangle_\psi \quad \triangle_\phi \end{array} \stackrel{(10)}{=} \sum_i \begin{array}{c} \triangle_i \\ | \\ \triangle_i \end{array} \begin{array}{c} \triangle_i \\ | \\ \triangle_i \end{array} \stackrel{(9)}{=} \sum_i \langle i|\psi\rangle \langle \phi|i\rangle = \langle \phi|\psi\rangle \stackrel{(9)}{=} \begin{array}{c} \triangle_\phi \\ | \\ \triangle_\psi \end{array} \quad (11)$$

The ϕ thus gets 'bent around' along with the wire such that we get back to the situation from equation (9).

Intuitively, wires can be seen as information carriers. Every wire has an associated **system type**, the space of the information that it carries. In the context of this report, the system types are generally Hilbert spaces. In section 2.1.11 we will also allow tensor products of Hilbert spaces to be the system types.

In [3] the authors introduce a couple of equations for bending around wires called the **yanking equations**. One of them will be used later on in this report, so it will be introduced here.

We can easily check that this is true by decomposing the wires to triangles as follows.

A **linear map** is given by the following diagram.

We can find the equivalent form in Dirac notation as follows.

In the context of a bra and a ket, we can translate the diagrammatic linear map to Dirac notation as such.

Finally, maps can have an arbitrary number (n) of in- and outputs.

$$\stackrel{(9)}{=} \sum_{\substack{j_0 j_1 \dots j_n \\ i_0 i_1 \dots i_n}} r_{j_0 j_1 \dots j_n} \langle \phi_0 | j_0 \rangle \langle \phi_1 | j_1 \rangle \dots \langle \phi_n | j_n \rangle \langle i_0 | \psi_0 \rangle \langle i_1 | \psi_1 \rangle \dots \langle i_n | \psi_n \rangle$$

A **spider** is a special map which functions as a Kronecker delta. It forces the inputs to be the same as the outputs. In the case where we have one input and one output this gives a trivial result.

$\bigcirc \equiv \sum_i \left(\begin{array}{c} \triangleup_i \\ \triangleleft_i \end{array} \right) \equiv \text{vertical line} \quad (8)$

In the case where there are multiple in- and outputs it forces all to be the same.

The diagram shows two equivalent representations of a summation operation. On the left, a single node (a circle with a dot) has m inputs at the bottom and n outputs at the top. This is shown to be equivalent (\equiv) to a structure on the right. The right structure consists of a summation symbol \sum_i followed by a parallel arrangement of summation nodes. Each node is a triangle with an 'i' inside, having one input and one output. There are m inputs at the bottom and n outputs at the top, with ellipses indicating multiple parallel paths.

By applying arbitrary in- and outputs we can see how the **Kronecker delta** arises from Dirac notation. The following example is for two two inputs and two outputs but the same principle extends to an arbitrary number of in- and outputs.

$$\begin{array}{c}
\triangle j \quad \triangle k \\
\quad \backslash \quad / \\
\quad \circ \\
\quad / \quad \backslash \\
\triangle l \quad \triangle m
\end{array}
\stackrel{(19)}{=} \sum_i \begin{array}{c} \triangle j \quad \triangle k \\ \downarrow \quad \downarrow \\ \triangle i \quad \triangle i \\ \downarrow \quad \downarrow \\ \triangle l \quad \triangle m \end{array}
\stackrel{(9)}{=} \sum_i \langle j|i \rangle \langle k|i \rangle \langle i|l \rangle \langle i|m \rangle = \delta_{jk} \delta_{lm} \delta_{jl}$$

Spiders with single in- or outputs also exist. A spider with a single output creates a **random classical bit**.

$$\bigcirc \stackrel{(19)}{=} \sum_i \begin{array}{c} | \\ \triangle_i \end{array} = |0\rangle + |1\rangle \quad (21)$$

And a spider with a single input **deletes** any classical variable $j \in \{0, 1\}$.

$$\begin{array}{c} \bigcirc \\ | \\ \triangle_j \end{array} \stackrel{(19)}{=} \begin{array}{c} \triangle_i \\ | \\ \triangle_j \end{array} \stackrel{(9)}{=} \langle 0|j\rangle + \langle 1|j\rangle = 1 \quad (22)$$

An important property of spiders is that they fuse.

$$\begin{array}{c} n \text{ outputs} \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ m \text{ inputs} \end{array} = \begin{array}{c} n \text{ outputs} \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ m \text{ inputs} \end{array} \quad (23)$$

Which can be derived from equation (19).

We can also have spiders without any inputs or outputs. These turn out to just give a number corresponding to the dimension of the Hilbert space.

$$\bigcirc \stackrel{(23)}{=} \begin{array}{c} \bigcirc \\ | \\ \bigcirc \end{array} \stackrel{(19)}{=} \sum_i \begin{array}{c} \triangle_i \\ | \\ \triangle_i \end{array} \stackrel{(9)}{=} 2 \quad (24)$$

2.1.7 Phase Spiders

Phase spiders follow the same rules as normal spiders. However, perhaps unsurprisingly, they carry a **phase**. This phase is subject to a new set of rules which is evident from the definition of the phase spider.

$$\begin{array}{c} n \text{ outputs} \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ m \text{ inputs} \end{array} \stackrel{(25)}{=} \sum_j e^{i\alpha_j} \begin{array}{c} n \text{ outputs} \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ m \text{ inputs} \end{array} = \begin{array}{c} n \text{ outputs} \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ m \text{ inputs} \end{array} + e^{i\alpha} \begin{array}{c} n \text{ outputs} \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ m \text{ inputs} \end{array} \quad (25)$$

With $\alpha_0 = 0$ and $\alpha_1 = \alpha$.

When two phase spiders meet, they fuse like normal spiders and their phases add.

$$\begin{array}{c} n \text{ outputs} \\ \vdots \\ \textcircled{\alpha} \quad \textcircled{\beta} \\ \vdots \\ m \text{ inputs} \end{array} \stackrel{(25)}{=} \sum_{jk} e^{i\alpha_j} \begin{array}{c} n \text{ outputs} \\ \vdots \\ \textcircled{} \quad \textcircled{} \\ \vdots \\ m \text{ inputs} \end{array} e^{i\beta_k} \stackrel{(23)}{=} \sum_j e^{i(\alpha_j + \beta_j)} \begin{array}{c} n \text{ outputs} \\ \vdots \\ \textcircled{} \\ \vdots \\ m \text{ inputs} \end{array} \stackrel{(25)}{=} \begin{array}{c} n \text{ outputs} \\ \vdots \\ \textcircled{\alpha + \beta} \\ \vdots \\ m \text{ inputs} \end{array} \quad (26)$$

With j and k the indices of the triangles that constitute the spiders.

Furthermore, note that since complex phases get flipped in conjugate transformations, the conjugate of a phase spider is that same spider with an inverted phase.

$$@^* \stackrel{(25)}{=} \sum_j (e^{i\alpha_j})^* = \sum_j e^{-i\alpha_j} \stackrel{(25)}{=} @_{\alpha} \quad (27)$$

2.1.8 Colors and Bases

In the diagrammatic notation, the **color** of an object such as a spider or a triangle determines its basis. For now, we will define two orthonormal **bases**, the Z and X bases. Later on, as more bases will be necessary they will be introduced accordingly. The Z basis has white diagrammatic elements and for the X basis they are gray. The following is an example of how to translate between bases in this notation.

$$\begin{array}{c} | \\ \text{gray triangle} \\ | \end{array} = \frac{1}{\sqrt{2}} \left(\begin{array}{c} | \\ \text{white triangle} \\ | \end{array} + \begin{array}{c} | \\ \text{white triangle} \\ | \end{array} \right) \quad (28)$$

$$\begin{array}{c} | \\ \text{gray triangle} \\ | \end{array} = \frac{1}{\sqrt{2}} \left(\begin{array}{c} | \\ \text{white triangle} \\ | \end{array} - \begin{array}{c} | \\ \text{white triangle} \\ | \end{array} \right) \quad (29)$$

This equation is of course entirely analogous to its Dirac notation counterpart, with $|0\rangle_x$ and $|1\rangle_x$ being the orthonormal basis states in the X basis and $|0\rangle$ and $|1\rangle$ the orthonormal basis states in the Z basis.

$$|0\rangle_x = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \quad (30)$$

$$|1\rangle_x = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \quad (31)$$

All of the rules derived up to this point count also for the gray basis, or for any basis with two orthonormal basis vectors for that matter. We also have gray spiders, for example, with the same set of rules as the white spiders. With that being said, for the cases where two diagrammatic components of different bases meet we have to develop new rules, which we will do in due course.

Some diagrammatic components get no color, such as maps. What a map actually does to some input state is independent of the basis that it was originally defined in. Of course, we could alter the map from equation (15) to have gray triangles rather than white, but this would also imply a change in r_{ij} from that same equation such that the mapping would be preserved, independent of the choice of input basis. The same goes for triangles that are not basis vectors, and thus the ones

that are on their side. They can represent arbitrary vectors in their respective Hilbert spaces, and can generally be described as superpositions in various bases. There is no obvious preference for which basis to then describe this arbitrary vector in.

2.1.9 Basis and phase translations on the Bloch sphere

In section 2.1.8 we saw how to translate between different bases diagrammatically. The approach was to translate Dirac bra's and kets directly to the appropriate triangles. Using phase spiders, we can employ a more purely diagrammatic method. This stems from the fact that applying a phase to a state corresponds to a rotation on the **Bloch sphere**. Let's see where a spider without a phase lies on the Bloch sphere.

$$\textcircled{\circ} \stackrel{21}{=} \sum_i \text{triangle}_{\downarrow i} = \text{triangle}_{\downarrow 0} + \text{triangle}_{\downarrow 1} \approx \frac{1}{\sqrt{2}} (\text{triangle}_{\downarrow 0} + \text{triangle}_{\downarrow 1}) \stackrel{28}{=} \text{triangle}_{\downarrow 0} \quad (32)$$

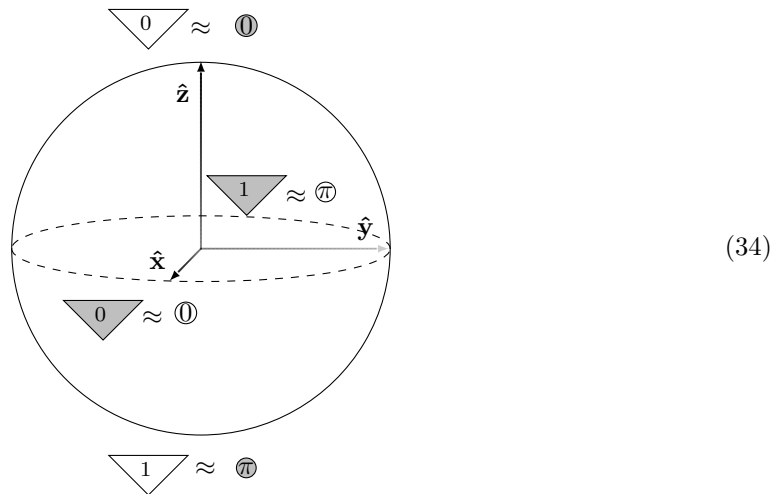
On the Bloch sphere in equation 34 we indeed see that the phaseless white spider is on the same location as the gray state 0.

For a white spider with a phase of π we get the following.

$$\textcircled{\pi} \stackrel{21}{=} \sum_i e^{i\alpha_i} \text{triangle}_{\downarrow i} = e^0 \text{triangle}_{\downarrow 0} + e^{i\pi} \text{triangle}_{\downarrow 1} \approx \frac{1}{\sqrt{2}} (\text{triangle}_{\downarrow 0} - \text{triangle}_{\downarrow 1}) \stackrel{29}{=} \text{triangle}_{\downarrow 1} \quad (33)$$

So a white spider with π phase corresponds to the gray state 1.

The gray spiders and white basis have a similar relationship and these are included as well in equation 34.



Where the gray triangles with a 0 and 1 are respectively at the positive and negative ends of the x axis on the Bloch sphere.

2.1.10 The xor map

The **Exclusive OR (xor) map** takes two inputs and gives an output of 0 when the inputs are the same and 1 when they are different. Mathematically, if we take $x, y \in \{0, 1\}$ as inputs the output of the xor map is $(x + y) \bmod 2$, or more commonly written as $x \oplus y$. We could just define a diagrammatic map and give it this property. However, there is another way of creating the xor map diagrammatically which only uses spiders. This has the advantage of allowing us to use spider rules to move xor maps around in diagrams.

If we enter two outputs of a certain basis into a spider that is of another basis orthonormal to the inputs this gives the xor map. To see how this works, let's use a gray spider and apply white test in- and outputs. First, recall that:

$$\begin{array}{c} \triangleup \\ |i\rangle \\ \downarrow \\ \triangle \\ 0 \end{array} \stackrel{9}{=} \langle 0 |_x |i\rangle = 1/\sqrt{2} \qquad \begin{array}{c} \triangleup \\ |i\rangle \\ \downarrow \\ \triangle \\ 1 \end{array} \stackrel{9}{=} \langle 1 |_x |i\rangle = (-1)^i/\sqrt{2} \quad (35)$$

Now applying the test in- and outputs to the proposed xor map [3].

$$\begin{array}{c} \triangleup \\ |k\rangle \\ \downarrow \\ \bigcirc \\ \swarrow \quad \searrow \\ \triangle \quad \triangle \\ |i\rangle \quad |j\rangle \end{array} \stackrel{19}{=} \begin{array}{c} \triangleup \\ |k\rangle \\ \downarrow \\ \triangle \\ 0 \end{array} \begin{array}{c} \triangleup \\ |k\rangle \\ \downarrow \\ \triangle \\ 0 \end{array} + \begin{array}{c} \triangleup \\ |k\rangle \\ \downarrow \\ \triangle \\ 1 \end{array} \begin{array}{c} \triangleup \\ |k\rangle \\ \downarrow \\ \triangle \\ 1 \end{array} \stackrel{35}{=} \frac{1}{\sqrt{2}} * \frac{1}{2} (1 + (-1)^{i+j+k}) \quad (36)$$

This equation results in 0 whenever the sum $i + j + k$ is uneven and $1/\sqrt{2}$ when it is even. Now $i + j + k$ is only even whenever $i \oplus j = k$. This thus, up to a factor of $\sqrt{2}$, gives the xor map.

2.1.11 Doubling

Doubling is the operation of horizontally composing the conjugate version of a diagram with itself. We already defined previously that having multiple components in the same diagram can be interpreted as the tensor product over those components. Therefore, doubling a diagram is equivalent in Dirac notation to taking the tensor product with its conjugate diagram. In the usual case, where the system type of a single wire is a Hilbert space (\mathcal{H}), the doubled wire has the set of density matrices on that Hilbert space as system type.

$$\mathcal{H} \otimes \mathcal{H} \cong \mathcal{D}(\mathcal{H}) \quad \therefore \quad \left| \begin{array}{c} \mathcal{H} \\ \mathcal{H} \end{array} \right| \cong \left| \begin{array}{c} \mathcal{H} \\ \mathcal{D}(\mathcal{H}) \end{array} \right| \quad (37)$$

The \cong denotes an isomorphism between the left and right hand sides. This is not an exact equality, but rather states that although the elements of the objects on both sides may be different, the structures of the objects themselves are the same. Intuitively they are the same object after one were to simply rename the elements of either one of the two. For all intents and purposes of this report we can assume \cong to refer to equality. Furthermore $\mathcal{D}(\mathcal{H})$ is the set of density matrices on \mathcal{H} and the letters next to the wires represent their system types.

Doubled maps and and states represent **pure** maps and states and single maps and states represent their **mixed** counterparts. Remember that pure states are those that can be described

by a single bra or ket, and can be any superposition of orthonormal basis states. A mixed state, on the other hand, is a statistical ensemble of pure states described by a classical probability distribution over those pure states. The most intuitive interpretation, and how it will mostly be used in this report, is that single diagrammatic components are used in the classical domain whereas doubled components are used in the quantum domain. A doubled state, for example, represents the preparation of a quantum state while a single state represents the preparation of classical information on a quantum state. Single wires then carry this classical information and doubled wires carry quantum information. This has one interesting implication: In the diagrammatic notation, classical information is represented in terms of a Hilbert space and thus also in an associated basis. In other words, a single wire carries *classical information on a bit encoded into a certain basis*, rather than just a *classical bit*. To some extent this is intuitive. Say we measured a qubit in a certain basis and received some classical information from this measurement. This classical information does not tell us much about the qubit if we do not know in what basis it was measured in in the first place.

Doubling a state is nothing more than placing a second conjugate state next to the original state. The doubled state itself is then drawn thick, this is represented as follows.

$$\begin{array}{c} \mathcal{H} \quad \mathcal{H} \\ \diagdown \quad \diagup \\ \psi \quad \psi \end{array} \equiv \begin{array}{c} \mathcal{D}(\mathcal{H}) \\ \diagdown \quad \diagup \\ \psi \end{array} \quad (38)$$

The notation of a state with a hat, such as $\hat{\psi}$, means that that state is pure.

The diagrammatic notation thus allows for a very similar representation of pure and mixed states; the only visual difference being that pure states have thick lines whereas mixed states have thin lines. In Dirac notation the differences between pure and mixed states stand out more, where mixed states are represented mathematically as classical probability distributions over pure states.

We can define an arbitrary **density matrix** as follows.

$$\sum_{ij} \rho_{ij} \begin{array}{c} | \\ \diagdown \quad \diagup \\ i \quad j \end{array} \equiv \begin{array}{c} | \\ \diagdown \quad \diagup \\ \rho \end{array} \quad (39)$$

Remember that in Dirac notation, an arbitrary density matrix ρ is mathematically structured as an operator.

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i| \quad (40)$$

In this regard, the diagrammatic notation provides a more intuitive representation of states as density matrices. In this notation density matrices are visualized as a special form of state preparation, whereas in Dirac notation they are mathematically structured as operators.

Note that in the case of doubled states, effects, or maps with multiple in- or outputs we need to make sure that the single wires are grouped correctly.

$$(41)$$

The notation of a map with a hat, such as \hat{f} , means that that map is pure.

Doubled spiders among themselves follow the same rules as normal spiders. That is, they follow the fusion rule from equation 23. However, when a single and double spider meet, they fuse to form one single spider with potentially doubled output wires, a so called bastard spider.

$$(42)$$

Which follows from the fact that doubled spiders are two single spiders that each fuse according to equation (23).

2.1.12 Discarding

Discarding is the process of removing part of a diagram from the whole, or removing the whole diagram altogether. It is the quantum equivalent of deleting for classical diagrams. Discarding is defined as follows.

$$(43)$$

It is trivial to see that applying discarding to any arbitrary (normalized) state always results in the number 1. In fact, discarding a state or map is equivalent to taking its **trace**. As an example, we discard an arbitrary normalized density matrix, ρ .

$$(44)$$

Discarding is not a pure map. It connects the two counterparts of a doubled state by a single wire. This allows for discarding to be used to purify any arbitrary state or map. For the case of a map Φ , **purification** is as follows.

$$\Phi = \hat{f} \stackrel{(41)}{=} f \otimes f \quad (45)$$

Note that purification is not only a diagrammatic result, but that it is a commonly used phenomenon in quantum information theory. In general, purification refers to the fact that any mixed state is equivalent to a pure state in a higher dimensional Hilbert space of which we trace a part away. In other words, every mixed state can be viewed as some pure state with part of its Hilbert space traced away.¹ This is essentially also what we see happen in equation (45). Φ is some (potentially mixed) quantum map and it is equal to \hat{f} with one output traced away. If we rewrite this \hat{f} and its discarded output in terms of single components we receive the right hand side, which we see is indeed not completely pure; it cannot be written as a diagram with its conjugate next to it.

Taking the conjugate of the discarding map is actually the preparation of the **fully mixed state** up to some constant.

$$\mathbb{1} \stackrel{(38)}{=} \sum_i \triangle_i \stackrel{(10)}{=} \sum_i \triangle_i \triangle_i \approx \sum_i \frac{1}{2} |i\rangle \otimes |i\rangle \quad (46)$$

2.1.13 Entanglement

Pure **entangled states** are those states that are not horizontally separable.

$$\psi \neq \psi_1 \otimes \psi_2 \quad (47)$$

Since horizontally composed states form the tensor product of those states diagrammatically, this definition is in line with theory, where entangled states are defined as those states that can not be written as the tensor product of two states [12]. An example of such a state is the following.

$$\text{curved line} \stackrel{(10)}{=} \sum_i \triangle_i \triangle_i \stackrel{(1)}{=} |00\rangle + |11\rangle \quad (48)$$

Although it may seem like this state separates since it is made up of two triangles, it does not. Both triangles are correlated through the same index. Indeed, if the indices were different for each of the triangles this would not be an entangled state.

2.2 Advanced diagrammatic concepts

2.2.1 Encoding and Decoding

To pass from classical information to quantum information a process called **encoding** is employed. Encoding puts the information from some classical bit into a quantum system, a qubit. There exist various physical systems that can be used for the realization of qubits. Later, in section 3.4 for example, we will see how a photon can be used as a qubit. After we are done with it, we

¹A full proof of this statement can be found in [12], page 110.

need a way to extract the quantum information stored in this qubit back into classical information. This process is the inverse of encoding and has been given its own name, decoding. Although the concept of measuring is much more broad than just decoding, we will sometimes use these words interchangeably throughout the rest of the report.

Diagrammatically, the encoding map is a single spider with one doubled wire as output and one single wire as input. The color of the spider is the basis in which the classical bit gets encoded. For example, encoding classical information p on a diagonal density matrix $\rho = \sum_i p_i |i\rangle \langle i|$ into a quantum state in the white basis with indices $i \in \{0, 1\}^n$.

$$\begin{array}{c} \text{Diagram 1} \end{array} \stackrel{38}{=} \begin{array}{c} \text{Diagram 2} \end{array} \stackrel{19}{=} \sum_i \begin{array}{c} \text{Diagram 3} \end{array} \stackrel{38}{=} \sum_i \begin{array}{c} \text{Diagram 4} \end{array} \stackrel{9}{=} \sum_i \langle p|i \rangle \begin{array}{c} \text{Diagram 5} \end{array} = \begin{array}{c} \text{Diagram 6} \end{array} \quad (49)$$

The decoding map is the adjoint of the encoding map. Its behavior follows directly from (49).

Phases get lost when decoding quantum states to classical states. We can rely on our knowledge of phase spiders to see how the phase gets eliminated in the process of **decoding** a quantum state with a phase α to classical information.

$$\begin{array}{c} \text{Diagram 1} \end{array} \stackrel{27}{=} \begin{array}{c} \text{Diagram 2} \end{array} \stackrel{25}{=} \sum_{jk} e^{-i\alpha_j} \begin{array}{c} \text{Diagram 3} \end{array} e^{i\alpha_k} \stackrel{26}{=} \sum_j e^{i(\alpha_j - \alpha_j)} \begin{array}{c} \text{Diagram 4} \end{array} \stackrel{26}{=} \begin{array}{c} \text{Diagram 5} \end{array} = \begin{array}{c} \text{Diagram 6} \end{array} \quad (50)$$

With j and k the indices of the triangles that constitute the spiders.

2.2.2 Complementary spiders

Two bases are said to be **complementary** if encoding in one and decoding in the other always gives a random measurement result. The two previously introduced bases X and Z are complementary. Alice encoding a classical state into a qubit in either one of these bases and then Bob measuring that qubit in the other basis results in Bob receiving a random bit.

$$\begin{array}{c} \text{Diagram 1} \end{array} = \frac{1}{2} \begin{array}{c} \text{Diagram 2} \end{array} \quad \begin{array}{c} \text{Diagram 3} \end{array} = \frac{1}{2} \begin{array}{c} \text{Diagram 4} \end{array} \quad (51)$$

Where the spider with a single output denotes the preparation of a random bit by equation (21) and the spider with a single input denotes the deletion of the incoming state by equation (22).

On the contrary, whenever Alice and Bob do respectively encode and measure in the same basis, they achieve successful communication.

$$\begin{array}{c} \text{Diagram 1} \end{array} = \begin{array}{c} \text{Diagram 2} \end{array} = \begin{array}{c} \text{Diagram 3} \end{array} \quad (52)$$

2.2.3 The essential uniqueness of purification

The **essential uniqueness of purification** is a theorem that states the following: Given two pure maps \hat{V}_1 and \hat{V}_2 that satisfy $V_1, V_2: \mathcal{H} \rightarrow \mathcal{L} \otimes \mathcal{K}$ and:


(53)

Then there exists some unitary $U: \mathcal{L} \rightarrow \mathcal{L}$ such that:


(54)

Intuitively, this theorem states that if \hat{V}_1 and \hat{V}_2 are similar to the extent that they satisfy equation (53), then the linear maps V_1 and V_2 can only differ by some unitary U . This is an adaptation of the theorem given by Kissinger and Westerbeaen in section 2.1 of [6].

Note that whereas here \hat{V}_1 and \hat{V}_2 have two output wires that are discarded, this theorem scales for maps with an arbitrary number of outputs. In other words, \mathcal{L} could be the Hilbert space of an arbitrarily large composite system.

2.2.4 The $k - k'$ commute rule

In this report we will on multiple occasions want to move certain phase spiders past one another. This does, however, require some more advanced diagrammatic derivations that are too extensive for the scope of this report. Therefore we show the result here, and refer to the relevant literature. In [3] on page 771, Coecke and Kissinger introduce the $k - k'$ **commute rule** and give the full diagrammatic derivation to arrive to this rule. As we will later see, this rule is the main ingredient for pushing Pauli's past one another diagrammatically. It states that for $k, k' \in \{0, \pi\}$:


(55)

This rule shows us how we can move X and Z phase spiders of a different color with either a π or no phase past one another. Note that it is important that the spiders are in the X and Z bases, and that this rule doesn't hold for just any two bases. In fact, the requirement is that the bases be strongly complementary; a concept that would require a lengthy explanation if it were to be introduced in this report, but is extensively explained in section 9.3 of [3].

3 Protocols

3.1 One Time Pad

The classical One Time Pad (OTP) is perhaps the oldest and most well known cryptographic method. We will first discuss this protocol as a setup to the quantum version, the quantum one time pad. Although originally proposed in 1884 [10], the OTP was first formalized and proven secure in 1949 [14]. Since then, the scheme has essentially remained the same due to its simplicity. A typical OTP goes as follows.

Alice and Bob start with a uniformly random key $k \in \{0, 1\}^n$. Alice wants to send some message $m \in \{0, 1\}^n$ to Bob. She encrypts m using the key k by means of an xor map, generating ciphertext $x^n = m \oplus k$.² After this step, she sends x to Bob who applies k again to x to receive m since $k \oplus x = m$. The assumption on the eavesdropper (Eve) in this scheme is that she doesn't know k and can intercept x . With that being said, if k was truly uniformly random, x is so too, leaving Eve with a random variable even after a successful interception. The main drawback of this scheme is the fact that the key has to be the same size as the message and that this key can only safely be used once.

3.1.1 Diagrammatic Representation of the One Time Pad

The OTP relies on classical communication channels and xor maps. Since we can represent both of these diagrammatically we can draw out the fully classical OTP protocol by means of diagrams. Recall from section 2.1.6 that the following state generates a random bit.

$$\begin{array}{c} | \\ \circ \end{array} \quad (56)$$

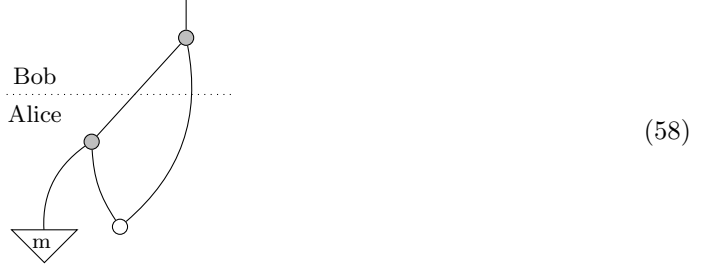
We choose this as the uniformly random key k and give one copy to Alice and one to Bob. Alice, who starts with message m , then applies the xor gate from section 2.1.10 to this message, generating ciphertext x .



Note that the dotted line used here is only to illustrate what part of the diagram Alice and Bob "control" and that it bears no physical significance. Therefore, in diagram equations we can also omit the dotted line as need be. We will use this notation at multiple points throughout the thesis.

Everything Bob now has to do is to apply an xor gate to the ciphertext x and the key k .

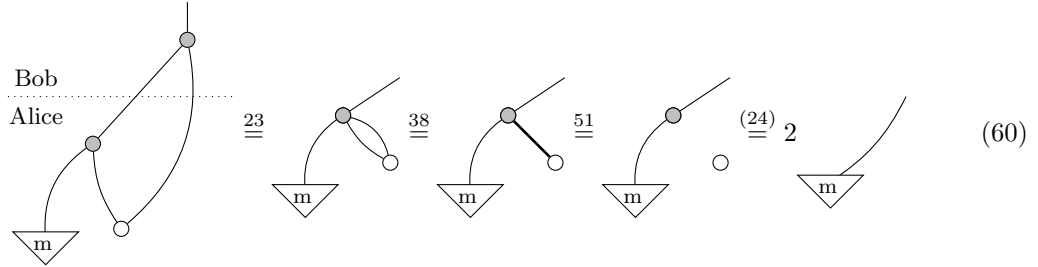
²In different realizations of the OTP, Alice has different methods of encrypting her message. With that being said, applying an xor map to the key and the message is one of the most common versions.



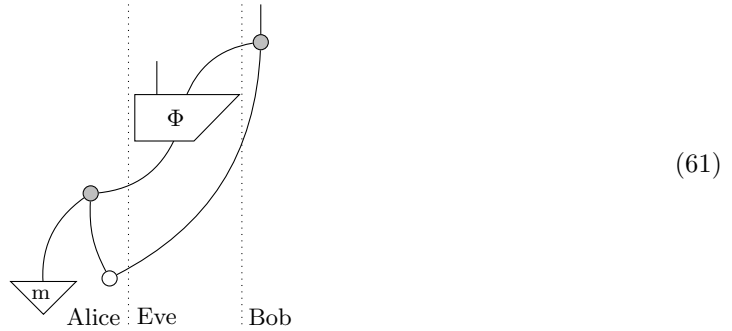
To check that this is indeed a viable communication protocol we want equation (58) to reduce to:



This happens up to a constant factor.



This is a nice result. We see that without any interference from the environment or a third party Alice and Bob achieve uninterrupted communication. But what if there is some eavesdropper that intercepts the communication? We introduce Eve. She receives the ciphertext that is being sent from Alice to Bob, applies some map Φ to it, and subsequently saves one output and sends another to Bob. Note that since the key k is a shared secret between Alice and Bob she doesn't learn about it. Diagrammatically, this situation is as follows.



Of course, we want Eve to learn nothing about the message m . To see the situation from her perspective we can delete Bob's output.

$$(62)$$

Using the translations between triangles and spiders from equation (34) and some spider fusion (23) brings us to:

$$(63)$$

By equation (36) we know that the two triangles that represent information on m and k get xor'd by the gray spider to give x .

$$(64)$$

What Eve thus learns in this situation is only the ciphertext, x . This was also what we expected beforehand. Furthermore, since to her this x is random, she does not learn anything on m , meaning that the message is secure.

3.2 Quantum One Time Pad

In the Quantum One Time Pad (QOTP) Alice and Bob share a uniformly random key $\alpha \in \{0, 1, 2, 3\}^n$ and want to communicate some quantum state, ρ . Alice first transforms ρ by applying to it some Pauli map, σ_α , which are written out for all α in appendix B. After she has

prepared this state she sends it to Bob over a quantum channel. He can then apply the Hermitian adjoint of this Pauli resulting in the original ρ sent by Alice. This step exploits the unitary property of the Pauli maps. Eve can observe the state sent from Alice to Bob. However, since she does not know α , from her perspective this is the fully mixed state. Mathematically, we can see as follows that encrypting any qubit state by two random (but the same) Pauli's results in the fully mixed state.

$$\rho' = \sum_{\alpha=0}^3 \frac{1}{4} \sigma_{\alpha} \rho \sigma_{\alpha}^{\dagger} = \frac{1}{2} \mathbb{I} \quad (65)$$

Where \mathbb{I} refers to the two by two identity matrix, and $\frac{1}{2}\mathbb{I}$ is the density matrix of the fully mixed state.

In principle, the QOTP is similar to the OTP. By means of encrypting a piece of information with a random key which is unknown to Eve but shared by Alice and Bob the latter can achieve secure communication. As a communication protocol, QOTP is interesting since it allows for the transmission of a continuum object by a 2-bit key. In other words, any state vector on the entire Bloch sphere can be communicated as a qubit by two bits of classical information. In the classical OTP, on the other hand, one bit is used as a key to communicate each bit of classical information.

3.2.1 Diagrammatic Representation of the Quantum One Time Pad

In the diagrammatic formalism we can construct a set of maps that have the same property as the Pauli maps in equation 65, the Bell maps. These are defined in terms of the Pauli matrices as follows [3].

$$\sigma_0 = B_0 \quad \sigma_1 = B_1 \quad \sigma_2 = iB_3 \quad \sigma_3 = B_2 \quad (66)$$

In appendix B, these Bell matrices are written out. Note that since they have the same properties as Pauli matrices we use them interchangeably throughout the report.

The Bell maps have the interesting property that we need just two of them to create all four. For random $u, w \in \{0, 1\}$ and $\alpha \in \{0, 1, 2, 3\}$ dependent on u and w we can write this mathematically as follows.

$$B_2^u B_1^w = B_{\alpha} \text{ (where } \alpha = u + 2w) \quad (67)$$

We thus see that applying these two Bell maps randomly is equivalent to applying a random Bell map in general. In order to form diagrams for these Bell maps we have to understand them a little bit better. One of their properties is that they correspond to rotations on the Bloch sphere around the z and x axes [5], also known as bitflips around these axes. B_1^1 refers to a bitflip in the x basis and a π rad rotation around z axis whereas B_2^1 refers to a bitflip in the z basis and a π rad rotation around the x axis. Diagrammatically, we know from section 2.1.9 that we can represent these rotations on the Bloch sphere by means of phase spiders.

$$B_1^1 \Leftrightarrow \text{⊗} \quad B_2^1 \Leftrightarrow \text{⊙} \quad (68)$$

The maps B_1^0 and B_2^0 should be entirely trivial. Any map to the power of 0 is the identity map. With that being said, it is still good to write them out diagrammatically since we will see that we have the freedom to make a slight distinction between them that will turn out to be useful later.

$$B_1^0 \Leftrightarrow \text{diagram} = \text{diagram} \Leftrightarrow B_2^0 \quad (69)$$

Let's send the 0 state through in the gray (x) basis to test if the B_1^0 map behaves as expected.

$$B_1^0 |0\rangle_x \Leftrightarrow \text{diagram} \stackrel{28}{=} \frac{1}{2} \text{diagram} + \frac{1}{2} \text{diagram} \stackrel{21}{=} \frac{1}{2} \text{diagram} \stackrel{23}{=} \frac{1}{2} \text{diagram} \stackrel{34}{=} \text{diagram} \Leftrightarrow |0\rangle_x \quad (70)$$

This map does seem to behave appropriately. We see that whenever we send a gray 0 through, we get a gray 0 as output and thus it doesn't flip the input state in the x basis. Let's test if the B_1^1 works as expected too.

$$B_1^1 |0\rangle_x \Leftrightarrow \text{diagram} \stackrel{29}{=} \frac{1}{2} \text{diagram} + \frac{1}{2} \text{diagram} \stackrel{21}{=} \frac{1}{2} \text{diagram} \stackrel{26}{=} \frac{1}{2} \text{diagram} \stackrel{34}{=} \text{diagram} \Leftrightarrow |1\rangle_x \quad (71)$$

These maps do exhibit the correct properties. Indeed, whenever we send a gray 1 through we also get the expected results. It seems that we have formed the B_1^0 and B_1^1 maps successfully. While we will not go through the full math here, the B_2^0 and B_2^1 maps follow very similar logic and also behave as expected.

Now that we have created our four constituent maps we are in principle ready to make any Bell map by equation (67). In the protocol, however, Alice and Bob choose a random Bell map to apply to their qubit states. We therefore still need to find a diagrammatic method to generate a random u and w on which our choice of Bell map depends. Recall from equation (21) that generating a random bit is the following diagram.

$$\text{diagram} \stackrel{21}{=} \sum_u \text{diagram} \quad (72)$$

Where we suggestively choose u as the index for the random classical bit.

We could encode this random classical bit into a qubit as follows.

$$\text{diagram} \stackrel{21}{=} \sum_u \text{diagram} \stackrel{49}{=} \sum_u \text{diagram} \quad (73)$$

We can use this qubit to control whether or not we apply a Bell map. It represents the encoded version of u . The previous two equations are entirely analogous for the gray (x) basis.³

We now have the necessary ingredients to make a map that applies a random bitflip or not in x and another map that makes a random bitflip or not in z. The results are as follows.

³In fact, we can make an even stronger statement about the diagrams in equation 73: They are actually equal to the analogous diagrams in the gray basis. Note that the rightmost diagram is the adjoint of discarding, which can be interpreted as preparing the fully mixed state, up to a number. Since this state is independent of the basis, these diagrams of white basis are equal to the analogous diagrams in the gray basis.


$$\text{for random w: } B_1^w \Leftrightarrow \left| \begin{array}{c} \bullet \\ \bullet \end{array} \right| \quad \text{for random u: } B_2^u \Leftrightarrow \left| \begin{array}{c} \circ \\ \circ \end{array} \right| \quad (74)$$

Where the gray spider in the B_1^w map determines the random choice of w and the white spider in B_2^u determines the random choice of u .

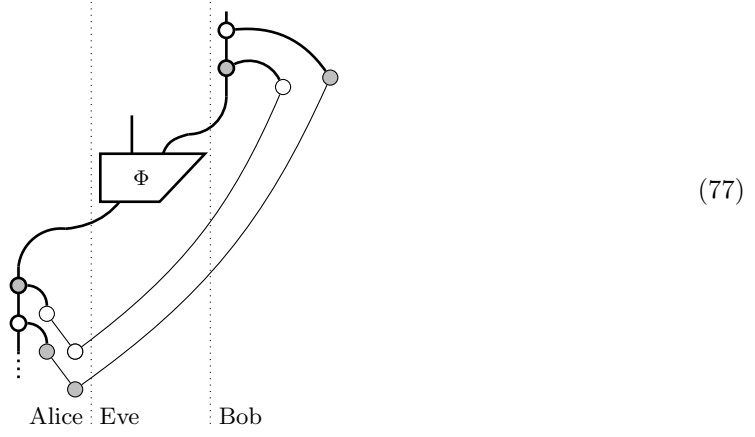
Let's confirm that the diagrammatic version of B_1^w does indeed behave as expected.

$$\begin{array}{c} \circ \\ | \\ \circ - \circ \\ | \\ \circ \end{array} \equiv_{21} \begin{array}{c} \circ \\ | \\ \circ - \circ \\ | \\ \triangle_0 \end{array} + \begin{array}{c} \circ \\ | \\ \circ - \circ \\ | \\ \triangle_1 \end{array} \equiv_{49} \begin{array}{c} \circ \\ | \\ \triangle_0 \end{array} + \begin{array}{c} \circ \\ | \\ \triangle_1 \end{array} \equiv_{34} \begin{array}{c} \circ \\ | \\ \circ \end{array} + \begin{array}{c} \circ \\ | \\ \circ - \circ \\ | \\ \circ \end{array} \equiv_{26} \begin{array}{c} \circ \\ | \\ \circ \end{array} + \begin{array}{c} \circ \\ | \\ \circ \end{array} \quad (75)$$

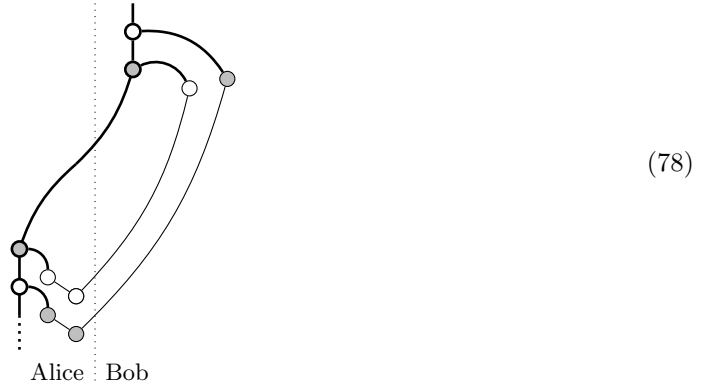
So depending on whether the random bit was a 0 or a 1, we get the map from equation 70 (B_1^0) or the map from equation 71 (B_1^1), which is what we needed. Again, this derivation is analogous for the map B_2^u . We have thus successfully made the constituents to our Bell maps. The random Bell map itself is then as follows.

for random α and $\alpha = u + 2w$: $B_\alpha \stackrel{67}{=} B_2^u B_1^w \Leftrightarrow$  (76)

To make a diagrammatic version of QOTP we need to compose the Hermitian adjoint of this Bell map to its output. The bottom Bell map can then be seen as Alice encrypting her state and the top Bell map can be seen as Bob decrypting this state. We also need to make sure that Alice and Bob make the same choice in α and thus use the same Bell map. This can be done by connecting the outputs of the spiders that generate the random numbers to the Bell maps of both Alice and Bob. Furthermore, whereas in the previously described protocol a state ρ is communicated as an example, here we can be more general. We allow the state that Alice wants to communicate to Bob to be any state in any basis. This is represented by a dotted line at the bottom of the diagram. Finally, we need to introduce Eve. She goes in between the Bell maps of Alice and Bob. She can be interpreted as an eavesdropper, but also more generally as the environment. In this sense, Eve could thus also just be a source of noise. She gets the map Φ that takes as input the state sent by Alice, and outputs one state that she keeps and another that is sent to Bob. In all, this makes QOTP to be the following diagram.



Let us first consider what would happen if there was no noise, and thus no Eve, in the quantum communication channels. Diagrammatically, this looks as follows.



For this protocol to be a good candidate for communication we need equation (78) to reduce to the following.



This simply states that Alice's qubit is sent to Bob through the identity map. We already know that this is true due to the unitarity of the Bell maps. Diagrammatically, the proof is as follows.

(80)

Which is up to a constant factor what we needed. Now that we have diagrammatically shown that QOTP without Eve is indeed a communication protocol, we can return to equation (77) to see what would happen if the protocol did include Eve. First of all, let's see what the result is of Alice encrypting her state by means of a Bell map. To understand what happens in this case diagrammatically we need to first extend equation (73). Note that the RHS of this equation is actually the Hermitian adjoint of discarding as it is defined in 2.1.12, which is in turn the fully mixed state. Since this state is independent of the choice of basis we can construct the following equation.

(81)

Using equation (81) we can prove that whenever Alice encrypts a state with a Bell map and deletes her classical u and w she outputs the fully mixed state, and discards the input.

(82)

This might give some intuition for what Eve would receive after Alice has encrypted her state. However, in the full protocol, Alice can not just delete her random u and w . Somehow she has to

communicate them with Bob. Remember from equation (77) the diagram of QOTP as a communication protocol and Eve with the ability to intercept the quantum state but not u or w . Let's see what happens when we look at this situation from Eve's perspective and thus trace away (discard) Bob's channel.

$$(83)$$

We could now use some of the rules that we have developed so far to propagate the discarding map further down. However, we could also realize that from Eve's perspective there is no difference between Bob applying random Bell maps to his qubit and then discarding it, and Bob discarding his state even before applying these Bell maps. Therefore, we can write:

$$(83) = \dots \stackrel{81}{=} \dots \stackrel{82}{=} \dots = \dots (84)$$

The third diagram from the left in equation (83) is up until Eve's map Φ the same as equation (82). We therefore see that from Eve's perspective Alice gives her the fully mixed state due to the Bell map that she applies to her original state. So even in the full protocol, the Bell map is responsible for encrypting the state such that Eve can extract no information from it.

3.3 Quantum Teleportation

In quantum teleportation, Alice and Bob want to communicate a qubit, ρ_1 , using the fact that they share an EPR pair, ρ_{23} , and a classical communication channel. Note that the subscripts in this section help keep track of the different particles present. Subscript 1 refers to Alice's particle, 2 to her side of the EPR pair and 3 to Bob's side of the EPR pair. Alice measures ρ_1 together with ρ_2 in a Bell measurement, entangling these two states to give the total entangled state of ρ_{12} . Since ρ_2 and ρ_3 were already entangled, this has as result that all three of the states are entangled. This projects ρ_3 to one of four possible pure states dependent on the result of the Bell measurement done by Alice. The four states actually being the 'QOTP-encrypted' versions of ρ_1 . In other words, ρ_3 is equal to ρ_1 with one of four possible Pauli's applied to it. Which Pauli this is is revealed by the outcome of the Bell measurement, the 'QOTP key'. Using the classical communication channel,

Alice can send this QOTP key to Bob in two classical bits. In order for Bob to extract ρ_1 from his part of the entangled state, ρ_3 , he needs to apply to it the hermitian adjoint of the Bell map that corresponds to this QOTP key.

The name of this protocol might lead one to think that it includes the teleportation of information. However, this is not the case. This protocol does not achieve faster than light communication. Although the projection of ρ_3 onto a pure state happens simultaneously with Alice making the Bell measurement, in order for Bob to receive any information on ρ_1 from the QOTP-encrypted ρ_3 he must apply the hermitian adjoint of the Bell map that corresponds to the QOTP key. Alice thus has to communicate classically to Bob the QOTP key.

Quantum teleportation carries some similarity to QOTP. In both protocols, Alice and Bob communicate two classical bits of information in order to communicate one qubit. Also, in both cases, Bob applies a Bell map. These similarities might give some intuition for the equivalence of these protocols. In the end of this section we will see this equivalence diagrammatically.

3.3.1 Diagrammatic Representation of Quantum Teleportation

The only component of this protocol that we have not worked out diagrammatically yet is the Bell measurement. We know that the Bell measurement is the adjoint of the Bell state. Therefore it suffices to create the Bell states and take their adjoints to make the corresponding Bell measurements.

Our approach to making these Bell states is by taking them apart and constructing their smaller constituents diagrammatically. The Bell states are actually composed by a Hadamard (H) and CNOT maps, applied to two qubits. The Hadamard map is applied first to one of the qubits. It transforms $|0\rangle$ to $(|0\rangle + |1\rangle)/\sqrt{2}$ and $|1\rangle$ to $(|0\rangle - |1\rangle)/\sqrt{2}$. These superposition states actually form the basis states of a different basis. Therefore, the Hadamard map can be seen as a basis transformation. For example, a Hadamard map applied to basis states in the z basis will transform those states to the x basis. The CNOT map is then applied to the two qubits. In the CNOT map, one input qubit controls whether or not the other is negated. If the former is $|0\rangle$ the latter undergoes the identity and if the former is $|1\rangle$ the latter is negated. The control qubit in the CNOT gate is in this case the output of the Hadamard map. The result of these operations combined entangles the two input qubits and puts them in one of four possible Bell states. Let's test this construction on input state $|00\rangle$ to see if it indeed behaves as expected:

1. The Hadamard map puts the first qubit in superposition: $|00\rangle \Rightarrow \frac{(|0\rangle + |1\rangle)|0\rangle}{\sqrt{2}}$
2. The CNOT negates the second qubit if the first is $|1\rangle$: $\frac{(|0\rangle + |1\rangle)|0\rangle}{\sqrt{2}} \Rightarrow \frac{|00\rangle + |11\rangle}{\sqrt{2}}$

The result is thus indeed one of the Bell states, namely B_0 .

Diagrammatically, we do not have to construct the whole Hadamard gate to put one of the qubits into the required superposition. In fact, remember that the Hadamard gate only changes the basis of its input state. Therefore, all we have to do is encode one qubit into a different basis than the other and choose one of the two bases to read the results in. The first part of our diagrammatic version of the Bell state thus becomes:

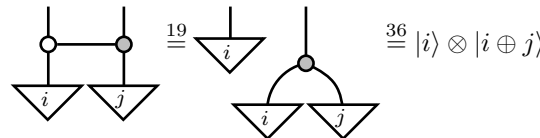

(85)

Where i and j are test inputs.

The CNOT gate itself is composed by a copy map and an XOR map. The first qubit is copied after which the second is XOR'd with the copied qubit [13]. Diagrammatically:


(86)

Applying test states:

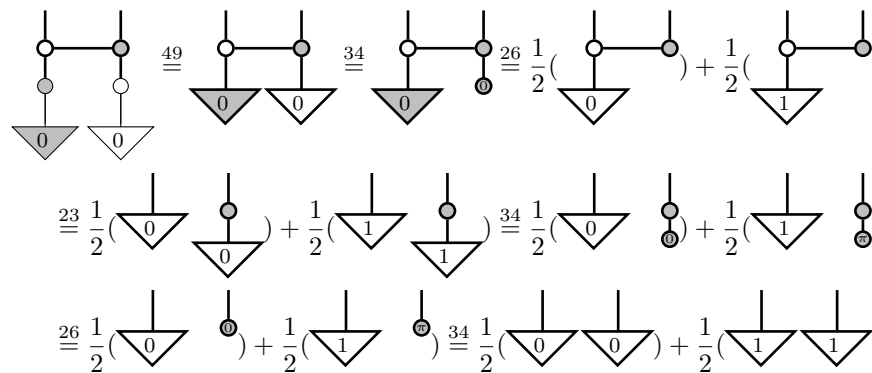

(87)

Which is indeed the behavior of the CNOT gate.

Composing equations (85) and (86) then gives the Bell state.


(88)

Let's again apply the state $|00\rangle$ to test the functionality of this Bell state preparation map.


(89)

The result is thus indeed the expected Bell state, B_0 . Note that whereas usually the constants in the Bell states are $1/\sqrt{2}$ here they are $1/2$ since the final state is doubled. In other words, each of the individual Bell states have a factor $1/\sqrt{2}$, but doubling them has also doubled this factor such that we receive $1/2$.

Let's choose the entangled state of equation (48) for state ρ_{23} . We can then compose the Bell measurement that Alice applies on the left of this entangled state and the Bell map that Bob applies to the right. Alice also inputs ρ_1 to the Bell measurement. Using a similar approach to the one we took in 3.2.1, we can generalize this to say that this state can in fact be any quantum state. Again, we can represent this by a dotted line which enters the diagram where ρ_1 would go. Finally, we need to make sure to connect the outputs of Alice's Bell measurement to Bob's Bell map such that Bob receives the QOTP key. The protocol for teleportation then becomes:

Alice | Bob

(90)

It is redundant to see what happens when we introduce an Eve or to show that this protocol allows for communication. This is because we can directly tell that this protocol is equivalent to the QOTP protocol described in section 3.2, which was shown to be secure in the noiseless case even when we introduce Eve and also allowed for communication. Placing QOTP without Eve and quantum teleportation side by side, this equivalence is strikingly simple. Everything we have to do is make a small adjustment to QOTP, writing it a bit more suggestively:

(78) =

$\stackrel{23}{=}$

(91)

3.4 Quantum Key Distribution

The first to develop a secure quantum key exchange protocol were Bennett and Brassard, who published an article on QKD in 1984. In order to understand their version of QKD, subsequently referred to as BB84, we first should understand the method of encoding information into quantum states employed in their protocol. For the previous three protocols, we did not care about how the information was encoded into quantum states since we only required Alice and Bob to communicate quantum states. In QKD and in the next section also for QKR, we do include the part where Alice and Bob actually encode classical information into a quantum state.

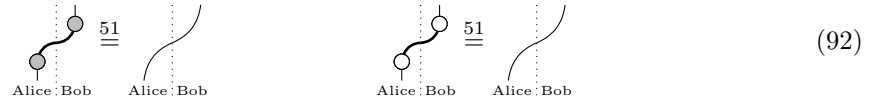
BB84 encoding uses an ubiquitous resource as information carrier, light. The constituents of light, photons, have the convenient property that their polarization forms a two level quantum system. The two orthonormal bases in this quantum system are diagonal (D) and rectilinear (R). In the rectilinear basis we can encode a 0 and a 1 as a photon polarized to 0 and 90 degrees respectively. In the diagonal basis we can do the same for 45 and 135 degrees. Physically encoding these qubits can then be done by means of a polarizer filter. This is a valid system of two mutually unbiased bases. Encoding any bit in one basis and then measuring in the other gives a random result. On the other hand, encoding a bit in one basis and measuring in the same gives the correct result.

In the BB84 protocol, Alice uses this method to encode a random bitstring $k \in \{0, 1\}^n$ into a series of qubits to be sent to Bob. She randomly chooses a set of bases to encode them in as well, $a \in \{R, D\}^n$. This creates n qubit states with each k_i encoded as a photon of a certain polarization in basis a_i . The set of qubit states, and thus the photons, that this results in are denoted as $|\Psi\rangle = \bigotimes_{i=1}^n |\psi_{k_i}^{a_i}\rangle$. She sends this $|\Psi\rangle$ to Bob who measures in a random basis $b \in \{R, D\}^n$. On a public channel, Bob sends b back to Alice who compares it with a . Alice then tells Bob where he measured in the correct basis, or for what i : $a_i = b_i$. If there would be no noise in the (quantum) communication channels and no Eve who disturbs the communication Bob should now have the k_i for all the qubits that he measured in the correct basis. Alice and Bob should then have some shared secret key consistent of the bitstring k_i where $a_i = b_i$. In the case where we do allow Eve to intercept the signal, Alice and Bob should execute one more step to estimate the amount of noise. For this step, Alice and Bob can use the fact that any attempt by Eve to measure the communicated quantum states disturbs them. Bob can send part of his result to Alice, who can compare it to the string she originally encoded. If these are the same, it is likely there was no Eve. If they are different, they can be sure that there was some Eve that somehow interfered with the quantum channel, either by attempting to intercept the qubits or by introducing noise. Finally, if Alice and Bob did achieve successful key exchange, they can safely use their k as future key for One Time Pad encryption (OTP).


3.4.1 Diagrammatic Representation of Quantum Key Distribution

QKD is the only protocol that has been diagrammatically (somewhat) thoroughly discussed in literature [6]. Therefore, we will not go over the full security proof here. This section is meant to get the reader to understand the diagrammatic version of QKD, such that we can use it later on in the context of equivalences.

To write it out diagrammatically we need to generalize it to a protocol that is not dependent on the physical realization of the qubits. We therefore say that Alice and Bob have the choice to encode their qubit in any two mutually unbiased bases, which we choose to be the z and x bases. Not considering Eve, if Alice and Bob then measure in the same basis, their quantum state is sent through just fine.



However, due to the fact that the bases are mutually unbiased we get the following situation when Alice and Bob's measurement bases do not agree.



These are the cases that Alice and Bob decide to not use upon discussing their choice of bases later in the protocol.

Now if we limit Eve's interference to only a measurement in either the z or x bases she will have a 50% chance of measuring in the correct basis. In the case where Alice and Bob both use the x basis Eve can thus measure correctly.

$$(94)$$

Or she measures in the wrong basis and both her and Bob receive a random bit.

$$(95)$$

This puts heavy constraints on Eve. Realistically, she could measure in different bases, or find even different ways to extract information from Alice's qubits. Therefore, it is more general to state that Eve can apply some map Φ , on which we can put constraints as needed.

$$(96)$$

This is exactly how Kissinger et al. approach this problem in [6]. They go on to prove security for the noiseless case; the case in which Alice and Bob detect no disturbance in the transmitted qubits. Since Eve represents the environment, and is therefore the cause of any potential noise, we can implement this statement diagrammatically as follows.

$$(97)$$

Tracing away Eve's output implements the statement that we look at the protocol from Alice and Bob's perspective. The fact that this is then equal to a single wire is representative of Alice and Bob seeing no noise. At various points throughout the rest of the report we will refer to the combination of these statements and equation (97) simply as 'the noiseless case'.

From here on Kissinger et al. in [6] include an extensive (fully diagrammatic) security proof which finally reaches the conclusion that for any Φ that satisfies 97 and mutually unbiased bases x and z there exists some ρ such that:

$$(98)$$

In words: Eve's channel has no connection to Alice and Bob's and she thus learns nothing about the state Alice sent to Bob.

3.5 Quantum Key Recycling with Eight State Encoding

The part of QKR in which Alice and Bob actually communicate a quantum state is very similar to the previously discussed protocols QKD and QOTP. The main differences between them are in the classical processing part and the encoding of the quantum states. This new method of encoding will be introduced soon. For now, we can assume Alice and Bob to use BB84 encoding.

In the QKR protocol introduced in [8], Alice and Bob both share knowledge beforehand about a set of bases $b \in \mathcal{B}^n$ and some one-time pad $j \in \{0,1\}^n$. Before Alice encodes her message $M \in \{0,1\}^m$ into a quantum state to be sent to Bob, she applies an error-correcting code $M \rightarrow E(M) \in \{0,1\}^n$ to it and xor's $E(M)$ with j to produce the payload $x \in \{0,1\}^n$. The error-correcting code allows Bob to determine and correct errors that may arise as a result of a noisy communication channel in the received payload at the cost of added redundancy [7]. The one-time pad is used to mask the message. After the payload is ready Alice prepares $|\Psi\rangle = \bigotimes_{i=1}^n |\psi_{x_i}^{b_i}\rangle$ and sends it to Bob. Bob, since he has the same set of bases should get the correct measurement result, x . Since he also knows j he can xor x and j to receive M , the message from Alice. In the case where there is no Eve detected he can use the same k and b to safely send a new payload back to Alice, who can then send one back to Bob, and so forth. However, if one of the two detects Eve's interference, both parties have to decide on a new key, which they can generate from previously uncompromised transmissions. Eve is allowed to attempt a measurement of the qubits transmitted between Alice and Bob, $|\Psi\rangle$. However, even if she would measure all of the qubits correctly, she only receives x . She would have to xor x with j to receive the message M but she does not know j .

Recently, it was proposed to use eight-state encoding in QKR protocols [4]. This method of encoding quantum states does not rely on the two mutually unbiased bases that we have seen up to this point, it essentially encodes bits into four different non mutually unbiased bases. Allowing each basis two states then results in the characteristic total of eight different states, hence the name of this encryption. The rest of the protocol then proceeds as before. In the paper in which this was originally brought forward [4], the authors suggest to place each of these states at equidistant locations on the Bloch sphere, corresponding to the eight corners of a cube if it were placed perfectly inside of the Bloch sphere. One of the bases is defined as having a 0 state $|\psi_0\rangle$ at $(1, 1, 1)^T/\sqrt{3}$ and a 1 state $|\psi_1\rangle$ at $(-1, -1, -1)^T/\sqrt{3}$. The others are defined as two Pauli transforms with respect to this basis, giving the following eight states.

$$|\psi_{uwg}\rangle \equiv B_1^w B_2^u |\psi_g\rangle \text{ where } u, w, g \in \{0,1\} \quad (99)$$

Where we used the Bell maps instead of the Pauli maps due to the fact that they have the same properties in this context.

More compactly, we can define $E_{uw} \equiv B_1^w B_2^u$ and have E_{uw} be our encryption operator. This gives rise to the representation of these eight states as given in figure 1.

3.5.1 Diagrammatic Representation of Quantum Key Recycling

We need to introduce some new diagrammatic notation for eight state encoding. After all, we need to move from a system of two bases to one of four. With that being said, a large part of it should be rather intuitive since we have essentially seen most of it already. The quantum one time pad is actually an eight state encryption protocol. The main difference is that in the protocol introduced for QKR, the eight states are distributed differently over the Bloch sphere. Diagrammatically, we can exploit the similarities by borrowing the notation that was previously introduced for the Bell

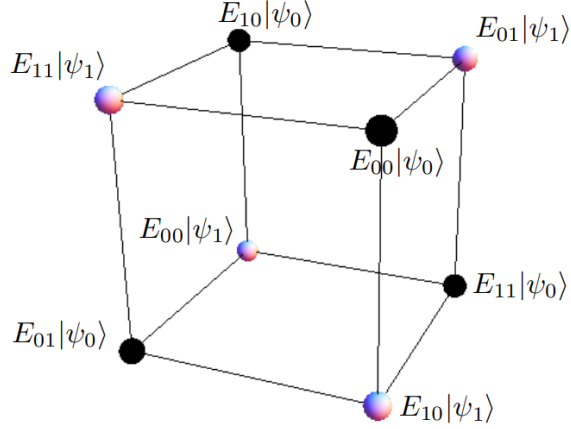


Figure 1: A cube representing the eight cipherstates $E_{uw} |\psi_g\rangle$ as its corner points $(\pm 1, \pm 1, \pm 1)^T / \sqrt{3}$ [4].

maps in the QOTP. According to equation (99), we only need to define the properties of one of the bases of eight state encoding and get the others for free by applying Bell maps to the states of this original basis. We choose the color red for this new basis. The red basis has the following two basis states: $E_{00} |\psi_0\rangle = \mathbb{I} |\psi_0\rangle = |\psi_{000}\rangle$ and $E_{00} |\psi_1\rangle = \mathbb{I} |\psi_1\rangle = |\psi_{001}\rangle$. We should first define triangles that represent classical information on these states, similarly to how we have classical triangles for the gray and white bases. We use the color red for this as well. This gives us the following diagrammatic components for the preparation of a state and the effect in this basis.

$$|\psi_{00g}\rangle \equiv \begin{array}{c} | \\ \text{red triangle down with } g \end{array} \quad \langle \psi_{00g}| \equiv \begin{array}{c} \text{red triangle up with } g \\ | \end{array} \quad (100)$$

Note that we have as well that.

$$\begin{array}{c} \text{red triangle up with } j \\ | \\ \text{red triangle down with } i \end{array} \equiv \delta_{ij} \quad (101)$$

Finally, we need to define a spider that encodes these $|\psi_{000}\rangle$ and $|\psi_{001}\rangle$ to the respective quantum states. In fact, since equation (19) is independent for the choice of basis, we can just copy it and exchange the white triangles for red ones.

Diagram (102) illustrates a spider rule. On the left, a red spider with m inputs and n outputs is shown. This is equated to a sum over i of a diagram consisting of two red triangles, each labeled with i , connected by a vertical line. The top triangle has n outputs and the bottom triangle has m inputs.

By a combination of equations (101) and (102) it is trivial to check that the red spider follows the spider rules as they are given in section 2.1.6. On the other hand, we have to tread carefully regarding its interaction with differently colored spiders, since we cannot exploit mutual unbiasedness.

We are now in principle ready to employ eight state encoding as it was proposed in [4] diagrammatically. Equation (76) tells us how we can make Bell maps diagrammatically, and equation (99) tells us that we can create all of our cipherstates by applying these Bell maps to the new basis states $|\psi_{00g}\rangle$ from equation (100). Defining a new color for each of the bases, we could thus encode all of our eight different cipherstates as follows.

Diagram (103) shows the encoding of the state $E_{uw} |\psi_{00g}\rangle$. On the left, a diagram shows a spider with two inputs, u and w , and a spider with output g . This is equated to a sum over four cases, each represented by a colored triangle labeled g :

- Red triangle: if $\{u, w\} = \{0, 0\}$
- Pink triangle: if $\{u, w\} = \{0, 1\}$
- Green triangle: if $\{u, w\} = \{1, 0\}$
- Blue triangle: if $\{u, w\} = \{1, 1\}$

Since the different bases will reappear quite often, we can save ourselves from unnecessarily complex diagrams by redefining each combination of u and w into a new spider, each with a different color, corresponding to the colors chosen for the different bases in equation (103). These eight state encoding spiders represent the four bases of eight state encoding in much the same way the gray and white spider represent the two bases of four state encoding.

$$\{u, w\} = \{0, 0\} \Leftrightarrow \bullet \quad \{u, w\} = \{0, 1\} \Leftrightarrow \circ \quad \{u, w\} = \{1, 0\} \Leftrightarrow \bullet \quad \{u, w\} = \{1, 1\} \Leftrightarrow \bullet \quad (104)$$

To make our lives easier, we introduce one more spider. This spider is actually nothing new, it is the set of the four spiders introduced in 104. Having this spider may come in handy whenever we want to make a statement that counts for all four of the colored spiders. Instead of making the same statement for each of the different spiders, we can then make it once for this new spider. We choose the color brown for this.

$$\bullet \equiv \{ \bullet, \bullet, \bullet, \bullet \} \quad (105)$$

Finally, we can do the same for triangles of all colors. A brown triangle thus represents the set of all colors of triangles. Using these new tools we can reduce equation (103) to:

$$(103) = \text{triangle with } g \text{ inside} \quad (106)$$

Let's first see what our QKR protocol looks like in this new notation.



Note that whenever there are multiple brown components in a diagram, this means that those components can be any color as long as they are the same.

Alice and Bob respectively encode and measure in the same basis. Eve controls some map Φ and receives one of its outputs. We do not include the one-time pad j or any of the other classical pre- and postprocessing here since we do not need it for this security proof. With this all set up we are in principle ready to develop a security proof for noiseless QKR, purely diagrammatically. Our security proof is very similar to the one proposed by Kissinger and Westerbaan in literature [6], where they prove security of four state QKD relying largely on the mutual unbiasedness of the gray and white bases. Although we do bring this mutual unbiasedness indirectly into our security proof as well by incorporating the Bell maps which consist of the gray and white spiders that we have seen before, we will extend this proof to also incorporate our four new non-mutually unbiased bases. As we will see, we will rely on gray and white spiders only through the use of these Bell maps..

Putting in the property that the quantum channels and maps exhibit no noise according to Alice and Bob is equivalent to saying that Eve goes undetected whenever Alice and Bob measure in the same basis. Since Eve represents the environment this can also be interpreted as the quantum channels being noiseless. We have already briefly discussed the security proof of Kissinger and Westerbaan in section 3.4, where we made the same statement for Alice and Bob using the gray and white bases in equation (97). In this section, we called this 'the noiseless case'. Noting that brown represents the set of four colors introduced in equation (105), we can implement this statement for our four new bases as follows.

$$\begin{array}{c} \text{brown dot} \\ \text{---} \\ \text{triangle } \Phi \\ \text{---} \\ \text{red dot} \end{array} = \bigwedge \begin{array}{c} \text{brown dot} \\ \text{---} \\ \text{triangle } \Phi \\ \text{---} \\ \text{green dot} \end{array} = \bigwedge \begin{array}{c} \text{brown dot} \\ \text{---} \\ \text{triangle } \Phi \\ \text{---} \\ \text{pink dot} \end{array} = \bigwedge \begin{array}{c} \text{brown dot} \\ \text{---} \\ \text{triangle } \Phi \\ \text{---} \\ \text{blue dot} \end{array} = \Leftrightarrow \begin{array}{c} \text{brown dot} \\ \text{---} \\ \text{triangle } \Phi \\ \text{---} \\ \text{brown dot} \end{array} \quad (108)$$

With this as a starting point we can pre- and postcompose decoding and encoding maps respectively as follows.

$$\begin{array}{c} \text{brown dot} \\ \text{---} \\ \text{triangle } \Phi \\ \text{---} \\ \text{brown dot} \end{array} = \text{brown dot} \quad (109)$$

Now we can purify the brown spiders in equation (109) by doubling and discarding an extra output according to equation (45). Note that we can also, without loss of generality, state that $\Phi = \hat{V}$ since Φ already fits the definition of a purified state given in equation (45). This gives rise to the following diagrams.

(110)

Where $\hat{\psi}$ can be any normalized pure state.

By applying the essential uniqueness of purification (the theorem introduced in section 2.2.3) to equation 110 we receive equation (111). Note that since we discarded multiple outputs in equation (110) we also have to apply the unitary U over multiple outputs.

(111)

Deleting the first and third outputs of the LHS of equation (111) gives the following.

(112)

Doing the same to the RHS of equation (111):

(113)

Note that this holds for spiders of all colors of equation (108). The following is thus also true for spiders of all these colors.

$$(114)$$

We can use this equation (114) to show that V separates. That is, the map itself splits up and leaves no connection between Eve's output and Alice and Bob's in- and outputs. Since V is just the undoubled version of Φ this then also implies that Φ separates in a similar fashion, which is what we wanted to achieve. Starting with V itself:

$$(115)$$

The strategy is to add something separated from the diagram in equation (115) and put V in there. We can just compose the creation and deletion of a random variable. Adding it to the diagram and writing one output of V a bit more suggestively:

$$(116)$$

Now, we need some method to attach the separated part to the diagram temporarily such that we can move V through. It turns out that eight state encoding gives us a map that can do exactly this: the random Bell maps. Equation (82) already showed how a random Bell map can separate two parts of a diagram. We place it between the two diagrams in equation (116).

$$(117)$$

What we need now is some way to pass V through these Bell maps and the surrounding encoding and decoding maps. Let's have a closer look at just the encoding map and the subsequent random Bell map by applying some test state g to it.

(118)

So what we have here is a random Bell map, precomposed by any of the four encoding maps and a test state of the same basis. Since each of the four bases are just Bell maps with respect to one another, combining the encoding and random Bell maps actually selects one of the four bases to encode in at random. More explicitly, remember by equation (99) that each of the cipherstates can be created by two Bell maps and the $|\psi_g\rangle$ basis states ($|\psi_{uwg}\rangle = B_1^w B_2^u |\psi_g\rangle$). Then applying a random Bell map ($B_1^n B_2^m$ where n and m are random) to any of the cipherstates gives the following.

$$B_1^n B_2^m B_1^u B_2^w |\psi_g\rangle = B_1^l B_2^p |\psi_g\rangle \stackrel{99}{=} |\psi_{lp g}\rangle \quad (\text{where } l \equiv n \oplus u \text{ and } p \equiv m \oplus w) \quad (119)$$

Which, by randomness of l and p , is a cipherstate encoded into a random one of the bases. Encoding a state into a cipherstate with a certain basis and then applying a random Bell map is thus equivalent to encoding this cipherstate into a random basis. We can find this diagrammatically too. Remember that equation (103) tells us how we can encode in any basis by applying two Bell maps with respect to a state in the red basis diagrammatically. The brown spider and state from equation (118) can thus be replaced by a red spider and state, followed by two Bell maps. Looking just at this encoding part and the random maps that follow immediately after we thus get:

(120)

Note that the letters next to the spiders have no physical significance, and are only there to help visualize which spider refers to which random variable.

The only ingredient that we are missing is some way to move Bell maps past one another. Mathematically, the fact that we can do this is evident from the anticommutative property of Pauli's. Anticommutativity implies that any two Pauli's σ_i and σ_j satisfy $\sigma_i \sigma_j = -\sigma_j \sigma_i$. Of course, this rule then also holds for the Bell maps. Diagrammatically we can commute Bell maps using the $k - k'$ commute rule from section 2.2.4 as follows.

Putting this back into equation 120 and moving around some spiders gives:

By equation (36) we know that the m and w here are xor'd by the gray spider and the n and u are xor'd by the white spider. We define $m \oplus w \equiv p$ and $n \oplus u \equiv l$, where randomness of m and n implies randomness of p and l . This gives:

Which is a cipherstate g encoded into a random basis by equation (103).

$$(123) \stackrel{(103)}{=} \frac{1}{4} \begin{array}{c} | \\ \text{g} \end{array} + \frac{1}{4} \begin{array}{c} | \\ \text{g} \end{array} + \frac{1}{4} \begin{array}{c} | \\ \text{g} \end{array} + \frac{1}{4} \begin{array}{c} | \\ \text{g} \end{array} \quad (124)$$

And thus we see that the encoding and subsequent Bell maps in equation (118) actually encode some input state g into a random one of the four bases. Also note that since g was a brown triangle, this result is independent of the basis of the input state. Encoding into a certain basis and then applying a random Bell map thus has the same effect as applying a random one of the basis spiders. We can put this result into the relevant parts of equation (117) as follows.

$$\begin{array}{c} \text{Diagram 1} \end{array} \stackrel{(124)}{=} \frac{1}{4} \begin{array}{c} \text{Diagram 2} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 3} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 4} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 5} \end{array} \quad (125)$$

By undoubling the wire and using equation (114) we can then move V through all of these.

$$\begin{array}{c} \text{Diagram 1} \end{array} \stackrel{(38)}{=} \frac{1}{4} \begin{array}{c} \text{Diagram 2} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 3} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 4} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 5} \end{array}$$

$$\stackrel{(114)}{=} \frac{1}{4} \begin{array}{c} \text{Diagram 6} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 7} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 8} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 9} \end{array} \quad (126)$$

$$\stackrel{(114)}{=} \frac{1}{4} \begin{array}{c} \text{Diagram 10} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 11} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 12} \end{array} + \frac{1}{4} \begin{array}{c} \text{Diagram 13} \end{array} \stackrel{(124)}{=} \begin{array}{c} \text{Diagram 14} \end{array}$$

Finally, applying this to the larger picture.

$$(117) = \begin{array}{c} \text{Diagram 1} \end{array} \stackrel{(126)}{=} \frac{1}{D} \begin{array}{c} \text{Diagram 2} \end{array} \stackrel{(82)}{=} \frac{1}{D} \begin{array}{c} \text{Diagram 3} \end{array} \stackrel{23}{=} \frac{1}{D} \begin{array}{c} \text{Diagram 4} \end{array} \quad (127)$$

And thus we see that V splits up such that Alice's input is connected to Bob's output directly, and that Eve's output is completely separated from this. Hence, under the condition that Alice and Bob measure and encode in the same four bases from equation (105) and see no disturbance or noise, Φ separates. In diagrams this as written as there existing some ρ such that Φ separates as follows.

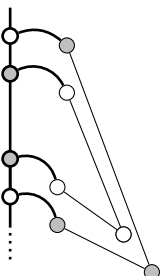
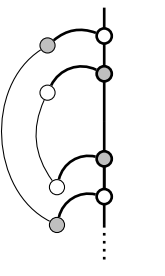
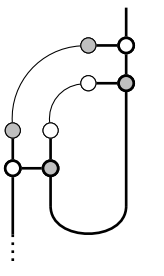
$$\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \diagup \\ \Phi \\ \diagdown \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \diagup \\ \rho \\ \diagdown \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \quad (128)$$

4 Equivalences

4.1 QOTP and Quantum Teleportation

The equivalence between QOTP and quantum teleportation is rather trivial. In fact, we already saw this equivalence back in equation (91).

$$\begin{array}{c} \text{Quantum One Time Pad} \end{array} \quad \begin{array}{c} \text{Quantum Teleportation} \end{array}$$

(78) =  $\stackrel{23}{=}$  =  = (90) \quad (129)

This is a strikingly simple equivalence. Bending the U-shaped curve in the quantum teleportation protocol such that it becomes a straight line already does the trick. In a sense, quantum teleportation could thus be seen as the EPR variant of the QOTP.

Due to their equivalence, whatever we can prove with the one diagram we can state as fact for the other without needing to prove it. This can be very convenient in security proofs.

4.2 Protocol equivalences from the paper "Quantum Alice and Silent Bob" [8]

In a recent article [8], Škorić and Leermakers propose a scheme for QKR that includes no classical communication from Alice to Bob and only one bit from Bob to Alice. To prove its security, the authors modify the proposed protocol to one that is better suited for this in a series of steps that preserve security-wise equivalence to the original. In this section, we will have a look at these equivalences diagrammatically using the eight state encoding QKR protocol from equation (130) as starting point which we have proven secure in the noiseless case in section 3.5.

$$\begin{array}{c} \text{Eve} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \diagup \\ \Phi \\ \diagdown \end{array} \begin{array}{c} \text{Bob} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \quad (130)$$

Alice

4.2.1 Masking the qubit payload with public randomness

In the first equivalent protocol Alice and Bob both xor the same bitstring $a \in \{0, 1\}^n$ in the classical domain with their payload and measurement result respectively. a is public, implying that Eve also learns it. The rest of the protocol remains the same. To implement this diagrammatically, we give Eve another map with respect to (96) to represent her doing classical post processing. In this map, she uses her knowledge on a and what she learned from intercepting the quantum state from Alice to Bob. We could also include this map later on. However, since we have to include it anyway we may as well do it here already, such that we can avoid to introduce it along with the more complicated equivalences that are to follow.

We need to create one more set of maps before we can implement this diagrammatically. Since we haven't fully worked out all of the properties of the colored spiders yet, it is more convenient to not use them whenever the protocol doesn't tell us to. Therefore, we propose a map that translates between the white and colored bases. This allows us to do all of the classical processing in terms of white and gray spiders which is necessary if we want to use, for example, mutual unbiasedness (2.2.2) or the xor map (2.1.10). The map that translates between the white and red bases is the following.

$$\begin{array}{|} \text{red spider} \end{array} : |g\rangle_z \mapsto |\psi_{00g}\rangle \quad \begin{array}{|} \text{red spider} \end{array} : |\psi_{00g}\rangle \mapsto |g\rangle_z \quad (131)$$

The other three colors then also have the respective boxes.

$$\begin{array}{|} \text{green spider} \end{array} : |g\rangle_z \mapsto |\psi_{10g}\rangle \quad \begin{array}{|} \text{green spider} \end{array} : |\psi_{10g}\rangle \mapsto |g\rangle_z \quad (132)$$

$$\begin{array}{|} \text{pink spider} \end{array} : |g\rangle_z \mapsto |\psi_{01g}\rangle \quad \begin{array}{|} \text{pink spider} \end{array} : |\psi_{01g}\rangle \mapsto |g\rangle_z \quad (133)$$

$$\begin{array}{|} \text{blue spider} \end{array} : |g\rangle_z \mapsto |\psi_{11g}\rangle \quad \begin{array}{|} \text{blue spider} \end{array} : |\psi_{11g}\rangle \mapsto |g\rangle_z \quad (134)$$

We call these maps the eight state encryption maps. Again, the brown box is the set of the four individual maps.

$$\begin{array}{|} \text{brown box} \end{array} \equiv \left\{ \begin{array}{|} \text{red spider} \end{array}, \begin{array}{|} \text{green spider} \end{array}, \begin{array}{|} \text{pink spider} \end{array}, \begin{array}{|} \text{blue spider} \end{array} \right\} \quad \begin{array}{|} \text{brown box} \end{array} \equiv \left\{ \begin{array}{|} \text{red spider} \end{array}, \begin{array}{|} \text{green spider} \end{array}, \begin{array}{|} \text{pink spider} \end{array}, \begin{array}{|} \text{blue spider} \end{array} \right\} \quad (135)$$

The maps on the right side of equation (135) can be summarized in Dirac notation as G_{uw} .

$$G_{uw} \equiv |0\rangle_z \langle \psi_{uw0}| + |1\rangle_z \langle \psi_{uw1}| \quad (136)$$

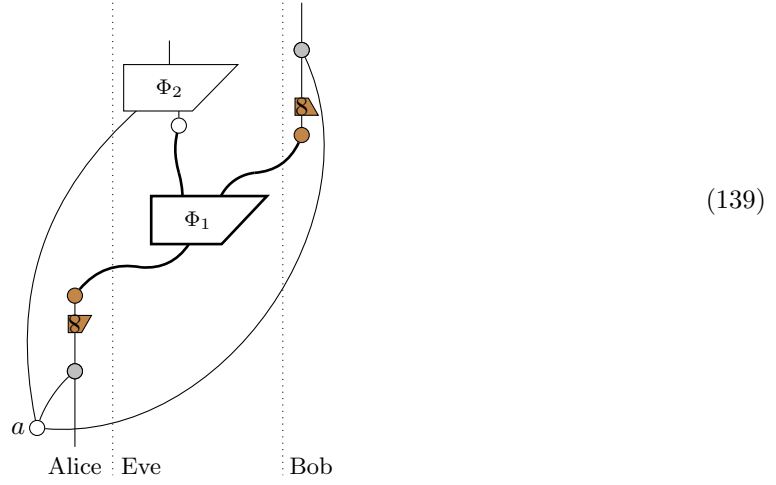
And the maps on the left side are then of course summarized as the Hermitian adjoint of G_{uw} , G_{uw}^\dagger .

$$G_{uw}^\dagger \equiv |\psi_{uw0}\rangle \langle 0|_z + |\psi_{uw1}\rangle \langle 1|_z \quad (137)$$

Since $G_{uw} \neq G_{uw}^\dagger$ the brown box is not self-adjoint and due to the complex elements in $|\psi_{uwg}\rangle$ it is not self-conjugate either. This is why we chose to give the eight state encryption maps a shape in which their orientations can easily be determined. We also know that they are unitary:

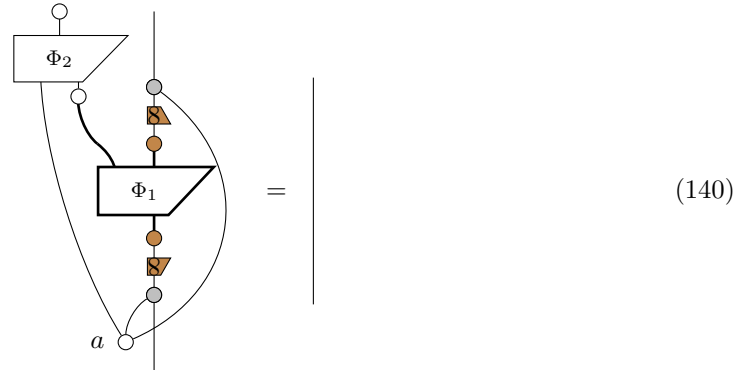
$$G_{uw}G_{uw}^\dagger = |0\rangle_z \langle 0|_z + |1\rangle_z \langle 1|_z = \mathbb{I} \Leftrightarrow \begin{array}{|} \text{brown box} \end{array} = \begin{array}{|} \text{brown box} \end{array} \quad (138)$$

Including these maps, and the fact that Alice and Bob both xor with a public variable a in their classical domain, we have the following diagram.



With Φ_1 the map that Eve uses to intercept the communication and Φ_2 the map that Eve uses to do classical processing on both this intercepted data and a .

For a security proof of this protocol, we can state that Eve's operations introduce no noise from Alice and Bob's perspective whenever they measure in the same basis. This is the same statement that we made in section 3.5 to prove the security of QKR, and in section 3.4 to prove the security of QKD, where we referred to this as the 'noiseless case'. For the diagram in equation (139), this looks as follows.



By causality we then know that applying Eve's map Φ_2 and then deleting its output is essentially the same as not having applied Eve's map altogether. In other words, from Alice and Bob's perspective, there is no difference between deleting the output of Eve's map Φ_2 or deleting its inputs even before she applies it. Therefore equation (140) reduces to:

$$(141)$$

Then we can pre and postcompose another xor with a on both sides of equation (141).

$$(142)$$

According to the diagrammatic rules for mutual unbiasedness given in equation (51) this separates a on both sides of equation (141).

$$(143)$$

Which is equivalent to the same diagrams without a random a .

(144)

Finally, postcomposing a brown eight state encryption map, precomposing its adjoint and exploiting the fact that it is unitary:

(145)

This equation we have already seen, it is the starting point for the security proof of eight state encoding QKR, equation (109).⁴ Therefore, we could just follow all the steps which we followed in section 3.5 and get as a result that Φ_1 separates as follows.

(146)

This is why this protocol is security-wise equivalent to the starting point of the equivalences, equation (130), it relies on the same security proof. If it turns out that the security proof in section 3.5 for the original QKR protocol only allows for security up to some factor, the protocol with public randomness from this section is also only secure up to that same factor. Putting this back into equation (139) we get the following:

⁴There is of course a minor difference: In (109) Eve has the map Φ and here Eve has the map Φ_1 . This is insignificant however. Φ_1 was only introduced to distinguish between Eve's quantum map and Eve's classical map. We can just say that $\Phi = \Phi_1$ without loss of generality since Eve's quantum map can be any (multitude of) unspecified operation(s) in both cases.

$$(147)$$

In a series of steps following rules which we have seen before we can show that Eve's map again separates entirely from Alice and Bob's part of the diagram.

$$(148)$$

So we see that even though Eve gets a and some quantum state, she is not attached to Alice and Bob's communication channel. Alice and Bob on the other hand achieve uninterrupted communication. This is what we would intuitively expect for the noiseless case.

4.2.2 EPR version of the protocol

The second equivalent protocol is an EPR version of the original. We have already seen a diagrammatic implementation of an EPR protocol in section 3.3, the diagram for quantum teleportation. The characteristic difference between an EPR version of a protocol and a normal version is that in the former Alice and Bob share some entangled qubit. In quantum teleportation this is the Bell state B_0 , given by the doubled wire shaped like a cup in equation (90).

Diagrammatically, it is rather trivial to go from the protocol introduced in the previous section (139) to an EPR version. In fact, we only need to bend around some wires. We split this up in a few steps to show where all the parts go. First of all, bending around Alice's wire according to the yanking equation (12):

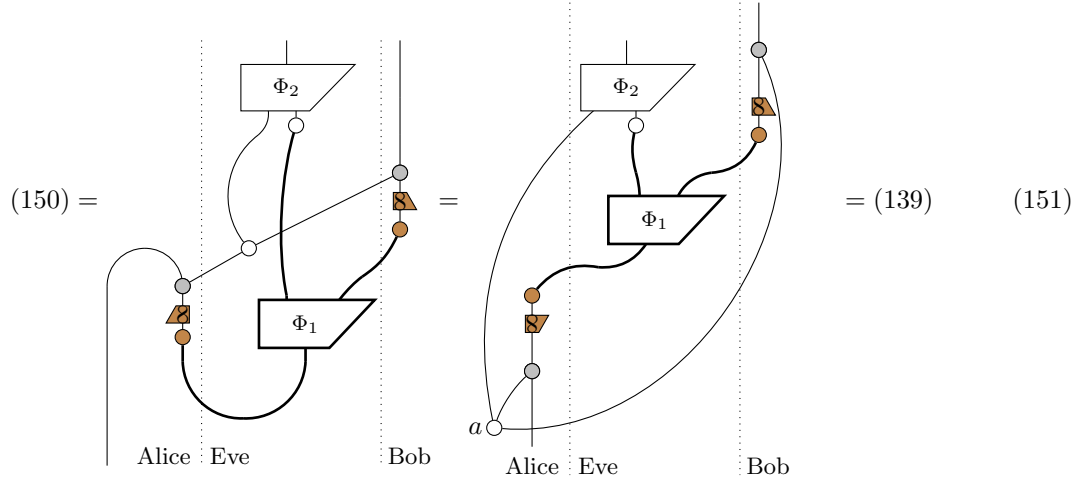
$$(149)$$

Subsequently pulling her encoding and encryption operations through:

$$(150)$$

Alice Eve Bob

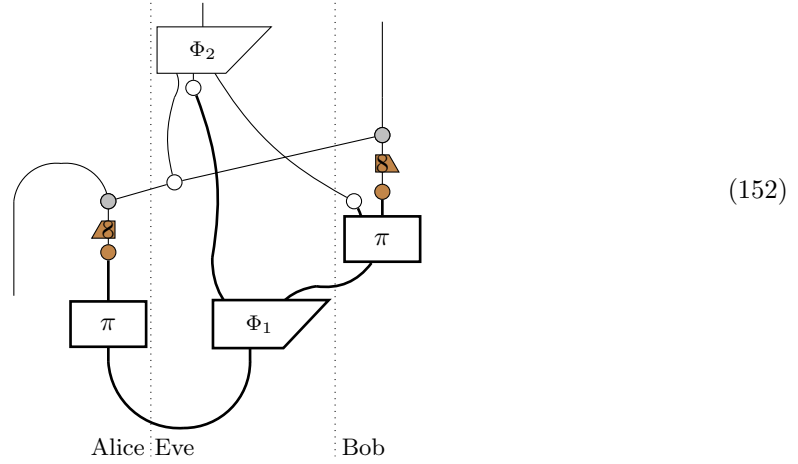
Which is indeed the EPR version of the protocol in equation (139).



Of course, if these protocols are equivalent, then by security-wise equivalence of (139) and (130) we also have security-wise equivalence of the EPR version introduced in this section, (150), and (130), the protocol that was the starting point of all these equivalences.

4.2.3 Adding a random permutation

In the next equivalent step, the authors propose that Alice and Bob both publicly decide on the same random permutation (π) to apply to their own set of qubit states before measuring, and forget it right after. Eve, however, can remember this permutation. Diagrammatically, this looks as follows.



Intuitively, it is evident that this step preserves equivalence to the original from Alice and Bob's perspective due to the fact that this step preserves entanglement of the EPR states. Whether this protocol works or not for Alice and Bob is independent on the choice of states that they receive, as long as they are entangled. The states that Eve sends to Alice and Bob are entangled up to a

certain degree dependent on the amount of noise Eve introduces, and the π 's completely preserve entanglement since the same operation is applied on both qubits. To make a diagrammatic proof from this intuition we have to first exploit the fact that we can turn Eve's map into a state. We can do this as follows.

$$(153)$$

Note that the dotted map in this diagram has no diagrammatic significance and is only intended for didactic purposes.

We can show that this is a valid operation diagrammatically. Isolating just the state of Eve in the RHS of equation (153) and realizing that Eve has the freedom to do whatever she wants in her state Φ_1 .

$$(154)$$

Then using the yanking equation (12):

$$(155)$$

We see that we can transform Eve's map into a state and reversedly. The operation from equation (153) is therefore valid.

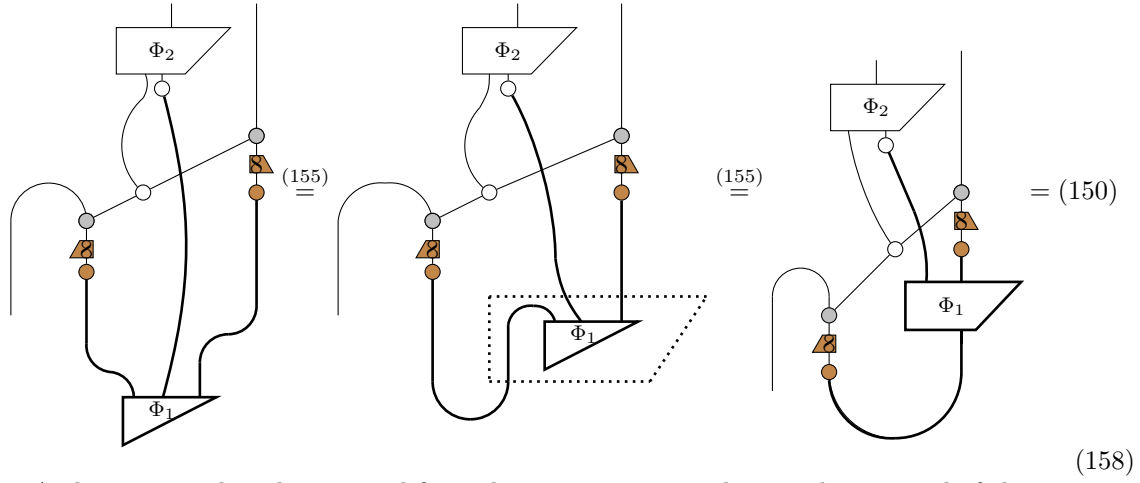
Starting with the RHS of equation (153) and using the yanking equation (12):

$$(153) = \text{Diagram 1} \stackrel{12}{=} \text{Diagram 2} \quad (156)$$

Now we are ready to put in the intuition that we had in the beginning of this section. From Alice and Bob's perspective, everything they need is for Eve to prepare and send them an (imperfect) EPR state. If Eve does this in her map Φ_1 the two subsequent random permutations will potentially change the particular choice of EPR states that Alice and Bob receive, but they will remain entangled since the same operation is applied on both sides. Furthermore, Alice and Bob forget π right after they apply it. Therefore, after it is applied, they are indifferent for as to whether they did it or Eve did it. From Eve's perspective, it does not matter whether she applies these π 's or whether Alice and Bob apply them. She knows the π 's that are applied in both cases. Taking this into account, we realize that we can just 'shove' the π 's into Eve's state and assume that she applied them rather than Alice and Bob. From a security perspective, this is an operation that preserves equivalence.

$$(156) = \text{Diagram 3} = \text{Diagram 4} \quad (157)$$

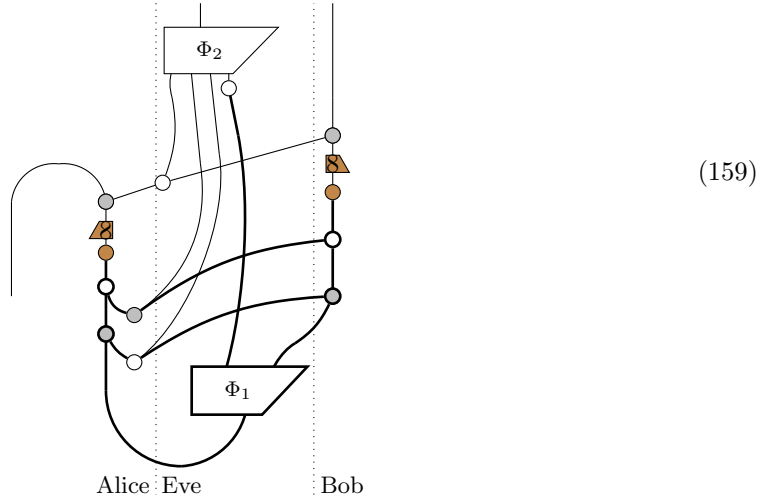
To show that this protocol is equivalent to the protocols from the previous sections we can just turn Eve's state back into a map.



And so we see that the protocol from this section is equivalent to the protocol of the previous section, equation (150), and therefore indirectly also to (130), the starting point of this series of equivalences.

4.2.4 Adding random Pauli transforms

In the final equivalent step, the authors propose that Alice and Bob both publicly decide on the same random Pauli to apply to their own qubit states before measuring. Afterwards they again forget about their particular choice of Pauli. Remembering the equivalence between the Pauli's and the Bell maps from equation (66) and how to construct Bell maps from equation (76), this looks as follows.



The difference between this protocol and the EPR protocol introduced in section 4.2.2 are the two single spiders in the middle which represent the classical random variables that determine the

Bell maps which are applied to both Alice and Bob's states. Information on the particular choice of Bell maps also goes to Eve by the thin lines originating from these two single spiders. Let's isolate the Bell map that Alice gets and her measurement and eight state encryption maps.

$$(160)$$

We do not consider the two triangles that go to Bob for now. These simply determine the Bell maps that he applies to his qubit state. All we need to know is that they are the same as Alice's. Furthermore we can take the brown decoding spider of Alice further apart according to equation (103). This equation shows how all the cipherstates can be encoded by a red spider and two Bell maps. Since decoding is the adjoint of encoding, we can decode in any of the four bases by first applying two Bell maps followed by a red spider. To then return back to the classical z basis we need to also exchange the brown eight state encryption box for a red one.

$$(160) \stackrel{103}{=} \sum_{nm} (161)$$

This equation is very similar to equation (120). In fact, for equation (120) we showed that encoding in a certain basis and then applying a random Bell map results in encoding in a random basis. Here we will show that applying a random Bell map and then decoding in a certain basis results in decoding in a random basis.

$$\begin{aligned}
 (161) &\stackrel{121}{=} \sum_{nm} \text{Diagram 1} \stackrel{23}{=} \sum_{nm} \text{Diagram 2} \stackrel{23}{=} \sum_{nm} \text{Diagram 3} \\
 &\hspace{15em} (162)
 \end{aligned}$$

By equation (36) we know that the gray n and u are xor'd by the white spider and that the white m and w are xor'd by the gray spider. Defining $m \oplus w \equiv p$ and $n \oplus u \equiv l$ and realizing that l and p are random then gives:

$$\begin{aligned}
 (162) &= \sum_{nmlp} \text{Diagram 4} \stackrel{21}{=} \sum_{lp} \text{Diagram 5} \\
 &\hspace{15em} (163)
 \end{aligned}$$

As expected, Eve receives the random variables n and m , but she does not learn l and p since these require knowledge of u and w . Alice and Bob thus measure in a random basis and Eve doesn't know this basis. More importantly for Alice and Bob, they measure in the same random basis since for Bob's side, the whole derivation follows the same series of steps. Putting this back into the larger picture, equation (159):

$$\begin{aligned}
(159) &\stackrel{163}{=} \sum_{lp} \text{Diagram 1} = \frac{1}{4} \text{Diagram 2} \\
&+ \frac{1}{4} \text{Diagram 3} + \frac{1}{4} \text{Diagram 4} + \frac{1}{4} \text{Diagram 5}
\end{aligned}
\tag{164}$$

Each of these protocols occur with probability $\frac{1}{4}$. We also know that they are all security-wise equivalent to the protocol in equation (150) since they form the members of the set of protocols in the RHS of equation 150. The protocol in equation (159) thus preserves security-wise equivalence to the protocol that was the starting point of this series of equivalences, equation (130).

5 Discussion

After reading up to this point, the novelty of this research may not be self-evident. After all, we did not develop any new protocols or give higher security bounds on existing protocols. Be that as it may, these are not the kind of things that should be expected of the diagrammatic method as of yet. The novelty of this work comes from the fact that it provides diagrammatic realizations of protocols and security proofs that have never been worked out in this fashion before. We took preexisting work and gave it a place in the context of the diagrammatic method for quantum cryptography. As of now it is not yet the task of the diagrammatic method to develop novel protocols or provide higher security bounds. Before we could do any of that, we should first see whether and how we can work up to the current state of the art diagrammatically. Only when and if we reach that point, we can attempt to use this method to actually bring the field of quantum cryptography further. This work provides a foundation to reach this point. Kissinger and Westerbaan built this foundation for

quantum key distribution in [6] and in this work we did the same for quantum key recycling with eight state encoding and the (quantum) one time pad.

With that being said, maybe it is not even the role of the diagrammatic method to eventually provide new protocols or bounds on security proofs. Coecke and Kissinger - in their first book on this method - propose it to be fully independent as an alternative to Dirac notation. Their book aims to give a first introduction into quantum mechanics, where new topics are introduced along with the notation and no knowledge of Dirac notation is assumed. Since this works well, the notation allows for a satisfactory representation of the introductory concepts of quantum mechanics. The question then arises of whether it also allows for the representation of more advanced quantum mechanical topics, in particular in the field of quantum cryptography. In this report, we take it a bit further than in the book of Coecke and Kissinger. We use the diagrammatic method to make security proofs and analyze some recent papers' equivalences. We do at no point, however, assume that Eve introduces noise since whenever we would do this, we would introduce more advanced equations for which we have not worked out the diagrammatic notation yet. Kissinger and Westerbaan in [6] do not shy away from combining the notation with such equations in their security proof for QKD with noise. With that being said, their approach is a bit clumsy since they remove the need for diagrams by using these equations. Their security proof by itself is also not state of the art. It is mostly relevant since it is the first security proof for QKD that is conducted mostly with the diagrammatic notation. Therefore the question remains whether this notation should really be used for tasks such as making rigorous security proofs or the development of new and improved protocols. So far it does not seem very suitable for this task. With that being said, it is a relatively new method - Coecke and Kissinger released their book in 2017 [3] - so it may in the future take up more responsibility. What we can say about it as of now is that it is a good notation for intuitively displaying protocols and a very useful method for visualizing their equivalences. The equivalence between the QOTP and quantum teleportation, for example, is not that straightforward just from reading the text, but is strikingly simple diagrammatically.

The critique that we can give to this method also depends on the role that it takes up. If we assume that the role of the diagrammatic method is to give in-depth security proofs for noisy protocols with advanced statistics, we could easily say that the notation is not rigorous enough. Under this assumption, it would have to be developed further to be relevant. If we assume, on the other hand, that the role of the diagrammatic method is to draft up and intuitively display proofs and equivalences, we would have to be much less critical. The latter is therefore also the role that we assume it will have in future work. A cause for confusion of this notation though, independent of its role, is the fact that single wires have a basis. Interpreting single wires as classical data is not very intuitive. Bits are not encoded into bases. A better way to interpret single wires is then as carriers of 'classical data on how to encode a qubit in a certain basis'. However, in this case every wire would carry around some information on what color it is in. This is not how wires are represented in this notation, they do not have a color. Coecke and Kissinger identify this problem as well in their book [3] on page 740, where they propose to place a small square next to each wire to show the basis of the information that it carries.

$$\begin{array}{c} | \blacksquare \end{array} \quad \begin{array}{c} | \square \end{array} \quad (165)$$

On the one hand, including this in the notation would make it more rigorous. There would be no confusion anymore on the basis of the information carried by each wire. On the other hand, it would remove a lot of the elegance and simplicity of the notation. It would probably be good to

use this notation whenever the basis of the information carried by a wire is not clear from context, and to omit it otherwise.

This work leaves open plenty of opportunities for future research. As a direct follow up to this report, a noisy version of the quantum key recycling security proof could be developed. It is probable that a security proof along the lines of the one for noisy quantum key distribution given in [6] also applies to the QKR protocol from section 3.5. Furthermore, since the diagrammatic method is very young and literature on it is scarce, it would benefit from more protocols and their security proofs being written out in this fashion.

6 Conclusion

The aim of this research was largely divided up into two parts. The first was to introduce the reader to the diagrammatic method in a concise manner. Therefore, chapter 2 functions as a sort of handbook to the diagrammatic notation for a reader familiar with undergraduate level quantum mechanics. Although all the necessary concepts were introduced and derived back to Dirac notation successfully, whether this fully satisfied this aim or not is up to the reader to decide. Secondly, this research aimed to place this notation into the relevant quantum cryptographic context. To this end, we worked out four protocols and their security diagrammatically, and also used the notation to represent a series of equivalences. In particular for quantum key recycling we developed new notation for eight state encoding, a concept that was recently introduced by Škorić and De Vries [4]. Furthermore we gave a security proof for eight state QKR in a similar fashion to how it was done for QKD in [6]. The equivalences that we worked out diagrammatically come from another publication [8]. In this research, the authors provide a series of security-wise equivalent steps, allowing them to prove security of one protocol indirectly by proving the security of another. We went through each of these steps diagrammatically, which allowed for both their formal proof and an intuitive understanding of these equivalences. With all of this taken into account, it can be said that this second aim - to build upon the novel notation and use it to add relevant research to the field of quantum cryptography - was successfully achieved. This work does leave open significant questions about the role of the diagrammatic method in the field of quantum cryptography though. We conclude that the diagrammatic notation should not be used for its rigorosity, but rather for its capability to intuitively describe protocols and their equivalences. In future work, a noisy version of QKR could be developed. In the larger context, the notation would benefit from being implemented in more protocols and their respective security proofs.

7 Critical reflection

In general, I would say that I am very satisfied with the project and its results. Of course, there are some points of critique and general thoughts to keep in mind for future projects that I can come up with.

7.1 Organization of tasks

At the start of the project I had many ideas about how to organize the project and optimize the time I had for communication with my supervisors. One of them, for example, was to use a Kanban board that my supervisors could access as well to organize tasks. This Kanban board was useful in

the beginning of the project when I was defining the scope and assessing what needed to be done. However, after a couple of weeks I found myself looking at the board only rarely. Although I did not get any feedback on this, I do not think my supervisors measured my progress by the Kanban board either. I forced myself to keep using it for a while but decided to stop doing this completely after a couple of weeks. I noticed that after I stopped using it I naturally went into a cycle of making progress for a few weeks and writing for a week or two after I reached some milestone. At the end of each cycle, writing about it would finish that part of the work such that I could put it to the side and clear my mind to think about what task to take on next. This method of organizing tasks is mainly different from the use of a Kanban board in that there is no predefined set of tasks to follow; each task is made up only after the previous one is finished. I enjoyed working like this more and I think it also increased productivity. This is a pattern that I actually noticed more often. As I started to feel more comfortable in the project I started to organize tasks more naturally rather than by following some predefined framework. In general, this actually made the work more enjoyable and increased productivity. This is something that I will remember and try to take into account from the start in future projects.

7.2 Listening while taking up information

Listening while taking up information was probably the most difficult aspect of the practical work of the thesis. I had many interesting and educative conversations with my supervisors and others who were doing research in the same field - something I actually did not expect to happen due to the limited amount of available time that academics tend to have. In the beginning, I used to write down as much as I could during these conversations such that afterwards I would be able to recall the points and actually implement them in the thesis. However, I noticed that these conversations actually became more fruitful after I stopped writing down so much because that gave me more time to both listen and think about a response or follow-up question. The downside of this was that I would sometimes forget points that were made during the conversation. In the end, I chose to write down very little, but would try to write down a keyword for each main discussed point. Although this worked quite well, I would still sometimes forget some details and I think the conversations still suffered from it a little. There are of course other options, such as recording the conversation. However, I cannot expect everyone to be comfortable with this, and this would probably also not be completely appropriate in more informal settings such as during lunch. I still do not know the optimal way to tackle this problem but I think that writing down just the main points while listening very closely is a good compromise. In the future I may also try to take some time right after each conversation, when memory of it is still fresh, to write down a summary of what was discussed.

7.3 Putting things in perspective

Sometimes I have trouble putting things in perspective. This is something that I was actually not aware of before or during the project, but was told to me afterwards by my research supervisor. After receiving this feedback I reread some parts with this in mind and I think I understand where this comes from. Firstly, I could have explicitly taken some more time to develop an understanding of the context but I chose to focus more on my own research and the papers that were directly relevant to it. Secondly, I think that experience in the field helps to place a work like this in the larger context. This experience gives a better view of what has come before, what research is being

done currently, and what the likely paths are for the future. I had people around me who were much more experienced in the field, and who did possess knowledge of the larger context. For example, my research supervisor and his PhD student have multiple articles on QKR, one of the main subjects of this thesis. With them, I often discussed how this work could be put in the relevant perspective. I think what went wrong here relates to the previous section, 7.2. Although we did talk much about how specific parts of this work could be put in perspective, it is not what I thought I should take away from these conversations. What I did try to remember from these conversations were usually more technical and tangible points since these were the kinds of things that I could put directly into words and diagrams after such a conversation. In the future I would approach this differently. I think the solution that I proposed before in 7.2 - to write a small summary after each conversation - would work well, provided I write about what was discussed about the larger context as well.

References

- [1] Abn amro investeert in quantumtechnologie, 2019. <https://www.abnamro.com/nl/newsroom/nieuws/2019/abn-amro-investeert-in-quantumtechnologie.html>.
- [2] Andor Buding. Race Against Time: Securing our Future Data with Quantum Encryption, 2015.
- [3] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes*. Cambridge University Press, Cambridge, 2017.
- [4] Manon de Vries and Boris Skoric. Quantum Key Recycling with Eight-State Encoding. *Cryptology ePrint Archive*, 2016.
- [5] Ivan Djordjevic. Chapter 7 - quantum error correction. In Ivan Djordjevic, editor, *Quantum Information Processing and Quantum Error Correction*, pages 227 – 276. Academic Press, Oxford, 2012.
- [6] Aleks Kissinger, Sean Tull, and Bas Westerbaan. Picture-perfect Quantum Key Distribution. Cryptology ePrint Archive, Report 2017/1704, 2017. <http://arxiv.org/abs/1704.08668>.
- [7] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Physical Review A - Atomic, Molecular, and Optical Physics*, 1997.
- [8] Daan Leermakers and Boris Skoric. Quantum alice and silent bob: Qubit-based quantum key recycling with almost no classical communication. Cryptology ePrint Archive, Report 2019/875, 2019. <https://eprint.iacr.org/2019/875>.
- [9] Göran Lindblad. A general no-cloning theorem. *Letters in Mathematical Physics*, 47(2):189–196, Jan 1999.
- [10] John Markoff. Codebook Shows an Encryption Form Dates Back to Telegraphs. *The New York Times*, 2011.
- [11] Oliver Morsch. *Quantum Bits and Quantum Secrets: How Quantum Physics is revolutionizing Codes and Computers*. John Wiley & Sons, Berlin, 2008.

- [12] Michael a. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 2011.
- [13] Mehdi Saeedi and Igor Markov. Synthesis and optimization of reversible circuits - a survey. *ACM Computing Surveys*, 45, 10 2011.
- [14] C. E. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 1949.
- [15] Peter W. Shor and John Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, 2000.
- [16] Stephen J. Weisner. Conjugate coding. *ACM SIGACT News*, 15(3):78–88, 1983.

A Keywords

Base, page 7
 Bloch sphere, page 10
 Bra, page 2
 Color, page 7
 Complementarity, page 11
 Decoding, page 11
 Deleting, page 6
 Density matrix, page 8
 Discarding, page 9
 Doubling, page 7
 Effect, page 2
 Encoding, page 11
 Entangled state, page 9
 Essential uniqueness of purification, page 12
 Fully mixed state, page 9
 Hermitian Adjoint, page 3
 Hermitian Conjugate, page 3
 Identity, page 3
 $k - k'$ commute rule, page 13
 Ket, page 2
 Kronecker delta, page 5
 Linear map, page 4
 Map, page 5
 Mixed state, page 8
 Phase, page 6
 Phase spider, page 6
 Purification, page 9
 Purity, page 8
 Random variable, page 6
 Spider, page 5
 State, page 2

System type, page 4
 Tensor product, page 3
 Trace, page 9
 Transpose, page 3
 Wire, page 3
 Xor map, page 12
 Yanking equations, page 10

B Bell and Pauli matrices

The Bell maps are represented by the following matrices:

$$\begin{aligned}
 B_0 = \sigma_0 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & B_1 = \sigma_1 &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
 B_2 = \sigma_3 &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} & B_3 = i\sigma_2 &= \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}
 \end{aligned} \tag{166}$$