Decoy-State Quantum Key Distribution with Arbitrary Phase Mixtures and Phase Correlations

by

Shlok Nahar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Physics (Quantum Information)

Waterloo, Ontario, Canada, 2022

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

I would like to acknowledge Twesh Upadhyaya's contributions to Section 3.3, Appendix A.1 and A.2, and walking me through the methods used in [1] which were extended to form Section 3.3.

I would also like to acknowledge Nicky Kai Hong Li's contributions to Appendix B.2 where he gave me an analytical proof of the monotonicity of the cross-click function. Before that, I was using numerics to prove this, which although accurate seems far less satisfying.

Abstract

We formulate a general method to find bounds on the statistics of states passing through an unknown channel from the statistics of another set of states. We pay special attention to the application of this method to decoy-state quantum key distribution (QKD) where the states that can be practically prepared are not always the most secure states. In contrast to standard decoy-state analysis, we do not assume that our states are phase-randomised and can consider a fairly general laser source. We also develop a method to accommodate phase correlations with minimal characterisation of the source. Thus, we develop general techniques to deal with phase imperfections in a large class of QKD protocols. We apply these methods to a simple implementation of the three-state protocol and discuss the effects of partial phase-randomisation on the key rate of the protocol.

Acknowledgements

First and foremost, I would like to express my deepest appreciation for my supervisor Prof. Norbert Lütkenhaus for giving me the opportunity to pursue my Master of Science at the Institute for Quantum Computing. He was always extremely generous with his time and patient through my failures. The combination of independence and guidance he gave me was crucial to bring me a step closer to becoming an independent researcher.

I would like to thank the entire Optical Quantum Communication Theory group for many valuable discussions. Special thanks to Nicky Kai Hong Li for showing me the ropes when I was new to QKD. I would also like to thank Twesh for many relevant discussions that helped push my research along.

I would like to express my sincere thanks to Prof. Vern Paulsen for his generous help in discussing the parts of my research related to functional analysis.

I would also like to thank my MSc advisory committee, Prof. Thomas Jennewein and Dr. Agata Branczyk for their valuable time and helpful advice.

Many thanks to those who carefully read through my thesis and gave me valuable feedback to improve it: Shreya Arya, Lars Kamin, Jie Lin, Twesh Upadhyaya, Pritam Priyadarsi, John Burniston, Nicky Kai Hong Li, and Prof. Norbert Lütkenhaus.

Finally, I thank my family and friends for their support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The power of QKD lies in its information theoretic security in contrast to classical cryptography that rely on computational assumptions. However, implementations of QKD also must make practical assumptions leading to a gap between theory and practice. One such assumption is made when theoretically modelling the devices used. Often inaccurate models of the devices are used either due to difficulties in the accurate characterisation of the devices, or because the existing theory is unable to accommodate more accurate models.

In an effort to mitigate the effect of the security loopholes caused by inaccurate models, one approach is to develop methods to perform QKD without trusting the devices used. To this end, there has been much work on measurement-device-independent QKD (MDI-QKD) [3, 4, 5, 6, 7] which is able to share secret keys without trusting the measurement devices.

There is comparatively less work done on the source side. The recent work on the source side takes into account general imperfections [8], correlations [9], and side-channels [10]. Parallelly, there has been work to increase the key rate of optical protocols that do not use a qubit space through decoy-state analysis [11]. This increase in key rate arises as a result of the use of additional (decoy) states that constrain the action of the insecure quantum channel on the signal states. However, there is no known way to combine the methods involving general source imperfections with standard decoy-state analysis which severely limits their tolerance to loss. Thus, there is some need to provide analysis that is compatible with decoy-state QKD that can be applied to partially characterised sources. Regarding this, there has been recent work that takes into account intensity correlations and fluctuations and applies it to decoy-state QKD [12] which assumes that all the states are fully phase-randomised.

Our work similarly focuses on expanding the source models that can be used for decoy-state analysis to achieve practical loss tolerances for QKD protocols. In specific, we deal with the issues that arise as a result of imperfectly phase-randomised states with correlations. This thesis is structured as follows. In Chapter 2 we give a brief review of the basic concepts for quantum information, QKD, and quantum optics required for this thesis.

In Chapter 3, we formulate a general method to bound the statistics of a state passing through an unknown channel given the statistics of another set of states that pass through the same unknown channel. We show how this can be applied to perform decoy-state QKD with no assumption on the kind of states we can use, aside from the fact that they must be independent and identically distributed (iid).

In Chapter 4, we discuss a wide class of phase imperfections including phase correlations. We give a physical example of phase correlations in lasers and develop a general method to relate an arbitrary, possibly correlated phase mixture of coherent states to an iid model laser state with minimal characterisation of the state. We show how in most QKD protocols, this model laser state would lower bound the key rate of a protocol using the more realistic general laser state.

In Chapter 5, we apply the methods we developed to the three state protocol where high clock rates cause phase correlations. We discuss the issues in characterising the laser source and show the effect of imperfect phase-randomisation for the protocol through a key rate plot. We also discuss some looseness in our bounds that could lead to the gap in key rates between the partially and fully phase-randomised cases seen in the plot. Further research is needed to tighten these bounds.

In Chapter 6, we conclude the thesis by summarising our work and suggesting future avenues of research.

Chapter 2

Preliminaries

Quantum mechanics is a theory developed to model the behaviour of systems where classical or Newtonian mechanics gives incorrect predictions. In this chapter, we will first review the basic quantum information theory required to understand this thesis in Section 2.1. A lot of what we do here does not require the nuances introduced by considering infinite dimensions, and so we shall give some intuition for the inifinite dimensional cases, while only fully explaining finite dimensional cases. For a more rigorous introduction to infinite dimensional quantum information, the reader can refer to [13]. We then briefly define semidefinite programs (SDPs) in Section 2.2 We then review the basics of QKD that is essential for the thesis in Section 2.3. Finally, we review the quantum optics needed to understand some simple optical implementations of QKD protocols in Section 2.4.

2.1 Quantum information theory

Our review in this section is based largely on [14] and [13]. We shall view quantum mechanics as a set of physical procedures that can be performed in the lab. We can appeal to classical probability distributions as a motivational example for this viewpoint. Consider a deck of cards. The probability that the top card is the ace of spades is 1/52 for a well-shuffled deck of cards. However, you could initially prepare and shuffle the deck in a way to bias towards drawing the ace of spades from the top of the deck. The probability distribution assigned to the card draw has nothing to do with the cards themselves, or the actual outcome. They are instead a description of the operations that you performed on the deck of initially preparing it and shuffling it in a deliberate manner.

Similarly, any physical quantum process can be thought of as successive applications of three operations:

1. Preparation: A process that outputs a fixed quantum system.
2. Transformation: A process that takes as input a quantum system and outputs a (possibly different) quantum system.
3. Measurement: A process that takes as input a quantum system and outputs a classical data.

Each of these operations can take as input a classical system that can partially or fully specify the operation being performed. The classical system can be deterministic or probabilistic. The mathematical framework describing these operations is of operators on Hilbert spaces.

2.1.1 Hilbert spaces

Before we can talk about these processes we need to explain some of the underlying mathematical framework. We shall first define the intuitive notion of an inner product.

Definition 1 (Inner Product Space)**.** An **inner product space** is a vector space V over the field $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$ together with a map

$$\langle \cdot, \cdot \rangle : V \times V \to \mathbb{F}$$

called an **inner product** that satisfies the following conditions for all vectors $x, y, z \in V$ and all scalars $c \in \mathbb{F}$:

1. Linearity in the second argument:
$$\langle x, cy \rangle = c\langle x, y \rangle,$$
$$\langle x, y + z \rangle = \langle x, y \rangle + \langle x, z \rangle$$
2. Conjugate symmetry:
$$\langle x, y \rangle = \langle y, x \rangle^*$$

3. Positive Definiteness:
$$\|x\|^2 := \langle x, x\rangle \geq 0$$
where equality holds iff $x = 0$. The quantity $\|x\|$ is called the induced **norm** of x or just norm for short.

In quantum mechanics, the field we deal with is the complex numbers $\mathbb{C}$.

As stated above, the underlying mathematical structure starts with Hilbert spaces.

Definition 2 (Hilbert space)**.** A complex **Hilbert space** $\mathcal{H}$ is an inner product space over the complex numbers $\mathbb{C}$ which is also complete in the norm.

Intuitively, completeness is the condition that any sequence that converges in norm should have a limit in the space. This is only a non-trivial condition in infinite dimensions. All finite-dimensional complex Hilbert spaces are isomorphic to complex Euclidean spaces $\mathbb{C}^d$, and even in infinite dimensions much of the intuition carries forward.

Definition 3 (Dual Hilbert space)**.** The **dual space** $\mathcal{H}^*$ is defined as the space of all continuous linear functions from $\mathcal{H}$ to $\mathbb{C}$.

Theorem 1 (Riesz representation theorem)**.** Let $\mathcal{H}$ be a Hilbert space. For every continuous functional $\varphi \in \mathcal{H}^*$ there exists a unique $f_\varphi \in \mathcal{H}$ such that
$$\varphi(x) = \langle f_\varphi, x\rangle \quad \text{for all } x \in \mathcal{H}.$$

This gives us a way to identify every dual vector with a vector from the Hilbert space. In physics, we typically denote vectors $x \in \mathcal{H}$ as $|x\rangle \in \mathcal{H}$ which we call **kets**, dual vectors $\varphi \in \mathcal{H}^*$ via the corresponding vector $\langle f_\varphi| \in \mathcal{H}^*$ which we call **bras**, and the inner product between two vectors $\langle x, y\rangle$ as $\langle x|y\rangle$. Intuitively, we can think of a Hilbert space vector $|v\rangle$ as a column vector, and its dual vectors as the conjugate transpose which will denote with $\dagger$ as $\langle v| = |v\rangle^\dagger$. So, the inner product of two vectors is equivalent to taking the dual of one, and performing matrix multiplication. This isn't quite so straightforward in infinite dimensions, but thanks to Theorem 1, most of this intuition carries forward.

2.1.2 Operators on Hilbert spaces

Most operations in quantum mechanics can be described by bounded linear functions, or bounded operators acting on Hilbert spaces. We denote $\mathcal{L}(\mathcal{H}_1, \mathcal{H}_2)$ to be the set of all linear operators from $\mathcal{H}_1$ to $\mathcal{H}_2$ and $\mathcal{L}(\mathcal{H})$ as shorthand for $\mathcal{L}(\mathcal{H}, \mathcal{H})$.

Definition 4 (Bounded operators)**.** Let $\mathcal{H}$ and $\mathcal{K}$ be Hilbert spaces and let $T \in \mathcal{L}(\mathcal{H}, \mathcal{K})$ be a linear operator. T is **bounded** if there exists a constant M such that $\|T(h)\| \leq M \|x\|$ for all $h \in \mathcal{H}$. We define

$$\|T\|_\infty \coloneqq \sup\{\|T(h)\| \mid \|h\| \leq 1\} \leq \infty \tag{2.1}$$

as the **operator norm** or the **infinity norm**.

We denote $\mathcal{B}(\mathcal{H}, \mathcal{K})$ to be the set of all bounded operators from $\mathcal{H}$ to $\mathcal{K}$, and $\mathcal{B}(\mathcal{H})$ as shorthand for $\mathcal{B}(\mathcal{H}, \mathcal{H})$.

Boundedness captures the intuition that the action of an operator on any unit vector should produce a vector with finite norm.

Definition 5 (Orthonormal Basis)**.** Given an inner product space a set of vectors S is called **orthonormal** if $|v\rangle \in S$ implies $\|v\| = 1$ and whenever $|v\rangle, |w\rangle \in S$ and $|v\rangle \neq |w\rangle$ then $\langle v|w\rangle = 0$.

We say that an orthonormal set $\{|e_a\rangle\}$ in a Hilbert space $\mathcal{H}$ is an **orthonormal basis** if there does not exist an orthonormal set that contains it as a proper subset.

All Hilbert spaces that we encounter in quantum mechanics have a countable orthonormal basis i.e. they are **separable**. Any Hilbert space vector $|v\rangle$ can be written as a linear combination of basis vectors $|v\rangle = \sum_a \langle e_a|v\rangle |e_a\rangle$. Although this is trivial in finite dimensions, it requires some proving in infinite dimensions (see Theorem 2.94 from [13]). We shall take this fact for granted throughout this thesis.

Since we only deal with linear operators, an orthonormal basis is useful as we only need to describe the action of the operator on the basis. This uniquely defines the action of the operator on the entire Hilbert space. So, we can write an operator $T \in \mathcal{B}(\mathcal{H}, \mathcal{K})$ as

$$T = \sum_{e_a \in \mathcal{H}, f_b \in \mathcal{K}} \langle f_b, T(e_a)\rangle f_b e_a^*,$$

where $\{e_a\}$ and $\{f_b\}$ are bases for $\mathcal{H}$ and $\mathcal{K}$ respectively. In finite dimensions, this means that T can be written as a matrix where the matrix elements would be $\langle f_b, T(e_a)\rangle$. Much of the relevant intuition for bounded operators in infinite dimensions can similarly be obtained by simply imagining it to be a matrix with (countably) infinitely many rows and columns. The action of the operator in bra ket notation can be written as $T(h) = T|h\rangle$ which corresponds exactly to matrix multiplication in finite dimensions. We shall now introduce a few important classes of operators that are useful for quantum information.

Definition 6 (Self-adjoint operators)**.** Let $\mathcal{H}$ and $\mathcal{K}$ be Hilbert spaces. Given a bounded operator $T : \mathcal{H} \to \mathcal{K}$, the Hilbert space **adjoint** is the unique bounded linear map $T^\dagger : \mathcal{K} \to \mathcal{H}$ such that

$$\langle k|T(h)\rangle = \langle T^\dagger(k)|h\rangle \tag{2.2}$$

for all $h \in \mathcal{H}$ and $k \in \mathcal{K}$. An operator T is **self-adjoint** or **Hermitian**[1] if it is the same as its adjoint $T = T^\dagger$. The set of all Hermitian operators acting on $\mathcal{H}$ is denoted by $\mathrm{Herm}(\mathcal{H})$.

In finite dimensions, the adjoint is exactly the conjugate transpose of the operator, and hence our use of † is consistent.

Definition 7 (Positive semidefinite operators)**.** An operator $P \in \mathcal{B}(\mathcal{H})$ is called **positive (semi)definite**, denoted $P > 0$ (≥ 0) if $\langle h|P|h\rangle > 0$ (≥ 0) for every $|h\rangle \in \mathcal{H}$.

Equivalently, P is a positive semidefinite operator if there exists $B \in \mathcal{B}(\mathcal{H})$ such that $P = B^\dagger B$. From this equivalent definition, it is easy to see that every positive semidefinite operator is also self-adjoint. We will sometimes use the notation $P \geq Q$ to mean that the operator $P - Q$ is positive semidefinite. We denote the set of all positive semidefinite operators acting on Hilbert space $\mathcal{H}$ as $\mathrm{Pos}(\mathcal{H})$.

A special set of positive semidefinite operators are sets of **orthogonal projection operators** or **projectors** for short. A set of projectors $\{\Pi_k\}$ satisfy $\Pi_k \Pi_l = \delta_{lk}\Pi_l$ and $\Pi_l = \Pi_l^\dagger$. Lastly, we need to define trace and density operators before talking about quantum processes.

Definition 8 (Trace)**.** Let $T \in \mathcal{B}(\mathcal{H})$ and let $\{|e_a\rangle\}$ be any orthonormal basis for $\mathcal{H}$. The **trace** of T denoted by $\mathrm{Tr}\,[T]$ is $\sum_a \langle e_a|T|e_a\rangle$ and its value is independent of the basis chosen.

An additional caveat for infinite dimensional operators is that the trace is only defined when the sum converges. The class of such operators where the sum converges is called **trace class operators** denoted by $\mathrm{TC}(\mathcal{H})$. For finite dimensions, the trace is always defined and is the sum of the diagonal elements of the matrix representation of the operator in any basis. The trace is linear and cyclic i.e. $\mathrm{Tr}\,[ABC] = \mathrm{Tr}\,[CAB]$.

Definition 9 (Density Operator)**.** An operator $\rho \in \mathcal{B}(\mathcal{H})$ is called a **density operator** or density matrix[2] if it is positive semidefinite and has trace 1.

[1] Hermitian and self-adjoint are the same for bounded operators, but not in general. The interested reader might wish to refer to Section 3.12 of [13].

[2] Although the operator is not strictly speaking a matrix in infinite dimensions, so much of the intuition from finite dimensions carries forward that we use the terms interchangably.

We denote the set of all density matrices on a Hilbert space $\mathcal{H}$ as $\mathrm{D}(\mathcal{H})$.

2.1.3 Preparations

We can now go back to our processes from the start of the Section 2.1. This section will talk about the first process, that of preparing quantum systems.

Postulate 1 (Quantum State)**.**

Ideal

An ideal preparation procedure is described by a Hilbert space vector and is called a **pure quantum state**.

General

A general preparation procedure or **quantum state** is described by a density operator acting on a Hilbert space.

As stated above, we view the quantum state as a description of the preparation procedure and not a physical object itself. If our preparation procedure produces a set of pure states $\{|\psi_i\rangle\}$ according to some probability distribution $p(i)$, we can describe this preparation procedure by the density operator $\rho = \sum_i p(i)|\psi_i\rangle\langle\psi_i|$. Thus, the trace condition is analogous to the probabilities of all events summing to 1. Note that sometimes subnormalised states might be used i.e. states with trace less than 1 for e.g. when discarding some events. We shall explicitly state when we use subnormalised states.

In the density matrix formalism we often refer to the operator $|\psi\rangle\langle\psi|$ and the vector $|\psi\rangle$ interchangeably as a pure state. This is entirely consistent as there is a bijective correspondence between the set of rank 1 projectors and the vectors on a Hilbert space. If a density operator is not a pure state, then we call it a mixed state. Note that we can represent classical probability distributions as diagonal density matrices. Given a basis, these are called **classical states**.

2.1.4 Measurements

This section will deal with the third process, that of measuring a quantum system to output classical data.

Postulate 2 (Measurements)**.**

Ideal

An ideal measurement procedure is described by a **projector-valued measure** (PVM) $\{\Pi_l\}$ which is a set of projectors that sum to the identity and

(a) each $\{l\}$ is an elementary event in probability theory, where the probability of observing outcome l, given some preparation $|\psi\rangle$, is prescribed by the probability $\Pr(a_l) = \operatorname{Tr}[\Pi_l |\psi\rangle\langle\psi|]$.

(b) the state after measurement result l is observed is $|\psi_{\text{meas}}\rangle = \frac{1}{\Pr(a_l)} \Pi_l |\psi\rangle$.

General

A general discrete-outcome measurement[3] is described by a **positive operator-valued measure** or POVM $\{E_\nu\}$ which is a set of positive operators that sum to the identity $\sum_\nu E_\nu = \mathbb{I}$.

(a) The probability of observing outcome ν given preparation procedure ρ is

$$\Pr(\nu) = \operatorname{Tr}[E_\nu \rho] .$$

(b) Given $F_{\nu,i}$ such that the positive operators $E_\nu = \sum_i F^\dagger_{\nu,i} F_{\nu,i}$, the state after oberving measurement outcome ν is $\rho_{\text{meas}} = \frac{1}{\Pr(\nu)} \sum_i F_{\nu,i} \rho F^\dagger_{\nu,i}$.

The **expectation value** of a self-adjoint operator A given a preparation ρ defined as $\operatorname{Tr}[\rho A]$ is denoted by $\langle A \rangle_\rho$ or $\langle A \rangle$ when the state being used is clear from context. Postulate 2(a), known as the **Born rule** connects the predictions in quantum mechanics to events in probability theory. A physical example of this would be measuring the spin of an electron. The physically observable outcomes would be the value of the spin, ±1/2. The two projections would correspond to clicks in two detectors, one corresponding to spin $+1/2$, and the other corresponding to spin $-1/2$. The orthogonality of the projectors implies that you can always prepare the state such that one detector always clicks, and the other never clicks.

The general measurement procedure can be obtained from the ideal analogously to how the general quantum state arose from the pure state. As an intuitive example, consider a measurement procedure that performs the PVM $\{\Pi_l\}$ with probability p, and the PVM

[3] Continuous-outcome measurements are a bit more mathematically involved that justifies naming these objects as POVMs. The interested reader can refer to chapter 12 of [14].

$\{\tilde{\Pi}_j\}$ with probability $(1-p)$. Assume for simplicity that each of these sets have D elements. So, the probability of an outcome i is given by

$$\Pr(i) = \begin{cases} p\mathrm{Tr}\left[\rho\Pi_i\right] & i \in \{1, \dots, D\} \\ (1-p)\mathrm{Tr}\left[\rho\tilde{\Pi}_i\right] & i \in \{D+1, \dots, 2D\}. \end{cases}$$

We could instead more compactly define

$$E_\nu = \begin{cases} p\Pi_\nu & \nu \in \{1, \dots, D\} \\ (1-p)\tilde{\Pi}_\nu & \nu \in \{D+1, \dots, 2D\}. \end{cases}$$

The resulting set $\{E_\nu\}$ is a POVM. Thus, we see that we get the probabilities $\Pr(\nu) = \mathrm{Tr}\left[E_\nu \rho\right]$ analogous to the Born rule.

2.1.5 Composite systems

Before we talk about transformations, we will introduce composite systems. We have so far gone over the preliminaries to model the preparation and measurement of a single quantum system. The question we now seek to answer is how do we merge our description of multiple systems? In other words, given two systems described by Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, how does the Hilbert space of both the systems together $\mathcal{H}_{AB}$ relate to the the individual Hilbert spaces?

We would expect that if we prepare our individual systems independently, the joint system should be able to accommodate every combination of states from each system individually. So, given a basis $\{|a_i\rangle\}$ and $\{|b_j\rangle\}$ for $\mathcal{H}_A$ and $\mathcal{H}_B$ respectively, $\mathcal{H}_{AB}$ should contain all possible pairs $\{|a_i\rangle, |b_j\rangle\}$. Additionally, since $\mathcal{H}_{AB}$ is a vector space, it must also contain the span of these pairs. The mathematical object that does this, is the **tensor product** denoted as $\otimes$.

The tensor product $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$ is the unique vector space spanned by the vectors $|a_i\rangle \otimes |b_j\rangle$ for all i, j. This forms a basis for $\mathcal{H}_{AB}$. Note that the tensor product is bilinear. The tensor product of operators is defined by its action on the basis and extends to the rest of the space via linearity. Given operators $T \in \mathcal{B}(\mathcal{H}_A, \mathcal{K}_A)$ and $S \in \mathcal{B}(\mathcal{H}_B, \mathcal{K}_B)$, the tensor product $T \otimes S$ is defined by its action $(T \otimes S)\left(|a_i\rangle \otimes |b_j\rangle\right) = (T|a_i\rangle) \otimes (S|b_j\rangle)$. Note that by choosing the operators to be dual vectors of $\mathcal{H}_A$ and $\mathcal{H}_B$, we can also define the inner product on the tensor product space as $(\langle a_k| \otimes \langle b_l|)(|a_i\rangle \otimes |b_j\rangle) = (\langle a_k|a_i\rangle)(\langle b_l|b_j\rangle)$. We denote a system composed of two subsystems as bipartite, three systems as tripartite, and so on. We sometimes write $|a\rangle \otimes |b\rangle$ as $|a, b\rangle$ for brevity.

This is entirely analogous to joint probability distributions, where the space in which the joint probability vector of two systems lies is described by the tensor product of the individual spaces. Just as the trace is analogous to summing over the entire probability distribution $\sum_{i,j} p(i,j)$, the partial trace is analogous to summing over a marginal $\sum_j p(i,j) = p(i)$.

Definition 10 (Partial trace)**.** Given a trace class operator $T \in \mathrm{TC}(\mathcal{H}_A \otimes \mathcal{H}_B)$ and let $\{b_j\}$ be an orthonormal basis for $\mathcal{H}_B$. The **partial trace** of T over system B denoted by $\mathrm{Tr}_B\left[T\right]$ is $\sum_j (\mathbb{I}_A \otimes \langle b_j|)\, T\, (\mathbb{I}_A \otimes |b_j\rangle)$ and its value is independent of the basis chosen.

The partial trace is a way to obtain the marginal states that make up the joint state. However, any correlations will be lost on taking the partial trace as we might expect i.e. in general $T \neq \mathrm{Tr}_B\left[T\right] \otimes \mathrm{Tr}_A\left[T\right]$.

The tensor product of density matrices can exhibit non-classical correlations called entanglement. Entanglement is an important property that for some information processing tasks is a measure of the "quantumness" of the state as it has no classical analogue and allows us to perform tasks such as superdense coding [15] that are impossible to do just via classical probabilities.

Definition 11 (Separable states and entanglement)**.** A bipartite state ρ_{AB} is said to be **separable** if it can be written as

$$\rho_{AB} = \sum_i X_i \otimes Y_i \qquad X_i, Y_i \geq 0\, \forall i.$$

If there is just one term in the sum above, ρ_{AB} is said to be a **product state**.
A bipartite state is said to be **entangled** if it is not separable.

Definition 12 (Purification)**.** Let $\rho \in \mathrm{D}(\mathcal{H}_A)$ be a density operator. Its **purification** $|\rho\rangle$ in a larger Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$ where $\mathcal{H}_A = \mathcal{H}_B$ is defined as

$$\rho = \mathrm{Tr}_B\left[|\rho\rangle\langle\rho|\right]. \tag{2.3}$$

This purification is unique upto a unitary U[4] on the second system. This is another way to obtain the general preparation procedure from an ideal preparation procedure; the

[4] These are maps such that $\mathrm{Tr}\left[UXU^\dagger\right] = \mathrm{Tr}\left[X\right]$. We shall talk about these maps in greater detail in the next section.

general preparation procedure is what we get as a result of only having access to part of a pure quantum state. As a general procedure to obtain a purification given a density operator ρ, consider the diagonal representation of the operator $\rho = \sum_i \lambda_i |i\rangle\langle i|$ where λ_i and $|i\rangle$ are the eigenvalues and eigenvectors of ρ. A purification of ρ can be given as

$$|\rho\rangle = \sum_i \sqrt{\lambda_i} |i\rangle \otimes |i\rangle. \tag{2.4}$$

We can take the partial trace to verify that this is a purification

$$\mathrm{Tr}_B\left[|\rho\rangle\langle\rho|\right] = \sum_{i,j,k} \left(\mathbb{I}_A \otimes \langle k|\right) \left(\sqrt{\lambda_i \lambda_j} |i\rangle\langle j| \otimes |i\rangle\langle j|\right) \left(\mathbb{I}_A \otimes |k\rangle\right) \tag{2.5}$$

$$= \sum_{i,j,k} \sqrt{\lambda_i \lambda_j} \left(|i\rangle\langle j|\right) \otimes \left(\langle k|i\rangle\langle j|k\rangle\right) \tag{2.6}$$

$$= \sum_{i,j,k} \sqrt{\lambda_i \lambda_j} \delta_{i,k} \delta_{j,k} |i\rangle\langle j| \tag{2.7}$$

$$= \sum_i \lambda_i |i\rangle\langle i| = \rho. \tag{2.8}$$

2.1.6 Channels

We will first define another important class of bounded operators.

Definition 13 (Isometry)**.** Let $\mathcal{H}$ and $\mathcal{K}$ be Hilbert spaces. A map $V \in \mathcal{B}(\mathcal{H}, \mathcal{K})$ is called an **isometry** iff $\|Vh\|_{\mathcal{K}} = \|h\|_{\mathcal{H}}$ where $\|\cdot\|_{\mathcal{X}}$ denotes the norm in the Hilbert space $\mathcal{X}$. An isometry $U \in \mathcal{B}(\mathcal{H})$ is called a **unitary** if it is onto. We denote the set of all isometries from $\mathcal{H}$ to $\mathcal{K}$ as $\mathrm{U}(\mathcal{H}, \mathcal{K})$ and the set of all unitaries acting on $\mathcal{H}$ to be $\mathrm{U}(\mathcal{H})$.

Equivalently, one can define an isometry to have the property $V^\dagger V = \mathbb{I}$. The adjoint of a unitary is its inverse $U^\dagger U = UU^\dagger = \mathbb{I}$. Unitaries are the generalisation of rotations for a complex field and carry much of the same intuition.

Before we can discuss transformations on density operators, we also need to talk about **superoperators**, linear maps taking linear operators to linear operators. We denote the set of all superoperators from $\mathcal{L}(\mathcal{H})$ to $\mathcal{L}(\mathcal{K})$ as $\mathrm{T}(\mathcal{H}, \mathcal{K})$.

Definition 14 (Trace preserving)**.** A superoperator $\Phi \in \mathrm{T}(\mathcal{H}, \mathcal{K})$ is said to be **trace preserving** if $\mathrm{Tr}\left[\Phi(X)\right] = \mathrm{Tr}\left[X\right]$ for all $X \in \mathrm{TC}(\mathcal{H})$.

The tensor product of two superoperators can be defined in a similar way to our definition of the tensor product of operators. Given superoperators $\Phi \in \mathrm{T}(\mathcal{H}_A, \mathcal{K}_A)$ and $\Psi \in \mathrm{T}(\mathcal{H}_B, \mathcal{K}_B)$ the tensor product of the operators $\Phi \otimes \Psi \in \mathrm{T}(\mathcal{H}_A \otimes \mathcal{H}_B, \mathcal{K}_A \otimes \mathcal{K}_B)$ is defined as

$$(\Phi \otimes \Psi)(X \otimes Y) = \Phi(X) \otimes \Psi(Y),$$

for all $X \in \mathcal{L}(\mathcal{H}_A)$, $Y \in \mathcal{L}(\mathcal{H}_B)$ and extended linearly to the rest of the space.

Definition 15 (Completely Positive)**.** A superoperator $\Phi \in \mathrm{T}(\mathcal{H}, \mathcal{K})$ is said to be **positive** if $\Phi(P) \geq 0$ for all $P \in \mathrm{Pos}(\mathcal{H})$. We say that $\Phi \geq 0$.

Φ is **completely positive** if $\Phi \otimes \mathbb{I}_{\mathcal{L}(\mathcal{H}_n)} \geq 0$ for all n, where $\mathbb{I}_{\mathcal{L}(\mathcal{H}_n)}$ is the identity superoperator on linear operators acting on an n-dimensional Hilbert space $\mathcal{L}(\mathcal{H}_n)$.

Intuitively, we would like to mathematically describe the class of all physical transformations that would take us from quantum states to quantum states. This would take the form of a linear operator in the ideal case and a superoperator in the general case. Keeping this in mind, we state the postulate for quantum transformations.

Postulate 3 (Quantum transformations)**.**
Ideal

An ideal quantum transformation can be represented by a unitary operator U as

$$|\psi_f\rangle = U|\psi_i\rangle$$

where $|\psi_i\rangle, |\psi_f\rangle \in \mathcal{H}$ are the initial and final pure states respectively. This is sometimes referred to as unitary evolution.

General

A **quantum channel** or channel for short is a completely positive trace preserving (CPTP) superoperator. A general quantum transformation can be represented by a quantum channel Φ as

$$\rho_f = \Phi(\rho_i)$$

where $\rho_i \in \mathrm{D}(\mathcal{H})$ and $\rho_f \in \mathrm{D}(\mathcal{K})$ are the initial and final states respectively.

For the ideal case, the operator must preserve the norm as all states have unit norm. Thus, the transformation must be characterised by a unitary operator. For the general case, we need to preserve trace and the positivity of our operators. This might seem to indicate that we only need positive trace preserving maps. However, the stronger constraint of complete positivity comes from observing that we can apply a transformation Φ only to a subsystem of a composite system while not doing anything to the other subsytem represented by the identity map $\mathbb{I}_{\mathcal{L}(\mathcal{H}_n)}$. This is a physical transformation and so the output must be a physical state i.e. a density operator. Complete positivity is needed to guarantee this. We denote the set of all channels from $\mathrm{TC}(\mathcal{H})$ to $\mathrm{TC}(\mathcal{K})$ as $\mathrm{C}(\mathcal{H},\mathcal{K})$.

An example of a channel we have already seen is the trace $\mathrm{Tr} \in \mathrm{C}(\mathcal{H},\mathbb{C})$. Thus, the partial trace $\mathrm{Tr}_B \in \mathrm{C}(\mathcal{H}_A \otimes \mathcal{H}_B, \mathcal{H}_A)$ can be written as a tensor product of the identity map with the Trace $\mathrm{Tr} \in \mathrm{C}(\mathcal{H}_B,\mathbb{C})$ as

$$\mathrm{Tr}_B = \mathbb{I}_{\mathcal{L}(\mathcal{H}_A)} \otimes \mathrm{Tr}. \tag{2.9}$$

The adjoint of a channel is defined similarly to the adjoint of an operator. Given a channel $\Phi \in \mathrm{C}(\mathcal{H},\mathcal{K})$, its adjoint is the unique superoperator $\Phi^\dagger : \mathcal{B}(\mathcal{K}) \to \mathcal{B}(\mathcal{H})$ such that

$$\langle \Gamma, \Phi(\rho)\rangle = \langle \Phi^\dagger(\Gamma), \rho\rangle, \tag{2.10}$$

for all $\rho \in \mathrm{TC}(\mathcal{H})$ and $\Gamma \in \mathcal{B}(\mathcal{K})$ and the inner product is defined as $\langle A, B\rangle = \mathrm{Tr}\left[A^\dagger B\right]$.

There are a few different equivalent representations of channels. The **Kraus representation** of a channel $\Phi \in \mathrm{C}(\mathcal{H},\mathcal{K})$ is

$$\Phi(X) = \sum_i K_i X K_i^\dagger, \tag{2.11}$$

where $X \in \mathcal{B}(\mathcal{H})$. The $K_i \in \mathcal{B}(\mathcal{H},\mathcal{K})$ are **Kraus operators** and $\sum_i K_i^\dagger K_i = \mathbb{I}$[5]. Given a basis $\{|e_a\rangle\}$ of the Hilbert space $\mathcal{H}$, the **Choi–Jamiołkowski isomorphism** or Choi representation of a channel $\Phi \in \mathrm{C}(\mathcal{H},\mathcal{K})$ is[6]

$$J = \sum_{a,b} |e_a\rangle\langle e_b| \otimes \Phi(|e_a\rangle\langle e_b|), \tag{2.12}$$

where $J \in \mathrm{Pos}(\mathcal{H}\otimes\mathcal{K})$ and $\mathrm{Tr}_{\mathcal{K}}\left[J\right] = \mathbb{I}_{\mathcal{H}}$[7]. A useful result for the Choi representation that

[5] In infinite dimensions, the notion of equality used is convergence in the strong topology. However, these nuances are not important for this thesis and we shall not mention it outside this footnote.

[6] There are some nuances in infinite dimensions that are not entirely relevant to us. The interested reader can refer to [16] for a more precise infinite dimensional generalisation.

[7] Note that this is an example of an unbounded operator. However, when we use the Choi representation in this thesis we shall use no more than the few defining properties mentioned here before taking finite projections.

we will use often is

$$\mathrm{Tr}\left[\Gamma\Phi(\rho)\right] = \mathrm{Tr}\left[\,\left(\rho^T \otimes \Gamma\right)\, J\right] \tag{2.13}$$

for all $\rho \in \mathrm{TC}(\mathcal{H})$ and $\Gamma \in \mathcal{B}(\mathcal{K})$.

2.1.7 Entropic and distance measures

We will briefly motivate and define several entropic quantities and matrix norms which form a crucial part of most information processing tasks.

Entropy

Classical, or **Shannon entropy** [17] is a measure of the uncertainty or information content of classical probability distributions. It must be positive and additive for independent probability distributions. The function that satisfies these properties (see Appendix A of [18] for a short proof) is

$$\mathrm{H}(p) = -\sum_i p(i)\log(p(i)). \tag{2.14}$$

The quantum analog, which attempts to measure the uncertainty in a quantum state is called the **von Neumann entropy** and is given by the Shannon entropy of its eigenvalues

$$\mathrm{H}(\rho) = \mathrm{H}(\lambda(\rho)) = -\mathrm{Tr}\left[\rho\log(\rho)\right], \tag{2.15}$$

where $\lambda(\rho)$ is the vector of the eigenvalues of ρ. We often denote the entropy by the system that the state describes or **register**, $\mathrm{H}(\rho_{AB}) = \mathrm{H}(AB)_{\rho_{AB}}$ where we omit mention of the state if it is clear from context what it is. Similarly, we denote the entropy of the marginals as $\mathrm{H}(A) = \mathrm{H}(\mathrm{Tr}_B\left[\rho_{AB}\right])$ and $\mathrm{H}(B) = \mathrm{H}(\mathrm{Tr}_A\left[\rho_{AB}\right])$.

The **conditional entropy** defined as

$$\mathrm{H}(A|B)_{\rho_{AB}} = \mathrm{H}(AB)_{\rho_{AB}} - \mathrm{H}(B)_{\rho_B} \tag{2.16}$$

is the amount of information needed to describe A given knowledge of B. The **mutual information** measures the inherent dependence expressed in the joint state of X and Y

relative to the marginal states of X and Y and is defined as

$$I(X:Y) = \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(XY). \tag{2.17}$$

If the states can be represented as a product of their marginals[8], then the mutual information would be zero and the conditional entropy would be given by $H(X|Y) = H(X)$, since knowledge of Y tells us nothing about X.

The **quantum relative entropy**, much like its classical counterpart, is a measure of distance between two quantum states and is given by

$$\mathrm{D}(\rho||\sigma) = \begin{cases} \mathrm{Tr}\left[\rho\log(\rho) - \rho\log(\sigma)\right] & \text{if } \mathrm{supp}(\rho) \subseteq \mathrm{supp}(\sigma) \\ \infty & \text{otherwise} \end{cases} \tag{2.18}$$

where $\mathrm{supp}(\rho)$ is the support of ρ. Note that it is not a formal metric as it is not symmetric.

Distance measures

We use distance measures as a quantitative estimate of how similar or different two states are. Given two density matrices ρ and σ, the maximum probability with which one can correctly distinguish [19, 20] between them is $\frac{1}{2} + \frac{1}{2}\left\|\rho - \sigma\right\|_1$. Here the **one-norm** of an operator $\left\|X\right\|_1$ is defined as

$$\left\|X\right\|_1 = \mathrm{Tr}\left[\sqrt{X^\dagger X}\right]. \tag{2.19}$$

When X is positive and trace-class, $\left\|X\right\|_1 = \sum_i \lambda_i(X)$.

Although the one-norm is physically motivated in terms of distinguishing two density matrices, it lacks a number of convenient mathematical properties that make it hard to compute and use sometimes, for e.g. $\left\|X \otimes Y\right\|_1 \neq \left\|X\right\|_1 \left\|Y\right\|_1$. We often use the fidelity for its convenient mathematical properties. The **fidelity** of two positive semidefinite operators $P, Q \geq 0$ is

$$\mathrm{F}(P, Q) = \mathrm{Tr}\left[\sqrt{\sqrt{P}Q\sqrt{P}}\right]. \tag{2.20}$$

[8] Recall from Section 2.1.5 that this is analogous to X and Y being described by independent probability distributions.

The fidelity of two density matrices is related to the one-norm via the Fuchs-van de Graaf inequalities (FvdG) [21]

$$1 - \mathrm{F}(\rho,\sigma) \leq \frac{1}{2}\left\|\rho - \sigma\right\|_1 \leq \sqrt{1 - \mathrm{F}(\rho,\sigma)^2}. \tag{2.21}$$

This can be tighter in the case where $\sigma = \Pi\rho\Pi$ is a projection of the state ρ (Lemma 5 of [22])

$$\frac{1}{2}\left\|\rho - \Pi\rho\Pi\right\|_1 \leq \sqrt{1 - \mathrm{Tr}\left[\Pi\rho\right]}. \tag{2.22}$$

2.2 Semidefinite programming

Semidefinite programs (SDPs) are widely applicable in quantum cryptographic tasks [23]. These are a class of optimisation problems that are generally numerically efficient to compute when finite-dimensional. Let $\mathcal{H}$ and $\mathcal{K}$ be finite-dimensional[9] Hilbert spaces. Let $A \in \mathrm{Herm}(\mathcal{H})$ and $B \in \mathrm{Herm}(\mathcal{K})$ be a Hermitian operators, and $\Phi \in \mathrm{T}(\mathcal{H},\mathcal{K})$ be a linear Hermitian-preserving map. An **SDP** is a triple (Φ, A, B) with which the following optimisation problem is associated:

$$\begin{gathered} \max\langle A, X\rangle \\ \text{such that}\Phi(X) = B \\ X \geq 0, \end{gathered} \tag{2.23}$$

where $\langle A, X\rangle = \mathrm{Tr}\left[A^\dagger X\right]$. The function that we optimise $\langle A, X\rangle$ is called the objective function. We define the **feasible** set $\mathcal{S}$ as

$$\mathcal{S} = \{X \in \mathrm{Pos}(\mathcal{H}) \mid \Phi(X) = B\}.$$

Note that the SDP can be equivalently be made into a minimisation problem by replacing A with $-A$.

[9] We only define finite-dimensional SDPs. This is because although we do formulate infinite-dimensional SDPs in this thesis, we do not use any of the usual properties like duality for SDPs, and we only numerically compute finite-dimensional SDPs. Thus, the infinite-dimensional details are not very important for this thesis and it is sufficient to just understand what a finite-dimensional SDP is as it is easy to numerically compute.

2.3 Quantum key distribution

QKD is a quantum information processing task where a sender (Alice) attempts to share a random bit string (key) with a receiver (Bob) in the presence of a malicious eavesdropper (Eve). Alice and Bob attempt to ensure that Eve has no information about their shared key without them detecting it. If Eve does obtain information about parts of the key, Alice and Bob can always discard it (abort the protocol) ensuring that if they do share a key, it is **secret**. In order to accomplish this task, Alice and Bob have access to two channels. A classical channel which Eve also receives, but cannot change (authenticated classical channel[10]), and a quantum channel which Eve can interact with in any way she likes (insecure).

Unsurprisingly, this task is impossible[11] to accomplish if we replace the insecure quantum channel with an insecure classical channel as Eve could simply copy any data that Alice sends Bob, and then imitate Bob to obtain the same key that Bob has; all without Alice or Bob's knowledge. However, this task is indeed possible by sending quantum states in a set of conjugate bases that are not orthogonal to each other [26]. The first QKD protocol [27], named BB84 after the two authors along with the year the protocol was developed, is often used as a blue-print for a large class of other QKD protocols. We shall look at one such variant, the 3-state protocol in some detail in a later chapter.

In this section we will review the basic steps involved in a basic prepare and measure (PM) protocol and explain how it connects to an entanglement-based (EB) protocol via a general technique called the source-replacement scheme. We then briefly review the security framework that allows us to calculate reliable lower bounds of key rates.

2.3.1 Prepare and measure protocols

Here we outline the steps in a generic PM protocol.

1. **Signal Preparation:** Alice randomly prepares one of a set of quantum states $\{\rho_1 \dots \rho_n\}$ with an apriori probability distribution $\{p_i\}$ where i denotes which state

[10] An authenticated classical channel can be implemented over an insecure classical channel by sharing a short secret key [24]. Thus, QKD is often referred to as growing secret keys rather than generating secret keys.

[11] Classical key distribution relies on making computational assumptions like in [25] and imposing restrictions on what Eve can do. This is called computationally secure as opposed to the information theoretically secure key distribution that QKD hopes to accomplish.

she chose. These states are called signal states. Without loss of generality we can always model Alice's signal preparation procedure as isometries acting on some fixed state $\rho_i = V_i \rho V_i^\dagger$. We denote the quantum system associated with each of these signal states A'_m for the m^{th} round of the protocol.

2. **Signal Transmission:** Alice sends her prepared states to Bob via the *insecure* quantum channel Φ_m where m denotes the round of the protocol. We denote the quantum system associated with each of the states Bob recieves as B_m.

3. **Measurement:** Bob measures the states that he receives by a k-outcome POVM $\{\Gamma_j\}_{j=1}^k$ and records the outcome from each round.

After repeating the above steps multiple times, we proceed to the next part of the protocol.

4. **Parameter estimation:** Alice and Bob randomly choose a subset of the rounds for testing or parameter estimation. They both reveal the recorded values of state chosen i and measurement outcome j to form a probability[12] distribution $p(i,j) = p_i \mathrm{Tr}\left[\Gamma_j \Phi_m(\rho_i)\right]$. This probability distribution effectively constrains Φ_m, and thus Eve's actions on the states that Alice sent Bob. Based on these constraints, if the probability distribution belongs to an agreed upon set they proceed with the protocol after discarding the test results. Otherwise, they abort.

5. **Announcements and sifting:** Alice and Bob make announcements over the authenticated classical channel. They sift the data based on the announcements made i.e. they choose a subset of signal and measurement data to keep and discard the rest based on the announcements. This is also referred to as post-selection.

6. **Key Map:** Alice[13] uses her signal state data i as well as the announcements to map it into a key string x. This is called the raw key. We assume here that the key is a bit string for simplicity, but all the steps can be applied more generally.

[12] This would actually be a frequency distribution instead of a probability distribution for a finite test set. However, we only consider the asymptotic limit in this thesis. For more of a discussion on finite size effects within the numerical framework we discuss in later sections, the interested reader can refer to [28]. Additionally, there are difficulties finding enough statistics to actually construct a probability distribution when the channel used in each round is different. We will address these difficulties in Section 2.3.3

[13] Bob could also perform the key map. This is called reverse reconciliation. In that case Alice and Bob's roles in the next steps are reversed.

7. **Error Correction:** Bob maps his measurement data into a key string y similarly to how Alice formed x. Alice and Bob then perform error correction over the authenticated classical channel to make y match with x. We denote the data leaked per key bit to Eve in this process as δ_{leak}.

8. **Privacy Amplification:** Alice and Bob produce their final secret key by applying a two-universal hash function on the raw key (Theorem 5.5.1 of [29]).

PM protocols are typically implemented in experiments. However, it is easier to analyse the security of another class of protocols, EB protocols where Alice and Bob share an entangled bipartite state instead of step 1 of the PM protocol. However, as we shall detail in the next section, for any PM protocol we can construct an EB protocol with the same key rate[14] via a source-replacement scheme.

2.3.2 Source-replacement schemes

A source-replacement scheme is a general tool that we can use in analysing QKD protocols in order to simplify the analysis. The general idea is to replace the real source[15] with another virtual source that is easier to analyse. We need to ensure that the virtual source either does not affect the security of the protocol or it makes the protocol less secure. So, if we use the virtual source in our security proof, this would always imply that the real source is also secure. As a more concrete example, we shall demonstrate how a PM protocol can be replaced by an EB protocol [30, 31].

PM protocol replaced by EB protocol

We shall use the notation used for the PM protocol above where Alice prepares the state ρ_i with probability p_i to send to Bob. We can instead replace the real source with a virtual source for Alice that instead prepares the entangled state

$$|\psi\rangle_{AA_SA'} = \sum_{i=1}^{n} \sqrt{p_i}|i\rangle_A \otimes |\rho_i\rangle_{A_SA'}. \tag{2.24}$$

Recall the notation used in section 2.1.5, $|\rho_i\rangle_{A_SA'}$ is the purification of the density matrix $\rho_i \in \mathrm{D}(\mathcal{H}_{A'})$ where $\mathrm{Tr}_{A_S}[|\rho_i\rangle\langle\rho_i|] = \rho_i$. The states $|i\rangle$ form an orthonormal basis for the

[14] We shall explain precisely what we mean by key rate in the following sections.

[15] Or more accurately, our model of the physical source.

space $\mathcal{H}_A$. The system corresponding to the Hilbert space $\mathcal{H}_{A_S}$ is termed the shield system [32] and is only needed if the signal states Alice sends Bob are mixed. Neither Alice nor Bob interact with the shield system at any point. She then sends the portion of the state corresponding to the Hilbert space $\mathcal{H}_{A'}$ to Bob through the insecure quantum channel. For notational simplicity, we define $|\psi_i\rangle = |i\rangle_A \otimes |\rho_i\rangle_{A_S A'}$.

Now, Alice can measure her system A with the PVM $\{|i\rangle\langle i|_A\}$. A measurement result i for Alice that occurs with probability p_i as

$$\mathrm{Tr}\left[\left(|i\rangle\langle i|_A \otimes \mathbb{I}_{A_S A'}\right) |\psi\rangle\langle\psi|\right] = p_i \mathrm{Tr}\left[|\psi_i\rangle\langle\psi_i|_{AA_S A'}\right] = p_i$$

ensures that the state that was sent to Bob is ρ_i as

$$\frac{1}{p_i}\mathrm{Tr}_{AA_S}\left[\left(|i\rangle\langle i|_A \otimes \mathbb{I}_{A_S A'}\right) |\psi\rangle\langle\psi|\right] = \frac{p_i}{p_i}\mathrm{Tr}_{AA_S}\left[|\psi_i\rangle\langle\psi_i|\right] \tag{2.25}$$

$$= \mathrm{Tr}_{A_S}\left[|\rho_i\rangle\langle\rho_i|\right] \tag{2.26}$$

$$= \rho_i. \tag{2.27}$$

Thus, the probabilities obtained by parameter estimation in step 4 of the protocol is given by $p(i,j) = \mathrm{Tr}\left[\left(|i\rangle\langle i|_A \otimes \mathbb{I}_{A_S} \otimes \Gamma_j\right) \rho_{AA_S B}\right]$ where $\rho_{AA_S B} = \left(\mathbb{I}_{\mathcal{L}(\mathcal{H}_{AA_S})} \otimes \Phi\right)(|\psi\rangle\langle\psi|)$. In addition to this, while deciding whether or not to abort we must keep in mind that the systems A and A_S do not leave Alice's laboratory and so Eve has no access to these systems. So we have the additional constraint

$$\rho_{AA_S} = \mathrm{Tr}_B\left[\rho_{AA_S B}\right] = \mathrm{Tr}_{A'}\left[|\psi\rangle\langle\psi|\right] \tag{2.28}$$

with which to constrain Eve's actions. We have shown that this EB source is able to send the states ρ_i to Bob with the apriori probability distribution p_i exactly like the PM source. So this EB protocol is equivalent to the corresponding PM protocol described in the previous section.

Source maps

We will now describe a class of source-replacement schemes that give Eve more power than she has in reality. This is useful because it greatly simplifies the security analysis in some cases while still giving good bounds on the key rates. This is a technique commonly used in QKD security proofs with ideas similar to squashing maps [33].

Let $\{\rho_i\} \in \mathrm{D}(\mathcal{H}_{A'^{\otimes N}})$ and $\{\tau_i\} \in \mathrm{D}(\mathcal{K}_{A'^{\otimes N}})$ be a set of states such that $\rho_i = \Psi(\tau_i)$ for all

i and for some channel $\Psi \in \mathrm{C}(\mathcal{K}_{A'^{\otimes N}}, \mathcal{H}_{A'^{\otimes N}})$ called a **source map**. If any QKD protocol that uses $\{\tau_i\}$ as signal states is secure, the corresponding QKD protocol that uses $\{\rho_i\}$ is secure. Given the states $\{\tau_i\}$ ($\{\rho_i\}$), let Eve's optimal eavesdropping strategy for N rounds cause the signal states to change as per the channel $\Phi_{\text{virtual}} \in \mathrm{C}(\mathcal{K}_{A'^{\otimes N}}, \mathcal{H}_{B^{\otimes N}})$ ($\Phi_{\text{real}} \in \mathrm{C}(\mathcal{H}_{A'^{\otimes} N}, \mathcal{H}_{B^{\otimes} N})$). Suppose that the protocol using $\{\tau_i\}$ is secure and Eve's eavesdropping strategy corresponding to the channel Φ_{virtual} does not give her enough information to prevent secret key generation[16]. Then, Eve's eavesdropping strategy corresponding to the channel $\Psi \circ \Phi_{\text{real}}$ certainly gives Eve no more information than Φ_{virtual}, since the application of the map Ψ is an additional restriction on Eve's more general attacks whose optimal is Φ_{virtual}. Thus, the protocol using $\{\rho_i\} = \{\Psi(\tau_i)\}$ is secure[17].

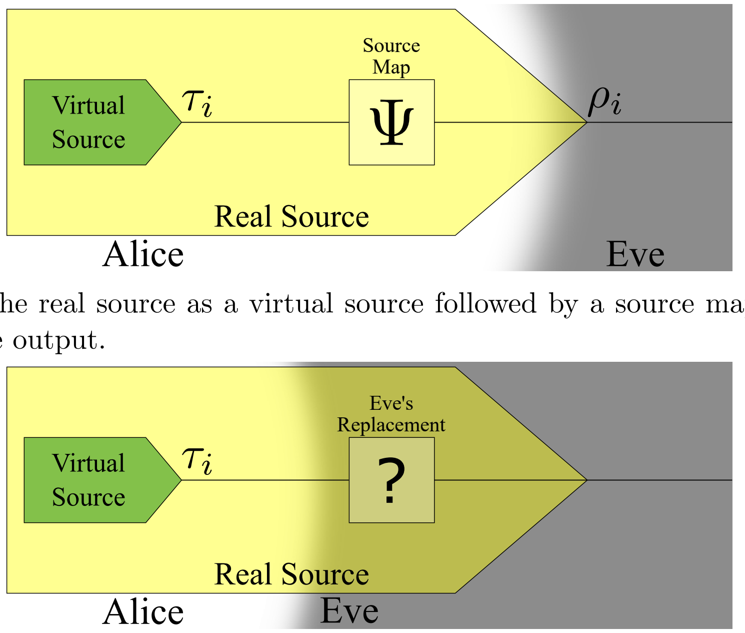
We can use this with a source-replacement as follows. Let Alice's real source produce the states $\{\rho_i\}$. We can equivalently represent the action of this source as producing the states $\{\tau_i\}$, followed by an application of the source map Ψ since $\rho_i = \Psi(\tau_i)$ for all i.. We then replace this source with our virtual source simply producing the states $\{\tau_i\}$ by giving Eve access to the part of our source that applied the channel Ψ. Eve can choose to honestly implement the channel and reproduce the real source, or Eve could choose not to implement it and do something else that might benefit her more. Thus, proving that the virtual source gives us a secret key is sufficient to prove that the real source can gives us a secret key. This method is generally useful when we suspect that Eve's optimal attack involves the application of Ψ, or if we expect Eve's optimal replacement to Ψ to not give her too much information. This conceptual process is depicted in Fig. 2.1. We shall use some concrete examples of this rather general proof technique in this thesis to simplify our analysis.

2.3.3 Security framework

The central problem we attempt to solve in QKD is to find the number of secret key bits for each signal sent that we can extract from a family of QKD protocols that send a different number of signals. This is called the **key rate** of the family of protocols. In this thesis we consider only the asymptotic limit of this family where we send an infinite number of signals. This is a simple limit that helps us compare different protocols and provides us with intuition on what helps us increase the key rate and so has pedagogical value even if

[16] We shall formalise this in later sections when we discuss the key rate by specifying quantitatively what we mean by Eve gaining partial information about the key.

[17] A more formal proof of this would be very similar to the proof of the security for squashing models found in Theorem 3.3.1 of [34]. However, the technical details are not very important for this thesis and so we omit giving an explicit proof of the fact.



(a) We can model the real source as a virtual source followed by a source map since they both have the exact same output.

(b) Once we give Eve control of the source map, she can perform any physical operation on the output of the virtual source, including the source map Ψ if reproducing the real state is beneficial to her.

Figure 2.1: The real source can always be replaced by the virtual source in security proofs if they are related via a source map since the virtual source gives Eve more power.

the end goal is to find the finite-size key rates. Additionally, the finite-size key rates [35] can often be obtained by modifying the asymptotic key rate formula.

Restricting Eve's attacks

Another simplification that we make is restricting Eve to **collective attacks**. This means that Eve interacts with each signal in the same way, i.e. the insecure quantum channel $\Phi_m = \Phi$ is the same for all rounds of the protocol. She stores the results of her interactions in a quantum memory in her system E and at the end of the protocol, she uses the classical information she gained via Alice and Bob's announcements to dictate her collective measurement on her system. If Alice's state preparation are taken from an iid probability distribution, then this leaves their joint state involving all rounds of the protocol in a tensor product of the joint state from each protocol round $\rho_{AB}^{\otimes N}$. This makes things significantly simpler as we can deal with the state ρ_{AB} which is significantly smaller than $\rho_{AB}^{\otimes N}$ especially in the asymptotic limit when $N \to \infty$.

This is in contrast to **coherent attacks** where Eve interacts with all the signals in possibly different ways. Here the joint state $\rho_{A^{\otimes N}B^{\otimes N}}$ need not be a product of the individual rounds ρ_{AB}. If in addition to Alice's state preparation being iid, Bob's measurements are also independent of the round security proofs against collective attacks can be lifted to security proofs against coherent attacks by various techniques such as the entropic uncertainty principle approach [36], post-selection technique [37] or quantum de Finetti theorem [35]. These techniques all effect the finite-size contributions, but largely leave the asymptotic key rates unchanged.

Asymptotic key rate

Under this iid assumption, we can consider each round of the QKD protocol separately to get the asymptotic key rate[18] from the Devetak-Winter formula [38]

$$R^{\infty} = \mathrm{H}(Z|E) - \mathrm{H}(Z|B) \tag{2.29}$$

where E is the system containing Eve's information of the key register Z including the classical announcements made in step 5 of the protocol, and B is Bob's[19] register. Intuitively, the first term is the amount of information that Eve needs about the key register to completely determine the key $\mathrm{H}(Z|E)$. The second term is the amount of information that would need to be communicated over the authenticated classical channel in order to perform error correction so that Alice and Bob have the same keys. As this information is available to Eve, we subtract this from the key rate[20]. In practice, error correction protocols used typically do not achieve this Shannon limit. We denote the information leaked due to error correction $\delta_{\text{leak}} \geq \mathrm{H}(Z|B)$. Thus, a more realistic bound on the key rate can be given by

$$R^{\infty} = \mathrm{H}(Z|E) - \delta_{\text{leak}}. \tag{2.30}$$

We can rewrite the first term $\mathrm{H}(Z|E)$ only as a function of the state Alice and Bob

[18] This is only the key rate if we know Eve's state exactly. If we do not know Eve's exact state then we would need to consider all possible states that Eve could have and assume that she holds the one that minimises this formula, since we need to guarantee that any key we produce is secret.

[19] Alice's if Bob performs the key map.

[20] This might not be tight as some of this information might already be known to Eve from Alice and Bob's announcements in step 5 of the protocol and already considered in the first term. However, tightening this bound is an open problem and we do not consider these intricacies in this thesis.

share ρ_{AA_SB} [39, 40] as

$$f(\rho_{AA_SB}) = \mathrm{D}(\mathcal{G}(\rho_{AA_SB})||\mathcal{Z}(\mathcal{G}(\rho_{AA_SB}))) \tag{2.31}$$

where $\mathcal{G}$ is a completely positive trace non-increasing map that represents the measurement, announcements, sifting, and key map steps of the protocol. $\mathcal{Z}$ is a **completely dephasing map** that acts on the key register Z by deleting all off-diagonal elements represented by Kraus operators $K_i = |i\rangle\langle i|_Z \otimes \mathbb{I}$. Although we do not know the exact state ρ_{AA_SB} as Alice's signals were acted on by the unknown channel before reaching Bob, we have Alice and Bob's measurements results which constrain the set $\mathbf{S}$ of all possible ρ_{AA_SB} compatible with the constraints described in section 2.3.2. So, we consider the smallest key rate that is compatible with these constraints. This leads us to the optimisation problem

$$\begin{aligned} R^\infty = \min & f(\rho_{AA_SB}) - \delta_{\text{leak}} \\ \text{such that } & \mathrm{Tr}\left[\left(|i\rangle\langle i|_A \otimes \mathbb{I}_{A_S} \otimes \Gamma_j\right)\ \rho_{AA_SB}\right] = p(i,j) \\ & \mathrm{Tr}_B\left[\rho_{AA_SB}\right] = \rho_{AA_S} \\ & \rho_{AA_SB} \in \mathrm{D}(\mathcal{H}_{AA_SB}). \end{aligned} \tag{2.32}$$

Since δ_{leak} is a constant that depends only on the statistics and not the actual quantum state, it does not affect the minimisation. Thus, for the purpose of this thesis it shall be sufficient to consider the case when $\delta_{\text{leak}} = 0$. When the spaces $\mathcal{H}_A$, $\mathcal{H}_{A_S}$ and $\mathcal{H}_B$ are finite, this can be lower bounded by an SDP [39] and the minimum can be numerically computed. However, when they are infinite we need to use different techniques that involve taking finite projections [1] or squashing models [33] to first make this into a finite-dimensional SDP that can be numerically solved.

More specifically, the finite projection technique enables us to take projections in these infinite dimensional spaces and loosen the constraints to form an SDP whose minimum will lower bound the minimum of the infinite dimensional SDP. In this thesis we will project onto Bob's space with a projection Π_N that commutes[21] with the POVM elements

[21] This is not needed for our methods to work although it does give better bounds. This assumption is not too restrictive as the commonly used measurement device, threshold detectors do commute with projections in the Fock basis. More on threshold detectors in the next section.

$[\Pi_N, \Gamma_j] = 0$. In this case we can use Eq. (49) from [1] to get the SDP

$$\begin{aligned}
&R^N = \min f(\rho^N_{AA_SB}) \\
&\text{such that } p(i,j) - W \leq \mathrm{Tr}\left[\left(|i\rangle\langle i|_A \otimes \mathbb{I}_{A_S} \otimes \Gamma^N_j\right)\, \rho_{AA_SB}\right] \leq p(i,j) \\
&\qquad\qquad \mathrm{Tr}_B\left[\rho^N_{AA_SB}\right] \leq \rho_{AA_S} \\
&\qquad\qquad 1 - W \leq \mathrm{Tr}\left[\rho^N_{AA_SB}\right] \leq 1 \\
&\qquad\qquad \rho^N_{AA_SB} \in \mathrm{Pos}(\mathcal{H}_{AA_SB}).
\end{aligned} \tag{2.33}$$

where we have defined $\rho^N_{AA_SB} = (\mathbb{I}_{AA_S} \otimes \Pi_N)\, \rho_{AA_SB}\, (\mathbb{I}_{AA_S} \otimes \Pi_N)$ and $\Gamma^N_j = \Pi_N \Gamma_j \Pi_N$. We have also used the fact that $\mathrm{Tr}\left[\Pi A \Pi\, B\right] = \mathrm{Tr}\left[\Pi A \Pi\, \Pi B \Pi\right]$ for any projection Π to get Γ^N_j in the trace constraints. W is a parameter that needs to be estimated that signifies the weight of ρ_{AA_SB} that lies outside the subspace we are projecting on i.e. $W \geq 1 - \mathrm{Tr}\left[\rho^N_{AA_SB}\right]$.

If in addition, each of the signal states can be written as a direct sum $\rho_i = \bigoplus_{\tilde{n}=1}^{\infty} p_{\tilde{n}} \rho_i^{\tilde{n}}$ where the direct sum structure is the same for all the states, then we can write the state $\rho_{AA_SB} = \sum_{\tilde{n}=0}^{\infty} p_{\tilde{n}} \rho^{\tilde{n}}_{AB}$ (See Eq. (D.6) from [34]). It then follows (Eq. (D.9) from [34]) that we can break up the objective function $f(\rho_{AA_SB}) = \sum_{\tilde{n}=0}^{\infty} p_{\tilde{n}} f(\rho^{\tilde{n}}_{AB})$ as a sum of positive numbers. Thus, taking finitely many of these terms is sufficient to lower bound the key rate. In practice, just one of these terms is usually enough to give a good bound on the key rate for most protocols. Now, if we could find the statistics to constrain each of these terms as $Y^L_{\tilde{n}}(i,j) \leq \mathrm{Tr}\left[\rho^{\tilde{n}}_{AB} \Gamma^N_j\right] \leq Y^U_{\tilde{n}}(i,j)$, we could obtain the set of SDPs

$$\begin{aligned}
&R^N_{\tilde{n}} = \min\, p_{\tilde{n}} f(\rho^{\tilde{n}N}_{AB}) \\
&\text{such that } Y^L_{\tilde{n}}(i,j) \leq \mathrm{Tr}\left[\left(|i\rangle\langle i|_A \otimes \Gamma^N_j\right)\, \rho^{\tilde{n}N}_{AB}\right] \leq Y^U_{\tilde{n}}(i,j) \\
&\qquad\qquad \mathrm{Tr}_B\left[\rho^{\tilde{n}N}_{AB}\right] \leq \rho^{\tilde{n}}_A \\
&\qquad\qquad 1 - W \leq \mathrm{Tr}\left[\rho^{\tilde{n}N}_{AB}\right] \leq 1 \\
&\qquad\qquad \rho^{\tilde{n}N}_{AB} \in \mathrm{Pos}(\mathcal{H}_{AB}).
\end{aligned} \tag{2.34}$$

Note that solving each of these SDPs independently could introduce some looseness since we do not take into account the fact that the constraints of different blocks could be correlated. This is because the minimum of each of these SDPs might independently be achieved at points that together violate the constraints in Eq. (2.33). Thus, $R^N \geq \sum_{\tilde{n}=0}^{\infty} R^N_{\tilde{n}}$ where each term in the sum is positive. So we can take any finite cut-off for this sum to solve finitely many of these optimisations to get a reliable lower bound on the key rate. Thus, the central problems we try to solve in this thesis are obtaining the direct sum structure for the signal states, and finding these bounds $Y^L_{\tilde{n}}(i,j)\,\left(Y^U_{\tilde{n}}(i,j)\right)$.

2.4 Quantum optics

The most common practical implementations of QKD protocols use states produced by lasers, and optical devices to manipulate the states and encode the key information in them. So in order to understand practical QKD, we first need to understand quantum optics. We give a very brief introduction to lasers and the optical components we use in this thesis. Another great reference for a brief overview of the quantum optics needed for this thesis is [41].

2.4.1 Fock states

The energy of an electromagnetic (EM) field can be shown to be quantized[22]. Each quanta or excitation of energy is called a **photon**. We call the Hilbert space vector describing the preparation procedure for n photons a **Fock state** or a number state $|n\rangle$. The set of all Fock states $\{|n\rangle\}_{n=0}^{\infty}$ form an orthonormal basis of the Hilbert space for the EM field. Note that $|0\rangle$ is a unit vector representing an EM field with no photons and is not the zero vector. We call this the **vacuum state**. We sometimes implicitly write the tensor product of vacuum states $|0,0\rangle$ as $|0\rangle$.

A pair of useful operators connecting the number states are the **ladder operators** a and $a^\dagger$. The **creation** $a^\dagger$ and **annihilation** operator a can be thought of as operators that create and destroy photons respectively as

$$a^\dagger|n\rangle =\sqrt{n+1}|n+1\rangle \tag{2.35}$$

$$a|n\rangle =\sqrt{n}|n-1\rangle. \tag{2.36}$$

Note that the action of the annihilation operator on the vacuum state gives the zero vector, $a|0\rangle = 0$. These operators are not Hermitian, and so do not correspond to physical observables. However, we can often write measurements and observables in terms of the ladder operators whose action on Fock states is easy to compute and intuitive.

The **commutator** of two operators A and B defined as $[A,B] = AB - BA$ is an important quantity throughout quantum mechanics. The ladder operators obey the commutation relation $[a, a^\dagger] = 1$ where 1 is the identity on the Hilbert space.

[22] For a more complete formulation of the quantization of the EM field see chapter 2 of [42].

2.4.2 Coherent states

Coherent states $\{\alpha\}_{\alpha\in\mathbb{C}}$ are an important class of states in quantum optics as they can be easily and cheaply produced by lasers. These are eigenstates of the annihilation operator

$$|\alpha\rangle = \alpha|\alpha\rangle \tag{2.37}$$

where the eigenvalue α is called the amplitude of the coherent state. Note that the coherent state $|0\rangle$ corresponds exactly to the Fock state $|0\rangle$. Coherent states span the Hilbert space, but are not orthogonal. We can write the coherent state in the Fock basis as

$$|\alpha\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}}|n\rangle \tag{2.38}$$

giving us the overlap between two coherent states $\langle\alpha|\beta\rangle = e^{i\mathrm{Im}(\alpha^*\beta)}e^{-\frac{|\beta-\alpha|^2}{2}}$.

Given a coherent state $|\alpha\rangle$, its phase is the phase θ of the complex number $\alpha = \sqrt{\mu}e^{i\theta}$ and its intensity $\mu = |\alpha|^2$. The probability of measuring n-photons in a coherent state with intensity μ is given by a Poissonian distribution

$$\begin{aligned} p_\mu(n) &= \mathrm{Tr}\left[|n\rangle\langle n|\alpha\rangle\langle\alpha|\right] && (2.39)\\ &= \langle n|\alpha\rangle\langle\alpha|n\rangle && (2.40)\\ &= e^{-\mu}\frac{\mu^n}{n!} && (2.41) \end{aligned}$$

We call a mixture of coherent states with a fixed intensity but uniformly random phase, a **completely phase-randomised** or fully phase-randomised state given by

$$\begin{aligned} \rho^\mu_{\mathrm{PR}} &= \frac{1}{2\pi}\int_0^{2\pi} d\theta|\sqrt{\mu}e^{i\theta}\rangle\langle\sqrt{\mu}e^{i\theta}| && (2.42)\\ &= \sum_n p_\mu(n)|n\rangle\langle n|. && (2.43) \end{aligned}$$

When it might be ambiguous, we refer to the pure coherent states as phase-coherent states in contrast to the fully phase-randomised coherent states. Coherent states are known as "classical states" as they interact with beam splitters and other optical components exactly as we might expect from classical electrodynamics as we will see in the following sections. Note that despite calling them "classical states", these are not the same as the classical states defined at the end of section 2.1.3.

2.4.3 Linear optics

Two lasers or laser pulses ρ_1, ρ_2 that are spatially or temporally separated are described by the composite system $\rho_1 \otimes \rho_2$. We denote the Fock basis and ladder operators for each individual pulse with a subscript $\{|n\rangle_j\}$ and a_j for the j^{th} pulse. We call these different **optical modes**[23]. When describing the action of optical componenets on arbitrary quantum states, we instead describe how the ladder operators would transform. This then describes how the Fock basis transforms as $|n\rangle = \frac{(a^\dagger)^n}{\sqrt{n!}}|0\rangle$.

Beam splitters

Classically, a beam splitter is a device that splits the intensity of an incoming beam in two. Quantum mechanically, a beam splitter is a quantum transformation that takes two input systems and produces two output systems. More precisely, a beam splitter can be represented by a quantum channel $\Phi \in \mathrm{C}(\mathcal{H} \otimes \mathcal{H}, \mathcal{H} \otimes \mathcal{H})$ where each Hilbert space $\mathcal{H}$ describes the space spanned by the Fock basis of a particular optical mode as depicted in Fig. 2.2. As mentioned above, it is sufficient to describe the action of this channel on the Fock basis and thus, a description of the transformation of the ladder operators would give us a description of the channel. The beam splitter transforms the input modes a_1, a_2 to the output modes a_3, a_4 as $a_3 = \sqrt{t}a_1 + e^{i\phi}\sqrt{r}a_2$ and $a_4 = \sqrt{t}a_2 - e^{-i\phi}\sqrt{r}a_1$ where r and t are the reflectivity and transmittivity of the beam splitter respectively with $r + t = 1$, and ϕ is the phase shift caused by the reflection. In this thesis we will only consider the cases when $\phi = \pi$. More concisely, we can write this transformation as

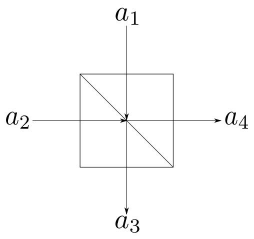


Figure 2.2: Schematic of a generic beam splitter. The input modes are labelled 1 and 2, while the output modes are labelled 3 and 4.

$$\begin{pmatrix} a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} \sqrt{t} & -\sqrt{r} \\ \sqrt{r} & \sqrt{t} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}. \tag{2.44}$$

[23] These can be obtained as solutions to Maxwell's equations as detailed in Chapter 2 of [43].

We can also write the input modes in terms of the output modes by inverting the above matrix. This channel is a unitary transformation since we have assumed that the beam splitter is lossless (when we assumed that $r+t=1$). So, it takes pure states to pure states.

Consider the action of a beam splitter given two coherent states $|\alpha\rangle$ and $|\beta\rangle$ as inputs into mode 1 and mode 2 respectively. By using the relation in Eq. (2.44) and the expansion of coherent states in the Fock basis as given in Eq. (2.38), we can show that the output of the beam splitter from mode 3 and 4 would be $|\sqrt{t}\alpha-\sqrt{r}\beta\rangle$ and $|\sqrt{r}\alpha+\sqrt{t}\beta\rangle$ respectively which corresponds to interference between classical electromagnetic waves.

Phase shifter

A phase shifter is a simple optical device that adds an optical phase ϕ to the input mode. This is represented as

$$a_{\text{out}} = e^{i\phi} a_{\text{in}} \tag{2.45}$$

where the phase ϕ is the phase shift. A schematic of the phase shifter is shown in Fig. 2.3. This acts on coherent states by changing the phase of the state where the input coherent state $|\alpha\rangle$ is transformed to the output state $|\alpha e^{i\phi}\rangle$. From this, we can show that the application of the phase shifter to a fully phase-randomised state leaves it unchanged as

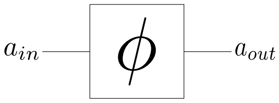


Figure 2.3: Schematic of a phase shifter that shifts the phase of the input mode a_{in} by ϕ.

$$\rho^{\mu}_{\text{PR}} \xrightarrow[\text{shifter}]{\text{phase}} \frac{1}{2\pi}\int_0^{2\pi} d\theta |\sqrt{\mu}e^{i(\theta+\phi)}\rangle\langle\sqrt{\mu}e^{i(\theta+\phi)}| \tag{2.46}$$

$$=\frac{1}{2\pi}\int_0^{2\pi} d\theta |\sqrt{\mu}e^{i\theta}\rangle\langle\sqrt{\mu}e^{i\theta}| = \rho^{\mu}_{\text{PR}} \tag{2.47}$$

Threshold detectors

Ideal[24] threshold detectors are the most basic measurement devices we use. These detectors depicted in Fig. 2.4 detect photons coming in a particular mode, and do not distinguish between the number of photons that arrive at the detector. So, the detector either does not click if there are no photons in that mode, or it clicks if there are one or more photons. Given the annihilation operator a_d of the detector mode, the POVM element for a click event is given by

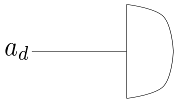


Figure 2.4: Schematic of a threshold detector with annihilation operator a_d.

$$F_{\text{click}} = \sum_n |n\rangle\langle n| \tag{2.48}$$

$$= \sum_n \frac{1}{n!}(a_d^\dagger)^n|0\rangle\langle 0|a_d^n \tag{2.49}$$

and the POVM element for a no-click event is given by

$$F_{\text{no-click}} = |0\rangle\langle 0| \tag{2.50}$$

The probability of getting a no-click event for a coherent state with intensity μ is the same as the probability of the coherent state being in the vacuum state $p_\mu(0)$, and the probability of getting a click is

$$p_{\text{click}} = 1 - p_\mu(0). \tag{2.51}$$

Note that these probabilities do not depend on the phase of the coherent state so it would be the same for any mixture of coherent states having the same intensity but different phases.

[24] By ideal we mean that we do not consider lossy detectors or detectors with dark counts in this thesis for simplicity. However, all the methods and results can be easily extended to these more realistic detectors.

Mach-Zehnder interferometer

The Mach-Zehnder interferometer is a collection of beam splitters and threshold detectors as shown in Fig. 2.5. Both the beam splitters used are 50-50 beam splitters i.e. $r = t = 1/2$.

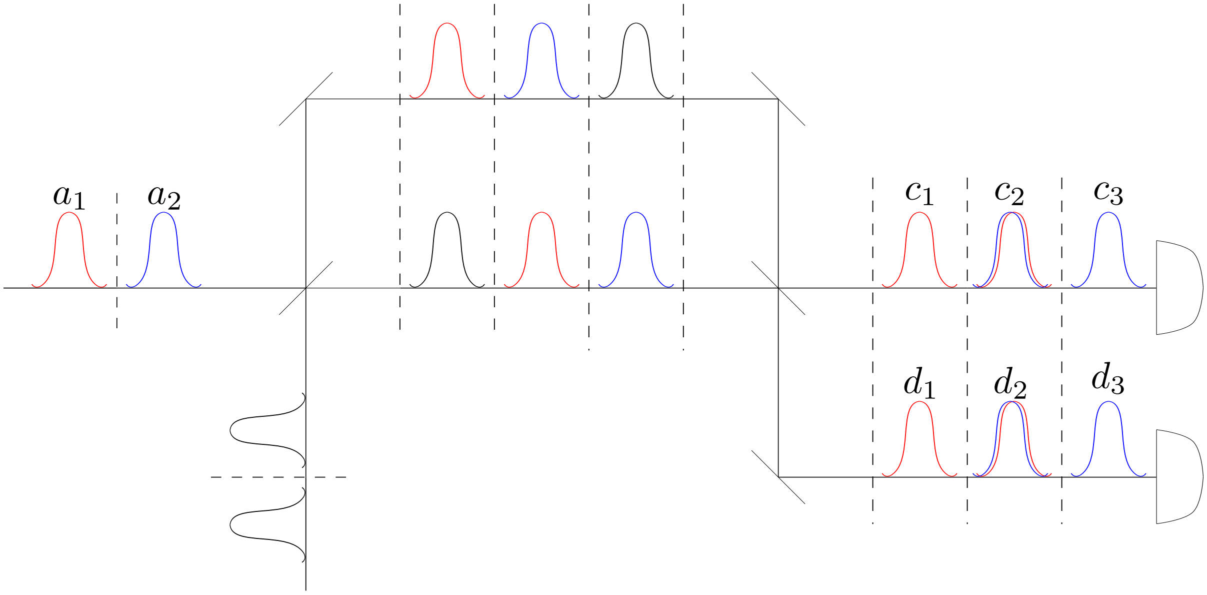


Figure 2.5: Schematic of a Mach-Zehnder interferometer. The black pulses are vacuum pulses.

The input of the first beam splitter has vacuum states coming in at one port, and has two states at different times coming in the other. We denote the annihilation operators corresponding to the non-vacuum input modes to be a_1 and a_2. The two arms of the interferometer are of different lengths so that the middle time-bin of each arm can interfere at the second beam splitter. For the first and third time-bins, one input of the beam splitter would be a vacuum state. Finally, the 6 outputs of the second beam splitter (two from each time-bin) are fed into threshold detectors. We denote the annihilation operators of the modes going into the detectors as d_1, d_2 and d_3 for the arm with destructive interference (which we call the '-' detector), and c_1, c_2 and c_3 for the arm with constructive interference (which we call the '+' detector).

We can show that the output modes d_i and c_i are related to the input modes a_1 and a_2

as follows

$$\begin{aligned}
d_1 &= \frac{1}{2}a_1 \\
d_2 &= -\frac{1}{2}a_1 + \frac{1}{2}a_2 \\
d_3 &= -\frac{1}{2}a_2 \\
c_1 &= \frac{1}{2}a_1 \\
c_2 &= \frac{1}{2}a_1 + \frac{1}{2}a_2 \\
c_3 &= \frac{1}{2}a_2
\end{aligned} \tag{2.52}$$

For a complete derivation of this, see Appendix A of [44]. Note that we consider the effective annihilation operators since we know that some of the beam splitter inputs are always vacuum corresponding to Eq. (A.6) in [44]. For the full equations with all the input modes, refer to Eq. (A.5) in [44].

Now we analyse the case where the input modes are in the state $|\alpha\rangle_1 \otimes |\beta\rangle_2$. As shown in Fig. 2.6 we can use the beam splitter relations above to get the coherent states going into the detectors. The coherent states going into the '-' detector would be $|\frac{\alpha}{2}\rangle \otimes |\frac{-\alpha+\beta}{2}\rangle \otimes |\frac{-\beta}{2}\rangle$ and the states going into the '+' detector would be $|\frac{\alpha}{2}\rangle \otimes |\frac{\alpha+\beta}{2}\rangle \otimes |\frac{\beta}{2}\rangle$. As a special case, consider when $\alpha = \beta$. Here, the states would be $|\frac{\alpha}{2}, 0, \frac{-\alpha}{2}\rangle$ and $|\frac{\alpha}{2}, \alpha, \frac{\alpha}{2}\rangle$. So, in this case the middle time-bin for the '-' detector would never click. Note that the probabilities of clicks of the detectors would not depend on the phase of the input states for the first and third time-bins, but would depend on the relative phase of α and β for the second time-bin.

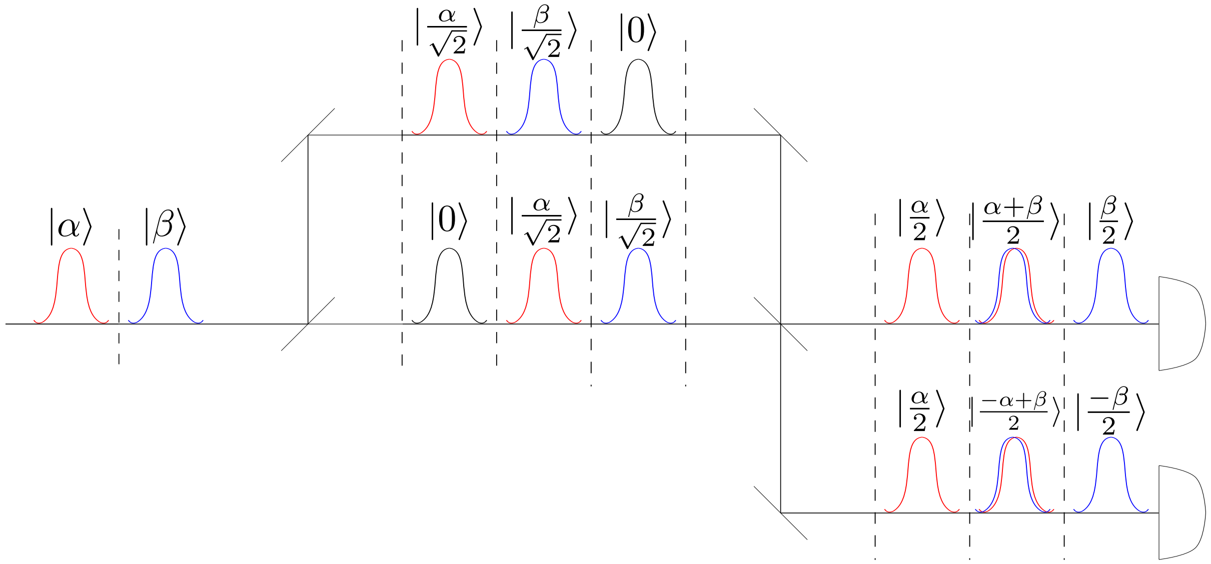

$|\frac{\alpha}{\sqrt{2}}\rangle$
$|\frac{\beta}{\sqrt{2}}\rangle$
$|0\rangle$
$|\alpha\rangle$
$|\beta\rangle$
$|0\rangle$
$|\frac{\alpha}{\sqrt{2}}\rangle$
$|\frac{\beta}{\sqrt{2}}\rangle$
$|\frac{\alpha}{2}\rangle$
$|\frac{\alpha+\beta}{2}\rangle$
$|\frac{\beta}{2}\rangle$
$|\frac{\alpha}{2}\rangle$
$|\frac{-\alpha+\beta}{2}\rangle$
$|\frac{-\beta}{2}\rangle$

Figure 2.6: The result of coherent states passing through a Mach-Zehnder interferometer. We have not depicted the vacuum states that enter the first beam splitter for simplicity.

Chapter 3

Generalised decoy-state method

In this section we describe a general method that was inspired by the decoy-state method [45, 11] which improves the key rates for optical QKD protocols with high loss. It does this by constraining Eve's actions through the use of additional prepared states called decoy states. However, it is more generally applicable for any quantum information processing task that sends some known states $\{\rho_i\}$ through an unknown channel which are then measured, where we wish to know the statistics of another set of states $\{\sigma_j\}$ that is not actually sent. It can be thought of as a limited version of channel tomography. This could be useful since very often the theoretically optimal states for the task might be hard to prepare experimentally.

As an example, the first implementations of QKD [27] used single-photon states. However, realistically we use low intensity coherent states. Any multi-photon states can be shown to be insecure in BB84 type protocols with basis announcements due to the photon-number splitting attack [46, 47] where Eve can use the fact that Bob's threshold detectors do not distinguish between single and multiple photons to gain full information about multi-photon states. So, instead Alice can send **decoy states**, coherent states with different intensities to constrain the number of single-photon signals that Bob would receive. This is a special case of the problem of finding the constraints $p^L_{\tilde{n}}(i,j)$ $\left(p^U_{\tilde{n}}(i,j)\right)$ in Eq. (2.34) as we shall explain in this section.

Another example where techniques like this might find some use is in twin-field type protocols like in [48]. Here, completely phase-randomised states with different intensities are used to estimate the statistics that would arise from sending superpositions of coherent states called cat states. These are then used with analytical methods to find the key rates for the protocol.

In this section we first precisely state the problem as an infinite-dimensional SDP in Section 3.1, we then explain how our general method reduces to the standard decoy-state analysis in Section 3.2. We then detail how to make the SDPs finite-dimensional in Section 3.3, and finally explain how it can be applied to QKD problems in Section 3.4.

3.1 General framework

We will first more precisely describe the general problem. Let $\Phi \in \mathrm{C}(\mathcal{H}, \mathcal{K})$ be the unknown channel[1] that we are trying to constrain. Let $\{\rho_k^\mu\}$[2]$\in \mathrm{D}(\mathcal{H})$ and $\{\Gamma_l\} \in \mathcal{B}(\mathcal{K})$ be the states and POVMs whose statistics we know $\mathrm{Tr}\left[\Gamma_l \Phi(\rho_k^\mu)\right] = \gamma_{kl}^\mu$. Let $\{F_j^N\} \in \mathcal{B}(\mathcal{K})$ be the POVM elements that live in a finite[3] dimensional subspace such that $\Pi_N F_j^N \Pi_N = F_j^N$. Let $\{\sigma_i\} \in \mathrm{D}(\mathcal{H})$ be the set of states whose statistics, when measured with F^N after going through the channel Φ, we wish to bound. These statistics can be written as $\mathrm{Tr}\left[F^N \Phi(\sigma_i)\right]$. We shall refer to $\{\rho_k^\mu\}$ $(\{\sigma_i\})$ and $\{\Gamma_l\}$ $\left(\{F_j^N\}\right)$ as the constraining (objective) states and POVMs respectively.

We assume here that $[\Gamma_l, \Pi_N] = 0$ for all l. Although this is not needed, it makes the bounds tighter and as mentioned in the previous section it is indeed often the case, for example when we use threshold detectors and project in the photon number basis.

The problem of trying to bound the statistics $\mathrm{Tr}\left[F^N \Phi(\sigma_i)\right]$, given that we know $\mathrm{Tr}\left[\Gamma_l \Phi(\rho_k^\mu)\right] = \gamma_{kl}^\mu$, can be phrased as a set[4] of optimisation problems as follows:

$$\begin{aligned} Y^L(i,j) = \min_\Phi & \mathrm{Tr}\left[\Phi(\sigma_i) F_j^N\right] & \qquad Y^U(i,j) = \max_\Phi & \mathrm{Tr}\left[\Phi(\sigma_i) F_j^N\right] \\ \text{s.t.} & \mathrm{Tr}\left[\Phi(\rho_k^\mu)\Gamma_l\right] = \gamma_{kl}^\mu \quad \forall k, l, \mu & \text{s.t.} & \mathrm{Tr}\left[\Phi(\rho_k^\mu)\Gamma_l\right] = \gamma_{kl}^\mu \quad \forall k, l, \mu \\ & \Phi \text{ is CPTP.} & & \Phi \text{ is CPTP.} \end{aligned} \tag{3.1}$$

[1] In QKD this would correspond to Eve's action.

[2] Our notation of using two indices to label elements of this set might seem redundant, but it originates from standard decoy-state analysis where μ denotes the intensity of the decoy state sent, and k denotes Alice's encoding of the state. We retain the notation here to make the link to decoy state analysis more apparent.

[3] If the POVM elements of interest are infinite, we can get F_j^N by taking their finite projections. However, we would then need additional methods to relate the results of the finite POVMs to the infinite POVMs like in Eq. 2.33.

[4] One for each signal state σ_i and POVM element F_j^N. Note that since we have different independent SDPs for each j, this might introduce some looseness to the bounds we get as we do not take into account constraints like $\sum_j F_j^N = \Pi_N$ across all j's. Thus, different choices of POVMs to use in the objective function might not all be equivalent and coarse-grained POVMs might actually do better. However, this looseness also exists in standard-decoy state analysis.

Optimisation problems are hard to solve in general. However, all the constraints are linear. So this can be made into an SDP if we can represent the channel as a positive semidefinite operator. The Choi-Jamiolkowski isomorphism of the channel does just that. Let $J \in \mathcal{L}(\mathcal{H} \otimes \mathcal{K})$ be the Choi-Jamiolkowski isomorphism of the channel Φ. Using Eq. (2.13) to obtain the equivalent trace constraints, we can rewrite the optimisation problem as an SDP as follows:

$$\begin{aligned}\underset{J}{\text{opt.}} \operatorname{Tr}\left[(\sigma_i^T \otimes F_j^N)J\right]&\\ \text{s.t.} \operatorname{Tr}\left[(\rho_k^{\mu T} \otimes \Gamma_l)J\right] &= \gamma_{kl}^{\mu} \quad \forall k, l, \mu\\ J &\geq 0\\ \operatorname{Tr}_{\mathcal{K}}[J] &= \mathbb{I}_{\mathcal{H}}\end{aligned} \tag{3.2}$$

where opt. indicates that we have to optimise the objective function to find both the maximum and the minimum as separate SDPs. We denote the constraints of the form $\operatorname{Tr}\left[(\rho_k^{\mu T} \otimes \Gamma_l)J\right] = \gamma_{kl}^{\mu}$ as expectation value constraints, the constraint $J \geq 0$ as the positivity constraint, and the constraint $\operatorname{Tr}_{\mathcal{K}}[J] = \mathbb{I}_{\mathcal{H}}$ as the partial trace constraint. The main difficulty in this generalised decoy-state SDP is that it is infinite dimensional. So we need to use methods similar to [1] in order to formulate a finite dimensional SDP whose results can bound this infinite SDP.

3.2 Reduction to standard decoy-state analysis

Before moving to the dimension reduction, we show the connection between our general framework and standard decoy-state analysis. In standard decoy, we wish to bound the single-photon statistics while sending only fully-phase randomised coherent states of different intensities. Thus, given Alice's encoding isometries V_i from step 1 of the protocol, and fitting this into the general framework, the objective states are $\sigma_i = V_i|1\rangle\langle 1|V_i^\dagger$ where $|1\rangle$ is a single-photon state, and the constraining states are $\rho_i^\mu = V_i \rho_{\text{PR}}^\mu V_i^\dagger$ where the fully phase-randomised state ρ_{PR}^μ with intensity μ is as defined in Eq. (2.42). The constraining POVMs are given by Bob's measurements Γ_l and the objective POVMs are the finite projections of these in the photon-number basis $F_j^N = \Pi_N \Gamma_j \Pi_N$. Note that by choosing to project in the photon number basis, we ensure that $[\Gamma_l, \Pi_N] = 0$ for all l.

The critical assumption required to perform standard decoy-state analysis is that the base states that the encoding isometries act on are diagonal in the same basis independent of the intensity μ, with the probability distribution of the different basis states changing with different μ. That holds true when the base states are fully phase-randomised as can be seen from Eq. (2.42) since we can write

$$\rho_k^\mu = \sum_n p_\mu(n) V_k |n\rangle\langle n| V_k^\dagger$$

where the photon-number states $|n\rangle$ are independent of μ and all intensity dependence is in the probability distribution $p_\mu(n)$. The constraints in this case become

$$\begin{aligned}
\mathrm{Tr}\left[(\rho_k^{\mu T} \otimes \Gamma_l) J\right] &= \mathrm{Tr}\left[\Phi\left(\rho_k^\mu\right)\Gamma_l\right] && (3.3)\\
&= \mathrm{Tr}\left[\Phi\left(\sum_n p_\mu(n) V_k |n\rangle\langle n| V_k^\dagger\right)\Gamma_l\right] && (3.4)\\
&= \sum_n p_\mu(n) \mathrm{Tr}\left[\Phi\left(V_k |n\rangle\langle n| V_k^\dagger\right)\Gamma_l\right] && (3.5)\\
&= \sum_n p_\mu(n) p(l|k,n) = \gamma_{kl}^\mu && (3.6)
\end{aligned}$$

where $p(l|k,n)$ is the probability of a detection corresponding to the POVM Γ_l given that Alice sent n photons encoded with the isometry V_k.

In standard decoy-state analysis this can now be written as a set of linear programming problems where we need to find bounds on $p(l|k,1)$ and we have linear constraints $\sum_n p_\mu(n)p(l|k,n) = \gamma_{kl}^\mu$. Note that this objective function corresponds to $\mathrm{Tr}\left[(\sigma_a^T \otimes \Gamma_j)J\right]$ and to find the bounds on the finite projection $\mathrm{Tr}\left[(\sigma_a^T \otimes \Gamma_j^N)J\right]$ we would need to further loosen the results of the linear programs using methods from [1] as in Eq. (2.33), whereas in the generalised decoy-state analysis we have the tools to more directly estimate bounds on the projected statistics. Thus, we have shown how our general framework encompasses the standard decoy case, but does not make the same assumptions. Importantly, the states need not be encoded in fully-phase randomised states.

3.3 Finite projections

We will now formulate a finite-dimensional SDP whose optimal value will give us bounds on the optimal value of the SDP in Eq. (3.2). Following [1], this is done by taking projections on the optimisation variable J to get a finite dimensional optimisation variable $J^{MN} = (\Pi_M \otimes \Pi_N)\ J\ (\Pi_M \otimes \Pi_N)$. In order to ensure that we can reliably bound the infinite-dimensional SDP, we then do the following:

1. We appropriately modify the constraints from the infinite-dimensional SDP:

    - Relax expectation value constraints by first projecting on the state space, and then projecting on the measurement space.
    - Showing that the positivity constraint is not affected on projecting.
    - Showing how the partial trace constraint is changed on projecting.

    This results in taking the projections to obtain the feasible set for the finite-dimensional SDP $\mathcal{S}_{MN}$ while ensuring that this contains the projection of the feasible set for the infinite-dimensional SDP $\mathcal{S}_\infty$ i.e. $(\Pi_M \otimes \Pi_N)\ S_\infty\ (\Pi_M \otimes \Pi_N) \subseteq S_{MN}$. This is to ensure that given the optimal point in S_∞ its projection lies in the finite-dimensional feasible set S_{MN} which we would optimise over.

2. We relate the objective functions of the finite and infinite-dimensional SDPs to ensure that the optimal value of the finite-dimensional SDP along with the appropriate correction term would bound the optimal value of the infinite-dimensional SDP.

3.3.1 Expectation value constraints

We shall first loosen the expectation value constraints

$$\mathrm{Tr}\left[(\rho_k^{\mu T} \otimes \Gamma_l)J\right] = \gamma_{kl}^\mu \quad \forall k, l, \mu. \tag{3.7}$$

Throughout this section we shall make repeated use of Eq. (2.13) to rewrite the constraints as

$$\mathrm{Tr}\left[\Gamma_l \Phi(\rho_k^\mu)\right] = \gamma_{kl}^\mu \quad \forall k, l, \mu \tag{3.8}$$

along with the definition of the adjoint map as

$$\mathrm{Tr}\left[\Phi^\dagger(\Gamma_l)\rho_k^\mu\right] = \gamma_{kl}^\mu \quad \forall k, l, \mu. \tag{3.9}$$

Projection on the state space

We know the different states ρ_k^μ. Thus, given a projection Π_M, following the procedure in Appendix A.1 we can find bounds $\left\|\rho_k^\mu - \rho_k^{\mu M}\right\|_1 \leq \epsilon_k^{\mu M}$ where $\rho_k^{\mu M} = \Pi_M\, \rho_k^\mu\, \Pi_M$. We often suppress the size of the projection M in our notation where it is clear from context and simply write the bound on the one-norm as ϵ_k^μ. Now using the statistics we know, we get

$$\begin{aligned} \gamma_{kj}^\mu - \epsilon_k^\mu \leq & \mathrm{Tr}\left[\rho_k^{\mu M}\Phi^\dagger(\Gamma_j)\right] \leq \gamma_{kj}^\mu + \epsilon_k^\mu & (3.10)\\ \implies \gamma_{kj}^\mu - \epsilon_k^\mu \leq & \mathrm{Tr}\left[\left(\rho_k^{\mu M^T} \otimes \Gamma_j\right) J\right] \leq \gamma_{kj}^\mu + \epsilon_k^\mu & (3.11) \end{aligned}$$

as shown in Appendix A.2. If we know that $[\rho_k^\mu, \Pi_M] = 0$, we can tighten the upper bound as shown in Appendix A.2,

$$\begin{aligned} \gamma_{kj}^\mu - \epsilon_k^\mu \leq & \mathrm{Tr}\left[\rho_k^{\mu M}\Phi^\dagger(\Gamma_j)\right] \leq \gamma_{kj}^\mu & (3.12)\\ \implies \gamma_{kj}^\mu - \epsilon_k^\mu \leq & \mathrm{Tr}\left[\left(\rho_k^{\mu M^T} \otimes \Gamma_j\right) J\right] \leq \gamma_{kj}^\mu & (3.13) \end{aligned}$$

We can more concisely write this by defining

$$C\left(\rho_k^{\mu M}, \Pi_M\right) = \begin{cases} 0 & [\rho_k^\mu, \Pi_M] = 0 \\ 1 & [\rho_k^\mu, \Pi_M] \neq 0 \end{cases} \tag{3.14}$$

so that we get

$$\begin{aligned} \gamma_{kj}^\mu - \epsilon_k^\mu \leq & \mathrm{Tr}\left[\rho_k^{\mu M}\Phi^\dagger(\Gamma_j)\right] \leq \gamma_{kj}^\mu + \epsilon_k^\mu\, C\left(\rho_k^{\mu M}, \Pi_M\right) & (3.15)\\ \implies \gamma_{kj}^\mu - \epsilon_k^\mu \leq & \mathrm{Tr}\left[\left(\rho_k^{\mu M^T} \otimes \Gamma_j\right) J\right] \leq \gamma_{kj}^\mu + \epsilon_k^\mu\, C\left(\rho_k^{\mu M}, \Pi_M\right). & (3.16) \end{aligned}$$

Here we note that the introduction of this discontinuous piecewise function C is a possible source of looseness in our methods.

Consider the following example to illustrate the possible source of looseness. Let

$$\rho = \begin{pmatrix} 1-W & -\Delta \\ -\Delta & W \end{pmatrix}, \; \Pi_M = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \text{ and } \Gamma = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$$

with $|\Delta| \leq \sqrt{W(1-W)}$ to ensure that $\rho \geq 0$. We can compute $\epsilon = 2\sqrt{W}$ from Appendix A.1 for ρ. We can compute the traces

$$\mathrm{Tr}\left[\rho\Gamma\right] = 1/2 - \Delta, \text{ and } \mathrm{Tr}\left[\rho^M\Gamma\right] = \frac{1-W}{2}.$$

Now, the trace bounds give us

$$\operatorname{Tr}\left[\rho^M \Gamma\right] \leq \operatorname{Tr}\left[\rho\Gamma\right] + \epsilon\, C(\rho, \Pi_M) \tag{3.17}$$

$$\frac{1-W}{2} \leq 1/2 - \Delta + 2\sqrt{W}\delta_{\Delta,0} \tag{3.18}$$

where the function $C(\rho, \Pi_M)$ simplifies to the Kronecker delta function

$$\delta_{\Delta,0} = \begin{cases} 0 & \Delta = 0 \\ 1 & \Delta \neq 0 \end{cases}$$

since the limit $\Delta = 0$ corresponds to ρ_1 which commutes with the projection Π_M. Rearranging and simplifying, we get

$$\Delta - \frac{W}{2} \leq 2\sqrt{W}\delta_{\Delta,0}. \tag{3.19}$$

Now we can see that whenever Δ is arbitrarily small but non-zero this bound would be loose since $\delta_{\Delta,0} = 1$.

So we would expect that $C(\rho, \Pi_M)$ is a continuous function that depends on the magnitude of the off block-diagonal components that interpolates between the two extremes 0 when these components are 0, and 1 when their magnitude is large. Some work on finding such a function for a similar case was shown in Lemma 4 of [49]. However, it is not easy to directly apply this lemma to our case and further work is needed to tighten the bounds we get.

Projection on the measurement space

We have the loosening of the statistics when we project onto the states ρ_k^μ from Eq. (3.15), but we still need to project onto the POVMs Γ_j to get $\Gamma_j^N = \Pi_N \Gamma_j \Pi_N$. To find this, we need to first find bounds on the weight of the transmitted state $\Phi(\rho_k^\mu)$ outside the Π_N projected subspace W_k^μ defined as $\operatorname{Tr}\left[\Pi_N \Phi(\rho_k^\mu) \Pi_N\right] \geq 1 - W_k^\mu$. Finding these weights W_k^μ, are protocol specific as a general method to find tight bounds on these weights is not known. [1] frames this as an SDP and uses analytical techniques to find tight bounds on the weights. Another method is to use clicks in multiple detectors at once, called cross-clicks as done in [31]. We shall also use this method when we discuss the three-state protocol.

Since we have already projected onto the state space, we would need to combine both ϵ_k^μ with W_k^μ to get the bound on the weight of the transmitted projected state $\Phi(\Pi_M \rho_k^\mu \Pi_M)$

outside the Π_N projected subspace as follows:

$$\begin{align}
\mathrm{Tr}\left[\Pi_N \Phi(\Pi_M \rho_k^\mu \Pi_M)\Pi_N\right] &= \mathrm{Tr}\left[\Pi_M \rho_k^\mu \Pi_M \Phi^\dagger(\Pi_N)\right] \tag{3.20}\\
&\geq \mathrm{Tr}\left[\rho_k^\mu \Phi^\dagger(\Pi_N)\right] - \epsilon_k^\mu \tag{3.21}\\
&= \mathrm{Tr}\left[\Phi(\rho_k^\mu)\Pi_N\right] - \epsilon_k^\mu \tag{3.22}\\
&\geq 1 - W_k^\mu - \epsilon_k^\mu. \tag{3.23}
\end{align}$$

The proof of the first inequality is given in Appendix A.2, and we have used the fact that $\left\|\Phi^\dagger(\Gamma)\right\|_\infty \leq 1$ for any POVM Γ and channel Φ.

Consider now the probability of event l occuring given that the projected state $\Pi_M \rho_k^\mu \Pi_M$ is transmitted through the channel, $\mathrm{Tr}\left[\Gamma_l \Phi(\Pi_M \rho_k^\mu \Pi_M)\right]$. The decrease in the expectation value that occurs as a result of considering the projected POVM Γ_l^N instead can be found as

$$\begin{align}
&\mathrm{Tr}\left[\left(\rho_k^{\mu M^T} \otimes \Gamma_l\right) J\right] - \mathrm{Tr}\left[\left(\rho_k^{\mu M^T} \otimes \Gamma_l^N\right) J\right] \tag{3.24}\\
=&\mathrm{Tr}\left[\Gamma_l \Phi(\Pi_M \rho_k^\mu \Pi_M)\right] - \mathrm{Tr}\left[\Gamma_l^N \Phi(\Pi_M \rho_k^\mu \Pi_M)\right] \tag{3.25}\\
=&\mathrm{Tr}\left[\overline{\Pi}_N \Gamma_l \overline{\Pi}_N \Phi(\Pi_M \rho_k^\mu \Pi_M)\right] \tag{3.26}\\
\leq&\mathrm{Tr}\left[\overline{\Pi}_N \, \mathbb{I} \, \overline{\Pi}_N \Phi(\Pi_M \rho_k^\mu \Pi_M)\right] \tag{3.27}\\
=&\mathrm{Tr}\left[\Phi(\rho_k^{\mu M})\right] - \mathrm{Tr}\left[\Pi_N \Phi(\rho_k^{\mu M})\right] \tag{3.28}\\
\leq&1 - (1 - W_k^\mu - \epsilon_k^\mu) \tag{3.29}\\
=&W_k^\mu + \epsilon_k^\mu \tag{3.30}
\end{align}$$

where the first inequality follows from the fact that $\Gamma_j \leq \mathbb{I}$, and $\overline{\Pi}_N \coloneqq \mathbb{I} - \Pi_N$ and we have used the result in Eq. (3.23) to get Eq. (3.29).

To summarise, in the previous subsection we found bounds on the statistics after projecting on the state space as shown in Eq. (3.16). We then made use of the bound on the weight of the transmitted projected space outside the Π_N projected subspace that we found in Eq. (3.23) to obtain a lower bound on the difference between the statistics on projecting on just the state space, and on projecting on both the state and measurement space in Eq. (3.30). We can combine these results to obtain a lower bound on the statistics after projecting on both the state and the measurement state

$$\begin{aligned}
\mathrm{Tr}\left[\left(\rho_k^{\mu M T} \otimes \Gamma_j^N\right) J\right] &= \mathrm{Tr}\left[\Gamma_j^N \Phi(\Pi_M \rho_k^\mu \Pi_M)\right] && (3.31)\\
&\geq \mathrm{Tr}\left[\Gamma_l \Phi(\Pi_M \rho_k^\mu \Pi_M)\right] - W_k^\mu - \epsilon_k^\mu && (3.32)\\
&\geq \gamma_{kj}^\mu - W_k^\mu - 2\epsilon_k^\mu. && (3.33)
\end{aligned}$$

We can also find the upper bound on $\mathrm{Tr}\left[\left(\rho_k^{\mu M T} \otimes \Gamma_j^N\right) J\right]$ as

$$\begin{aligned}
\mathrm{Tr}\left[\left(\rho_k^{\mu M T} \otimes \Gamma_j^N\right) J\right] &= \mathrm{Tr}\left[\Gamma_j^N \Phi(\Pi_M \rho_k^\mu \Pi_M)\right] && (3.34)\\
&\leq \mathrm{Tr}\left[\Gamma_j \Phi(\Pi_M \rho_k^\mu \Pi_M)\right] && (3.35)\\
&\leq \gamma_{kj}^\mu + \epsilon_k^\mu \, C\left(\rho_k^{\mu M}, \Pi_M\right) && (3.36)
\end{aligned}$$

where the first inequality is proved in Appendix A.2 and uses the fact that $[\Pi_N, \Gamma_j] = 0$. The second inequality follows directly from Eq. (3.15).

So we have found the loosened expectation value constraints after projection

$$\begin{aligned}
&\gamma_{kj}^\mu - W_k^\mu - 2\epsilon_k^\mu \leq \mathrm{Tr}\left[(\rho_k^{\mu M T} \otimes \Gamma_l^N) J\right] \leq \gamma_{kj}^\mu + \epsilon_k^\mu \, C\left(\rho_k^{\mu M}, \Pi_M\right) && (3.37)\\
\implies &\gamma_{kj}^\mu - W_k^\mu - 2\epsilon_k^\mu \leq \mathrm{Tr}\left[(\rho_k^{\mu M T} \otimes \Gamma_l^N) J^{MN}\right] \leq \gamma_{kj}^\mu + \epsilon_k^\mu \, C\left(\rho_k^{\mu M}, \Pi_M\right) && (3.38)
\end{aligned}$$

where the implication that allows us to consider a finite J^{MN} follows from the cyclic property of trace along with the idempotency of projections i.e. $\Pi^2 = \Pi$ for all projections Π.

3.3.2 Positivity constraint

Projecting does not affect the positivity of an operator as shown in Appendix A.1. Thus, the finite-dimensional constraint $J^{MN} \geq 0$ follows from $J \geq 0$.

3.3.3 Partial trace constraint

We shall now describe how the partial trace constraint for the infinite-dimensional SDP

$$\mathrm{Tr}_{\mathcal{K}}\left[J\right] = \mathbb{I}_{\mathcal{H}}$$

is modified on projecting. We first show that $\mathrm{Tr}_K\left[J\right] - \mathrm{Tr}_K\left[\left(\mathbb{I}\otimes\Pi_N\right)J\left(\mathbb{I}\otimes\Pi_N\right)\right] \geq 0$ as follows:

$$\begin{aligned}
\mathrm{Tr}_K\left[J\right] - \mathrm{Tr}_K\left[\left(\mathbb{I}\otimes\Pi_N\right)J\left(\mathbb{I}\otimes\Pi_N\right)\right] =& \mathrm{Tr}_K\left[\left(\mathbb{I}\otimes\Pi_N + \mathbb{I}\otimes\overline{\Pi}_N\right)J\left(\mathbb{I}\otimes\Pi_N + \mathbb{I}\otimes\overline{\Pi}_N\right)\right] \\
&- \mathrm{Tr}_K\left[\left(\mathbb{I}\otimes\Pi_N\right)J\left(\mathbb{I}\otimes\Pi_N\right)\right] && (3.39) \\
=& \mathrm{Tr}_K\left[\left(\mathbb{I}\otimes\overline{\Pi}_N\right)J\left(\mathbb{I}\otimes\overline{\Pi}_N\right)\right] \\
&+ \mathrm{Tr}_K\left[\left(\mathbb{I}\otimes\Pi_N\right)J\left(\mathbb{I}\otimes\overline{\Pi}_N\right)\right] \\
&+ \mathrm{Tr}_K\left[\left(\mathbb{I}\otimes\overline{\Pi}_N\right)J\left(\mathbb{I}\otimes\Pi_N\right)\right] && (3.40) \\
=& \mathrm{Tr}_K\left[\left(\mathbb{I}\otimes\overline{\Pi}_N\right)J\left(\mathbb{I}\otimes\overline{\Pi}_N\right)\right] \\
&+ \mathrm{Tr}_K\left[\left(\mathbb{I}\otimes\overline{\Pi}_N\right)\left(\mathbb{I}\otimes\Pi_N\right)J\right] \\
&+ \mathrm{Tr}_K\left[\left(\mathbb{I}\otimes\Pi_N\right)\left(\mathbb{I}\otimes\overline{\Pi}_N\right)J\right] && (3.41) \\
=& \mathrm{Tr}_K\left[\left(\mathbb{I}\otimes\overline{\Pi}_N\right)J\left(\mathbb{I}\otimes\overline{\Pi}_N\right)\right] \geq 0. && (3.42)
\end{aligned}$$

where we have used the cyclic property of the trace to get the third equality.

We can thus find the modified partial trace constraint

$$\begin{aligned}
\mathrm{Tr}_{\mathcal{K}}\left[J^{MN}\right] &= \mathrm{Tr}_{\mathcal{K}}\left[\left(\Pi_M\otimes\Pi_N\right)J\left(\Pi_M\otimes\Pi_N\right)\right] && (3.43) \\
&= \Pi_M \mathrm{Tr}_{\mathcal{K}}\left[\left(\mathbb{I}\otimes\Pi_N\right)J\left(\mathbb{I}\otimes\Pi_N\right)\right]\Pi_M && (3.44) \\
&\leq \Pi_M \mathrm{Tr}_{\mathcal{K}}\left[J\right]\Pi_M && (3.45) \\
&= \Pi_M && (3.46)
\end{aligned}$$

where we have used the fact that $\mathrm{Tr}_{\mathcal{K}}\left[J\right] - \mathrm{Tr}_{\mathcal{K}}\left[\left(\mathbb{I}\otimes\Pi_N\right)J\left(\mathbb{I}\otimes\Pi_N\right)\right] \geq 0$ from Eq. (3.42) to obtain the first inequality.

3.3.4 Objective function

Having relaxed all the constraints to obtain the feasible set $\mathcal{S}_{MN}$ such that it contains the projection of the infinite-dimensional feasible set $\left(\Pi_M\otimes\Pi_N\right)S_\infty\left(\Pi_M\otimes\Pi_N\right)$, we now turn our attention to the objective function[5]

$$\mathrm{Tr}\left[\left({\sigma_i}^T \otimes F_j^N\right)\ J\right].$$

The measurement F_j^N is already finite in the required space, in fact that is how we chose the projection Π_N. Thus, we need only concern ourselves with the projection on the state space. Similar to Section 3.3.1, we know the states σ_i and so can follow the procedure in Appendix A.1 to find bounds $\left\|\sigma_i - \sigma_i^M\right\|_1 \leq \epsilon_i$ where $\sigma_i = \Pi_M \sigma_i \Pi_M$. We can further use the results proved in Appendix A.2 to get

$$\mathrm{Tr}\left[\left({\sigma_i}^T \otimes \Gamma_j\right) J\right] - \epsilon_i \leq \mathrm{Tr}\left[\left({\sigma_i^M}^T \otimes \Gamma_j\right) J\right] \leq \mathrm{Tr}\left[\left({\sigma_i}^T \otimes \Gamma_j\right) J\right] + \epsilon_i\, C\left(\sigma_i, \Pi_M\right) \tag{3.47}$$

just like in Eq. (3.15). Rearranging the terms to find bounds on the infinite-dimensional objective function we get

$$\mathrm{Tr}\left[\left({\sigma_i^M}^T \otimes \Gamma_j\right) J\right] - \epsilon_i\, C\left(\sigma_i, \Pi_M\right) \leq \mathrm{Tr}\left[\left({\sigma_i}^T \otimes \Gamma_j\right) J\right] \leq \mathrm{Tr}\left[\left({\sigma_i^M}^T \otimes \Gamma_j\right) J\right] + \epsilon_i \tag{3.48}$$

The upper (lower) bound gives us the correction term that we would need to include to get a reliable upper (lower) bound to the maximisation (minimisation) of the infinite-dimensional SDP.

[5] We note that finding the correction term for a general convex function that must be added to the optimal value of the finite-dimensional SDP in order to bound the infinite-dimensional SDP is not simple. This correction term for the lower bound for a large class of functions was found in Theorem 1 of [1]. However, since our objective function is linear, our task is significantly simpler and follows similar to the bounds in Section 3.3.1.

We have our finite-dimensional SDPs

$$
\begin{aligned}
Y_M^L(i,j) = \min_{J^{MN}} \ & \mathrm{Tr}\left[\left(\sigma_i^M \otimes \Gamma_j^N\right)\ J^{MN}\right] \\
& \text{s.t.} \gamma_{kj}^{\mu} - W_k^{\mu} - 2\epsilon_k^{\mu} \leq \mathrm{Tr}\left[(\rho_k^{\mu MT} \otimes \Gamma_l^N) J^{MN}\right] && \forall k, l, \mu \\
& \mathrm{Tr}\left[(\rho_k^{\mu MT} \otimes \Gamma_l^N) J^{MN}\right] \leq \gamma_{kj}^{\mu} + \epsilon_k^{\mu}\, C\left(\rho_k^{\mu}, \Pi_M\right) && \forall k, l, \mu \\
& J^{MN} \geq 0 \\
& \mathrm{Tr}_{\mathcal{K}}\left[J^{MN}\right] \leq \Pi_M \\
Y_M^U(i,j) = \max_{J^{MN}} \ & \mathrm{Tr}\left[\left(\sigma_i^M \otimes \Gamma_j^N\right)\ J^{MN}\right] \\
& \text{s.t.} \gamma_{kj}^{\mu} - W_k^{\mu} - 2\epsilon_k^{\mu} \leq \mathrm{Tr}\left[(\rho_k^{\mu MT} \otimes \Gamma_l^N) J^{MN}\right] && \forall k, l, \mu \\
& \mathrm{Tr}\left[(\rho_k^{\mu MT} \otimes \Gamma_l^N) J^{MN}\right] \leq \gamma_{kj}^{\mu} + \epsilon_k^{\mu}\, C\left(\rho_k^{\mu}, \Pi_M\right) && \forall k, l, \mu \\
& J^{MN} \geq 0 \\
& \mathrm{Tr}_{\mathcal{K}}\left[J^{MN}\right] \leq \Pi_M
\end{aligned}
\tag{3.49}
$$

where we have once again used the cyclic property of trace along with the idempotency of projections to make all the operators finite. We can use the optimal value of this SDP to bound the optimal value of the infinite-dimensional SDP of interest by using the correction term from Eq. (3.48) to get $Y^U(i,j) \leq Y_M^U(i,j) + \epsilon_i$ and $Y^L(i,j) \geq Y_M^L(i,j) - \epsilon_i\, C\left(\sigma_i, \Pi_M\right)$. We can now use these bounds for the quantum information processing task of interest.

3.4 Application to QKD

We will now specifically talk about the generalised decoy-state method when applied to QKD. Borrowing notation and terminology from standard decoy, we will assume that we have a set of decoy states ρ_k^{μ}, where μ is a label that denotes the different decoys which we use to constrain Eve's attacks, and k denotes Alice's encoding as $\rho_k^{\mu} = V_k \rho^{\mu} V_k^{\dagger}$ for some base state $\rho^{\mu} \in \mathrm{D}(\mathcal{W})$. We will denote $\rho_k^{\mu \longleftrightarrow \nu}$ as the signal states from which we extract the key. Note that in the case of standard decoy, ρ^{μ} needed to be the fully-phase randomised state. We make no such assumption.

Let $\rho^{\nu} = \sum p_{\tilde{n}} |\tilde{n}\rangle\langle\tilde{n}|$ be the decomposition of ρ^{ν} in terms of its eigenvectors and eigenvalues[6]. In the case of standard decoy $|\tilde{n}\rangle$ were simply the Fock states. However, they could

[6] This decomposition might not be easy to find as diagonalising arbitrary infinite-dimensional density

in general be different. We label them in decreasing order of the eigenvalues so that $p_{\tilde{n}}$ is the n^{th} largest eigenvalue. Thus, we can write the signal states as $\rho_k^\nu = \sum_{\tilde{n}=0}^{\infty} V_k|\tilde{n}\rangle\langle\tilde{n}|V_k^\dagger$. Note that we do not require that ρ^μ have the same eigenvectors, so it might be more apt to label them as $|\tilde{n}_\nu\rangle$ although we omit the label ν for notational simplicity. Having established the notation that we will use in this section, we now explain the framework in which we can apply the methods we have developed in QKD.

3.4.1 Tagging

Recall from our discussion in Section 2.3.3 that we need the signal states to all have the same block-diagonal structure in order to decompose the key rate into separate SDPs for each block as in Eq. (2.34) which helped us eliminate the shield system. This is only the case when the encoding isometries and eigenvectors are such that $\langle\tilde{n}|V_k^\dagger V_l|\tilde{m}\rangle = \delta_{\tilde{n}\tilde{m}}\delta_{kl}$ for all k, l and $\tilde{m}, \tilde{n}$, i.e. $\{V_k|\tilde{n}\rangle\}_{k,\tilde{n}}$ form a set of orthogonal vectors.

This is quite restrictive and does not hold in general, for e.g. if we use coherent states instead of fully-phase randomised states in decoy-state QKD. So to create this block-diagonal structure, we implement a source map as described in Section 2.3.2. The virtual source we use to replace the real source produces the **tagged** states

$$\tau_k^\nu = \sum_{\tilde{n}} p_{\tilde{n}} V_k|\tilde{n}\rangle\langle\tilde{n}|V_k^\dagger \otimes |\tilde{n}\rangle\langle\tilde{n}| \tag{3.50}$$

where we call the second system the tagged system. The source map we use is simply the partial trace on the tagged system. Thus, the virtual source together with the source map can reproduce the real source as

$$\rho_k^\nu = \mathrm{Tr}_2\left[\tau_k^\nu\right].$$

So if we use τ_k as Alice's signal states in the key rate calculations, we obtain a lower bound on the key rate that has ρ_k as the signal states as Eve could always choose to trace out the second system to obtain the ρ_k from the τ_k. This could potentially give Eve more power, and can cause us to underestimate the key rate of the real protocol. We note that in the ideal case where the ρ_k^ν are all block-diagonal in the same basis, tagging simply adds a redundant register. Thus, for ρ_k^ν that are almost block-diagonal, we would intuitively expect the key rate results to be close to the real key rate.

The τ_k^ν all have the same block diagonal structure due to the tagged system. This lets us formulate the key rate optimisation after taking finite projections described in [1] as in

operators is a hard problem. However, we will later explain how to find vectors that are close to these.

Eq. (2.34)

$$\begin{aligned} R^N_{\tilde{n}} &= \min \, p_{\tilde{n}} f(\rho^{\tilde{n}N}_{AB}) \\ \text{such that } & Y^L_{\tilde{n}}(i,j) \leq \mathrm{Tr}\left[\left(|i\rangle\langle i|_A \otimes \Gamma^N_j\right) \rho^{\tilde{n}N}_{AB}\right] \leq Y^U_{\tilde{n}}(i,j) \\ & \mathrm{Tr}_B\left[\rho^{\tilde{n}N}_{AB}\right] \leq \rho^{\tilde{n}}_A \\ & 1 - W \leq \mathrm{Tr}\left[\rho^{\tilde{n}N}_{AB}\right] \leq 1 \\ & \rho^{\tilde{n}N}_{AB} \in \mathrm{Pos}(\mathcal{H}_{AB}), \end{aligned} \tag{3.51}$$

where $\rho^{\tilde{n}N}_{AB}$ is the state that Alice and Bob would share after taking finite projections if the signal states were $V_i|\tilde{n}\rangle\langle\tilde{n}|V_i^\dagger$. Recall that this means that $\rho^{\tilde{n}}_{AB} = \left(\mathbb{I}_{\mathcal{L}(\mathcal{H}_A)} \otimes \Phi\right)(|\psi\rangle\langle\psi|_{AA'})$ where $|\psi\rangle_{AA'} = \sum_i \sqrt{p(i)}|i\rangle_A \otimes V_i|\tilde{n}\rangle_{A'}$ and $p(i)$ is Alice's probability of choosing state i. Thus, we can compute the reduced state $\rho^{\tilde{n}}_A = \sum_{i,j} \sqrt{p(i)p(j)}\langle\tilde{n}|V_j^\dagger V_i|\tilde{n}\rangle \; |i\rangle\langle j|$.

3.4.2 Reduction in dimensions

All that we need to numerically solve the key rate SDPs is to find bounds on the expectation value constraints in Eq. (3.51) for which we use the generalised decoy-state methods developed in this chapter. Namely, we wish to bound $\mathrm{Tr}\left[\left(|i\rangle\langle i|_A \otimes \Gamma^N_j\right) \rho^{\tilde{n}N}_{AB}\right]$. By using the cylicity of trace and idempotency of the projection Π_N, we can show that

$$\mathrm{Tr}\left[\left(|i\rangle\langle i|_A \otimes \Gamma^N_j\right) \rho^{\tilde{n}N}_{AB}\right] = \mathrm{Tr}\left[\left(|i\rangle\langle i|_A \otimes \Gamma^N_j\right) \rho^{\tilde{n}}_{AB}\right].$$

Using Eq. (2.27), we can write this as

$$\mathrm{Tr}\left[\left(|i\rangle\langle i|_A \otimes \Gamma^N_j\right) \rho^{\tilde{n}}_{AB}\right] = p(i)\mathrm{Tr}\left[\Gamma^N_j \Phi(V_i|\tilde{n}\rangle\langle\tilde{n}|V_i^\dagger)\right]. \tag{3.52}$$

Since, $p(i)$ is fixed, we seek to find bounds on $\mathrm{Tr}\left[\Gamma^N_j \Phi(V_i|\tilde{n}\rangle\langle\tilde{n}|V_i^\dagger)\right]$.

Thus, within our generalised decoy-state framework the objective states $\sigma_i = V_i|\tilde{n}\rangle\langle\tilde{n}|V_i^\dagger$, and the objective POVMs $F^N_j = \Gamma^N_j$. For the lower bound on the trace constraint $\mathrm{Tr}\left[\rho^{\tilde{n}}_{AB}\right]$, we can simply add another constraint $F^N_k = \Pi_N$ to get it within the same framework. We make note here of the fact that it is not necessary to set the objective POVMs to be Bob's projected POVMs. We could take some linear combination of them and use these coarse-grained POVMs in the key rate computation instead. Using coarse-grained POVMs might decrease the key rate, as we use less statistics to constrain Eve's actions. However, it might counteract some of the looseness in using a separate SDP for each POVM element as we do not take into account correlations between the optimal values of the different SDPs for each POVM element. So, we might have two competing factors and the choice that gives the tightest key rates would be protocol dependent.

The constraining POVMs and states are simply given by Bob's full measurements Γ_j and the states that Alice actually sends $\rho^\mu_k = V_k \rho^\mu V_k^\dagger$. This gives us the infinite-dimensional SDPs that we need to solve

$$\begin{aligned} &\underset{J}{\text{opt.}}\text{Tr}\left[\left(\left(V_i|\tilde{n}\rangle\langle\tilde{n}|V_i^\dagger\right)^T \otimes F_j^N\right) J\right] \\ &\text{s.t.}\text{Tr}\left[\left(\left(V_k\rho^\mu V_k^\dagger\right)^T \otimes \Gamma_l\right) J\right] = \gamma^\mu_{kl} \quad \forall k, l, \mu \\ &\quad J \geq 0 \\ &\quad \text{Tr}_{\mathcal{K}}\left[J\right] = \mathbb{I}_{\mathcal{H}} \end{aligned} \tag{3.53}$$

We already have all we need to use our method. However, note that the encoding isometries V_i often expand the space[7]. This is not an issue in theory, but it would often make the numerics significantly slower. So we suggest a relaxation of the problem where the bounds are not always tight, but the reduction in dimensions might make it worth it for some protocols. Of course, where large dimensions are not a concern, or if the isometries do not actually significantly expand the space it is better to solve the full problem without this relaxation.

Our relaxation involves ignoring all constraints where the isometry for the constraint is different from that acting on the objective state i.e. $k \neq i$. The resultant SDP would optimise over a larger set so that the max (min) of the relaxed SDP would be an upper (lower) bound of the original SDP. Further rearranging the isometries and using the cyclic property of the trace, we get

$$\begin{aligned} &\underset{J}{\text{opt.}}\text{Tr}\left[\left(|\tilde{n}\rangle\langle\tilde{n}|^T \otimes F_j^N\right)\left(V_i^T \otimes \mathbb{I}\right) J \left(V_i^{\dagger T} \otimes \mathbb{I}\right)\right] \\ &\text{s.t.}\text{Tr}\left[\left(\rho^{\mu T} \otimes \Gamma_l\right)\left(V_i^T \otimes \mathbb{I}\right) J \left(V_i^{\dagger T} \otimes \mathbb{I}\right)\right] = \gamma^\mu_{il} \quad \forall l, \mu \\ &\quad J \geq 0 \\ &\quad \text{Tr}_{\mathcal{K}}\left[J\right] = \mathbb{I}_{\mathcal{H}} \end{aligned} \tag{3.54}$$

We can absorb the isometries into the Choi isomorphism of the channel. This absorbtion is not a relaxation and was only made possible by our earlier relaxation when we ignored the

[7] The maximum size of the expanded space that we need to consider is the span of all the states ρ^μ_k and $V_i|\tilde{n}\rangle\langle\tilde{n}|V_i^\dagger$.

constraints where $k \neq i$. The resulting operator is also a Choi isomorphism of a different channel as it can be thought of the Choi isomorphism of the composition of the channel Φ with the channel described by the action of the isometry V_i. The proof is straightforward and we show it here for completeness as follows:

1. Positivity:

$$\begin{aligned}
&J \geq 0 && (3.55)\\
&\implies \exists\, A\,\text{s.t.}\, J = AA^\dagger && (3.56)\\
&\implies \left(V_i^T \otimes \mathbb{I}\right)\, J\, \left({V_i^\dagger}^T \otimes \mathbb{I}\right) = \left(V_i^T \otimes \mathbb{I}\right)\, AA^\dagger\, \left({V_i^\dagger}^T \otimes \mathbb{I}\right) = BB^\dagger && (3.57)\\
&\implies \left(V_i^T \otimes \mathbb{I}\right)\, J\, \left({V_i^\dagger}^T \otimes \mathbb{I}\right) \geq 0 && (3.58)
\end{aligned}$$

2. Partial trace:

$$\begin{aligned}
\mathrm{Tr}_{\mathcal{K}}\left[\left(V_i^T \otimes \mathbb{I}\right)\, J\, \left({V_i^\dagger}^T \otimes \mathbb{I}\right)\right] &= V_i^T \mathrm{Tr}_{\mathcal{K}}\left[J\right] {V_i^\dagger}^T && (3.59)\\
&= V_i^T \mathbb{I}_{\mathcal{H}} {V_i^\dagger}^T && (3.60)\\
&= V_i^T {V_i^\dagger}^T && (3.61)\\
&= \left(V_i^\dagger V_i\right)^T && (3.62)\\
&= \mathbb{I}_{\mathcal{W}} && (3.63)
\end{aligned}$$

Thus, $J' := \left({V_i^\dagger}^T \otimes \mathbb{I}\right)\, J\, \left(V_i^T \otimes \mathbb{I}\right)$ is the Choi isomporphism of a channel acting on the smaller space $\mathcal{W}$.

We replace J with the new optimisation varaible J' which reduces the dimensions of our SDPs before applying the finite projections to obtain the appropriate finite dimensional

SDPs with the objective state $\sigma_{\tilde{n}} = |\tilde{n}\rangle\langle\tilde{n}|$ and constraining states ρ^μ as follows:

$$\begin{aligned}
Y^U_{\tilde{n}M}(k,j) = \max_{J'^{MN}} \, & \mathrm{Tr}\left[\left(\sigma^M_{\tilde{n}} \otimes F^N_j\right) \, J'^{MN}\right] \\
\text{s.t. } & \gamma^\mu_{kj} - W^\mu_k - 2\epsilon^\mu \leq \mathrm{Tr}\left[\left(\rho^{\mu M^T} \otimes \Gamma^N_l\right) J'^{MN}\right] && \forall l, \mu \\
& \mathrm{Tr}\left[\left(\rho^{\mu M^T} \otimes \Gamma^N_l\right) J'^{MN}\right] \leq \gamma^\mu_{kj} + \epsilon^\mu \, C\left(\rho^\mu, \Pi_M\right) && \forall l, \mu \\
& J'^{MN} \geq 0 \\
& \mathrm{Tr}_{\mathcal{K}}\left[J'^{MN}\right] \leq \Pi_M \\
Y^L_{\tilde{n}M}(k,j) = \min_{J'^{MN}} \, & \mathrm{Tr}\left[\left(\sigma^M_{\tilde{n}} \otimes F^N_j\right) \, J'^{MN}\right] \\
\text{s.t. } & \gamma^\mu_{kj} - W^\mu_k - 2\epsilon^\mu \leq \mathrm{Tr}\left[\left(\rho^{\mu M^T} \otimes \Gamma^N_l\right) J'^{MN}\right] && \forall l, \mu \\
& \mathrm{Tr}\left[\left(\rho^{\mu M^T} \otimes \Gamma^N_l\right) J'^{MN}\right] \leq \gamma^\mu_{kj} + \epsilon^\mu \, C\left(\rho^\mu, \Pi_M\right) && \forall l, \mu \\
& J'^{MN} \geq 0 \\
& \mathrm{Tr}_{\mathcal{K}}\left[J'^{MN}\right] \leq \Pi_M
\end{aligned} \tag{3.64}$$

where Π_M acts on $\mathcal{W}$ and we have a different SDP for each k and j.

3.5 Approximate diagonalisation

There is one more general tool that we might find useful to link our generalised decoy-state method to QKD. At the start of Section 3.4, we assumed that we had the eigenvectors and eigenvalues of the density operator ρ^ν. The eigenvectors were important when using the generalised decoy-state method where they formed the objective states. The eigenvalues were important when breaking up the key rate function into blocks as in Eq. (3.51). These might not always be easy to analytically find. However, any finite-dimensional matrix can be numerically diagonalised. So, we instead numerically diagonalise a suitable finite projection $\rho \coloneqq \frac{\Pi\rho\Pi}{\mathrm{Tr}[\Pi\rho\Pi]}$ for a finite projection Π. We denote the eigenvalues and eigenvectors of the operator to be $\lambda_{\tilde{n}}(\rho)$ and $|v_{\tilde{n}}\rangle$. For this section, we also denote the eigenvalues of ρ^ν by $\lambda_{\tilde{n}}\left(\rho^\nu\right)$ to make the notation easier to read.

We can bound the one-norm as shown in Appendix A.1 $\left\|\rho - \rho^\nu\right\|_1 \leq \epsilon_{\text{proj}}$. This can be used to quantify the maximum deviation in eigenvectors and eigenvalues of the projected operator as shown in Appendix A.3. The bound on eigenvalues follows from Theorem 2

$$|\lambda_{\tilde{n}}\left(\rho^{\nu}\right) - \lambda_{\tilde{n}}\left(\rho\right)| \leq \epsilon_{\text{proj}}. \tag{3.65}$$

The bound on the one-norm of the difference between the eigenvectors of the two operators ϵ_{vec} is given by

$$\left\| |v_{\tilde{n}}\rangle\langle v_{\tilde{n}}| - |\tilde{n}\rangle\langle\tilde{n}| \right\|_1 \leq 2\epsilon_{\text{proj}}/\delta_{\tilde{n}} \tag{3.66}$$

where $\delta_{\tilde{n}} \coloneqq \min\{\lambda_{\tilde{n}}(\rho) - \lambda_{\tilde{n}}(\rho) - \epsilon_{\text{proj}}, \lambda_{\tilde{n}}(\rho) - \lambda_{\tilde{n}}(\rho) - \epsilon_{\text{proj}}\}$. We have proved this bound in Appendix A.3. We note here that this bound makes no use of the structure of the state ρ^{ν}, and it might be possible to obtain a better bound by using more information about the state for e.g. the magnitude of off-diagonal elements. As an example of a case where this bound would be loose is when ρ^{ν} is already diagonal and we project in its eigenbasis. In this case, the eigenvectors of ρ is a subset of the eigenvectors of ρ^{ν} and we would get $\left\| |v_{\tilde{n}}\rangle\langle v_{\tilde{n}}| - |\tilde{n}\rangle\langle\tilde{n}| \right\|_1 = 0$. Finding tighter bounds by using the structure of ρ^{ν} is an interesting avenue for future research.

We can now use the bounds relating the eigenvectors and eigenvalues of the full state ρ^{ν} and projected state ρ when trying to use the generalised decoy-state method for QKD as follows. What we need for our problem is a bound $Y_{\tilde{n}M}(i,j)$ on the statistics of $|\tilde{n}\rangle\langle\tilde{n}|$ when measured with a POVM element F_j^N as shown in Eq. (3.64). However, in order to make use of the generalised decoy-state method we need to know the objective state exactly. So keeping in mind that $|v_{\tilde{n}}\rangle\langle v_{\tilde{n}}|$ lies in an ϵ_{vec} ball around $|\tilde{n}\rangle\langle\tilde{n}|$, we can instead use the generalised decoy-state method with $|v_{\tilde{n}}\rangle\langle v_{\tilde{n}}|$ as the objective state in the SDPs shown in Eq. (3.64) and get the corresponding optimal values $Y'_{\tilde{n}M}(i,j)$. This optimal value is related to the optimal value we want $Y_{\tilde{n}M}(i,j)$ as

$$|Y'_{\tilde{n}M}(i,j) - Y_{\tilde{n}M}(i,j)| \leq \epsilon_{\text{vec}}. \tag{3.67}$$

Note that when solving these SDPs, we would need to project $|v_{\tilde{n}}\rangle$, which lives in the space spanned by the projection Π we use to approximately diagonalise ρ, into another finite space spanned by Π_M as described in Section 3.3.4. So, it might be a good idea to simply choose $\Pi \geq \Pi_M$ so that we have no additional correction term ϵ_i. Finally, when using these bounds in the key rate SDPs as in Eq. (3.51), we would need to use $\lambda_{\tilde{n}}(\rho) - \epsilon_{\text{proj}}$ in lieu of $p_{\tilde{n}}$ to ensure that we obtain a reliable lower bound on the key rate.

A flowchart depicting the entire process of applying the generalised decoy-state method to QKD along with the appropriate modifications needed when we use the approximate diagonalisation method is shown in Fig. 3.1. To summarise, we can apply the generalised

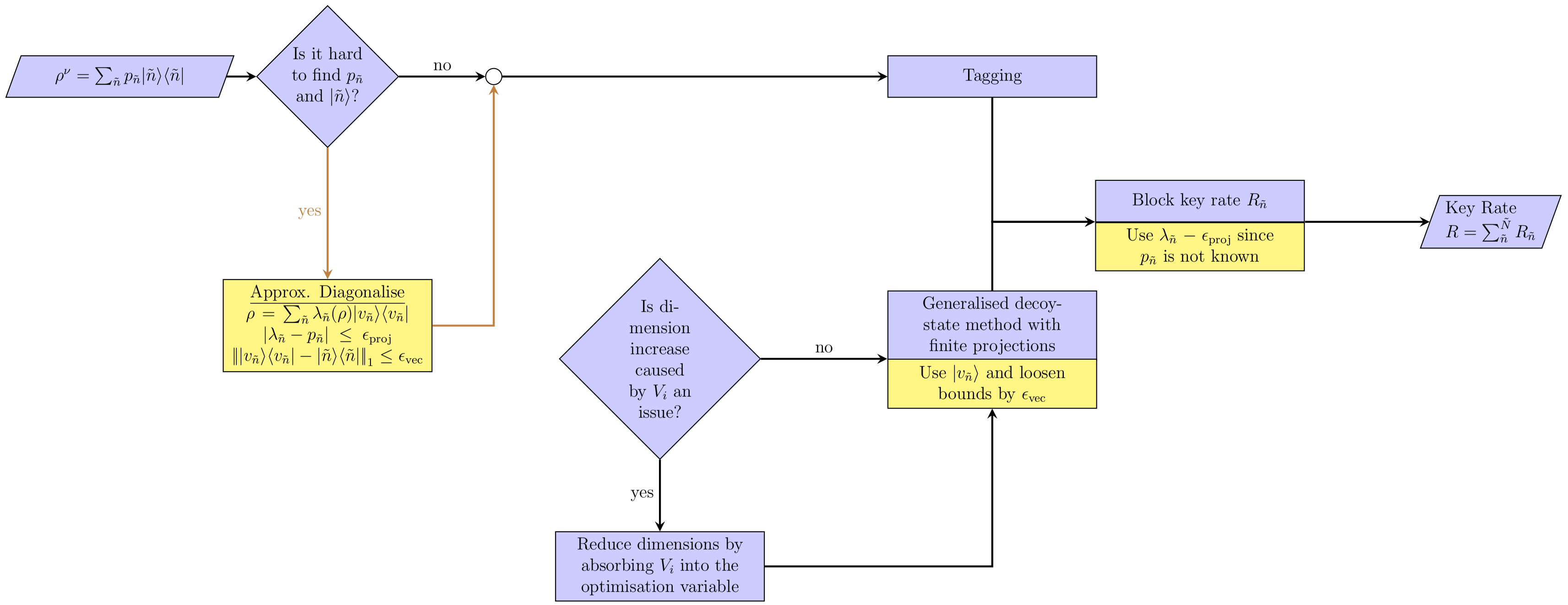

$\rho^\nu = \sum_{\tilde{n}} p_{\tilde{n}} |\tilde{n}\rangle\langle\tilde{n}|$
Is it hard to find $p_{\tilde{n}}$ and $|\tilde{n}\rangle$?
no
yes
Approx. Diagonalise
$\rho = \sum_{\tilde{n}} \lambda_{\tilde{n}}(\rho)|v_{\tilde{n}}\rangle\langle v_{\tilde{n}}|$
$|\lambda_{\tilde{n}} - p_{\tilde{n}}| \leq \epsilon_{\text{proj}}$
$\||v_{\tilde{n}}\rangle\langle v_{\tilde{n}}| - |\tilde{n}\rangle\langle\tilde{n}|\|_1 \leq \epsilon_{\text{vec}}$
Tagging
Is dimension increase caused by V_i an issue?
no
yes
Reduce dimensions by absorbing V_i into the optimisation variable
Generalised decoy-state method with finite projections
Use $|v_{\tilde{n}}\rangle$ and loosen bounds by ϵ_{vec}
Block key rate $R_{\tilde{n}}$
Use $\lambda_{\tilde{n}} - \epsilon_{\text{proj}}$ since $p_{\tilde{n}}$ is not known
Key Rate
$R = \sum_{\tilde{n}}^{\tilde{N}} R_{\tilde{n}}$

Figure 3.1: Flowchart depicting the application of the generalised decoy-state method to QKD. The yellow parts denote the modifications to be made if we need to approximately diagonalise the density operator.

decoy-state method to QKD through the following steps:

1. We must first diagonalise the operator ρ that Alice applies her encoding isometries V_i on. If we cannot exactly diagonalise it, we can use the methods oulined in Section 3.5 to approximately diagonalise it.
2. We implement a source-replacement scheme called tagging to ensure that the signal states are all block-diagonal in the same basis as described in Section 3.4.1. This allows us to decompose the key rate function into multiple smaller SDPs given in Eq. (3.51) that we can individually solve, given the expectation value constraints. We would need to subtract a factor from the eigenvalue $p_{\tilde{n}}$ if we used the approximate diagonalisation procedure.
3. If the generalised decoy SDPs take too long to solve numerically, and if V_i increase the dimension of the space we can loosen the problem and reduce the dimensions as outlined in Section 3.4.2 to get Eq.(3.64).
4. We can obtain the expectation value constraints by using the generalised decoy-state method either fully or with the reduction in dimension. If we used the approximate eigenvectors, we need to loosen the bounds obtained from the generalised decoy SDPs.

3.6 Concluding remarks

In this chapter, we have developed a general tool to bound the statistics of states measured after passing through an unknown channel $\mathrm{Tr}\left[F_j^N \Phi(\sigma_i)\right]$, given the statistics of another set of states measured by another set of measurements passing through the same unknown channel $\mathrm{Tr}\left[\Gamma_l \Phi(\rho_k^\mu)\right]$. In order to apply this to QKD, we developed another general tool to approximately diagonalise (possibly infinite-dimensional) density matrices. These tools are particularly interesting in the context of QKD as they generalise the existing method of decoy-state analysis which assumes that the states are fully phase-randomised. The standard decoy-state analysis results as a special case of our methods where the base states ρ^μ of the constraining states $\rho_k^\mu = V_i \rho^\mu V_i^\dagger$ are all diagonal in the same basis independent of the intensity μ, and the base state σ of the objective states $\sigma_i = V_i \sigma V_i^\dagger$ is an eigenvector of the base signal state ρ^μ. Thus, this generalisation is an important step toward accommodating imperfections in decoy-state analysis.

Chapter 4

Phase imperfections in QKD

The application of the generalised decoy-state method to QKD required the protocol to be iid. This allowed us to apply the quantum de Finetti method and analyse the security of the protocol in the collective attack regime where the channel Φ is the same in all rounds of the protocol as explained in Section 2.3.3. In addition, the implicit assumption we make while using the generalised decoy-state method is that we know exactly what our signal and decoy states are i.e. our source is perfectly characterised. However, using an inaccurate model for the source often leads to security loopholes in QKD that can be exploited by Eve [50].

There has already been some work attempting to find key rates for general correlated sources [9], and uncharacterised sources [10]. However, these methods are not directly applicable to decoy-state QKD which is crucial to achieve distances over 100 km. More recently, there have been advances in using fully phase-randomised sources with intensity correlations for decoy-state QKD [12]. In this chapter, we address the effect of phase correlations in partially characterised laser pulses on the key rate. In conjunction with the generalised decoy-state methods we developed, this can ensure that we no longer need to assume that Alice's laser pulses are fully phase-randomised, or uncorrelated.

For the remainder of this thesis we shall focus on optical protocols where Alice's signal states are encoded in laser pulses. We first motivate the problem by giving some intuition regarding the generation of laser pulses in Section 4.1. We then provide a general model for these laser pulses and provide a reduction of the general source with phase correlations to a simplified source model that is iid in Section 4.2.

4.1 Laser pulses

Laser emission is typically produced by atoms in an optical cavity [51]. Each atom in the cavity is a quantum system which has energy levels. We call higher energy states **excited states**, and the lowest energy state the **ground state**. Whenever the atom goes from an excited state to the ground state, it loses the energy in the form of photons.

There are two mechanisms for this to occur. The first, **spontaneous emission** is what starts the laser. Most of the atoms in the optical cavity are initially excited by pumping energy for e.g. in the form of electric currents for laser diodes. These excited atoms transition to the ground state and randomly emit photons. The second, **stimulated emission** occurs when the existing photons stimulate further emissions to be in the same optical mode. Both types of emissions from the laser are considered to be coherent states [52]. The phases of the coherent states emitted as a result of different spontaneous emissions are entirely uncorrelated from each other. The phases of the coherent states emitted from stimulated emission generated from the same initial seed photons are completely correlated.

So if sufficient power is continuously supplied to the laser, stimulated emission would be the dominant mechanism for laser emission. Thus, the output would be close to a phase-coherent state. However, fully phase-randomised states give better key rates than phase-coherent states for a large class of optical QKD protocols. Intuitively, this might be because the phase information of the coherent states produced by the laser is not used by Bob but can still be used by Eve in most QKD protocols[1]. Thus, such optical protocols ideally try to use fully phase-randomised states in every round.

The simplest way to achieve phase-randomisation is to turn the laser off between pulses. This results in the atoms eventually reverting to the ground state between each pulse. The coherent state generated from each pulse is seeded from spontaneous emission each time since the photons from the previous lasing have vanished from the cavity by the time the laser is turned on again. Thus, the phase of each coherent state is uniformly random resulting in the quantum state of the laser pulse being represented by a fully phase-randomised state.

However, for higher repetition rates [53], the photons from the previous pulse might not all disappear from the laser cavity. These photons might seed the phase of the next laser

[1] This intuition would seem to suggest that the fully phase-randomised states has the highest key rate when compared to any other mixture of coherent states with a fixed intensity. To the best of our knowledge, this has not been rigorously shown to be the case although fully phase-randomised states certainly give better key rates than phase-coherent states and most protocols try to use fully phase-randomised states due to this intuition.

pulse resulting in imperfect phase randomisation and phase correlations across pulses. This makes it hard to accurately model the phase distribution of the source accurately and using the model of completely phase-randomised iid coherent states might result in underestimating Eve's information about the key [54]. Using this as a motivating example, we introduce generic probabilistic phase mixtures of coherent states keeping in mind that they might be correlated and not fully characterised.

4.2 Partially characterised correlated lasers

We consider the space of N laser pulses described by the tensor product of Hilbert spaces for each pulse $\mathcal{H}^{\otimes N}$ since each pulse is temporally separated. Recall that the completely phase-randomised state with intensity μ is given by

$$\rho^{\mu}_{\text{PR}} = \frac{1}{2\pi}\int_0^{2\pi} d\theta |\sqrt{\mu}e^{i\theta}\rangle\langle\sqrt{\mu}e^{i\theta}|$$

as given in Eq. (2.42) where $|\sqrt{\mu}e^{i\theta}\rangle$ is a coherent state with amplitude $\sqrt{\mu}$ and phase θ. Thus, the ideal N pulse train would be written as ${\rho^{\mu}_{\text{PR}}}^{\otimes N}$. We first model the laser pulses before providing a reduction to an iid state.

4.2.1 General laser model

We model a train of N laser pulses as a generic probabilistic mixture of coherent states all having the same intensity μ. This can be given by

$$\rho^{\mu}_{\text{laser}} = \int d\phi_1 \ldots d\phi_N \, p(\phi_1 \ldots \phi_N) |\sqrt{\mu}\mathrm{e}^{\mathrm{i}\phi_1}\rangle\langle\sqrt{\mu}\mathrm{e}^{\mathrm{i}\phi_1}| \otimes \ldots \otimes |\sqrt{\mu}\mathrm{e}^{\mathrm{i}\phi_{\mathrm{N}}}\rangle\langle\sqrt{\mu}\mathrm{e}^{\mathrm{i}\phi_{\mathrm{N}}}| \tag{4.1}$$

where $\phi_i \in [0, 2\pi)$ is the phase of the i^{th} pulse and $p(\phi_1 \ldots \phi_N)$ is the joint probability distribution of the phases of all N pulses. Note that the generic laser state reduces to the ideal state ${\rho^{\mu}_{\text{PR}}}^{\otimes N}$ when the probability distribution is uniform i.e. $p(\phi_1 \ldots \phi_N) = \left(\frac{1}{2\pi}\right)^N$ and the phase-coherent state $|\sqrt{\mu}e^{i\phi}\rangle\langle\sqrt{\mu}e^{i\phi}|^{\otimes N}$ when the probability distribution is a product of Dirac delta functions $p(\phi_1 \ldots \phi_N) = \delta(\phi_1 - \phi)\delta(\phi_2 - \phi) \ldots \delta(\phi_N - \phi)$.

This description is in general not iid. It captures most realistic correlations and other phase imperfections. The only assumptions that we have made are assuming that we have a *probabilistic* mixture, and assuming that all the coherent states in our mixture have the same intensity.

4.2.2 Reduction to characterised uncorrelated lasers

Our strategy for bounding the key rates for protocols that have sources that emit such generic laser pulses is to use a source map as discussed in Section 2.3.2. This source map would help us replace this source with a simpler iid source. We shall first define a precursor to this source map for a single pulse and describe the virtual states we use to replace the laser. We then extend the map to a general pulse train before taking into account the encoding isometries to fully construct the source map.

Single pulse

The state of a single laser pulse is

$$\rho^{\mu}_{\text{laser}} = \int_0^{2\pi} d\phi_1 \, p(\phi_1) |\sqrt{\mu} \mathrm{e}^{\mathrm{i}\phi_1}\rangle\langle\sqrt{\mu}\mathrm{e}^{\mathrm{i}\phi_1}|. \tag{4.2}$$

Consider the state of a phase-coherent laser pulse $\rho^{\mu}_{\text{PC}} = |\sqrt{\mu}\rangle\langle\sqrt{\mu}|$. As described in Section 2.4.3, by using a phase shifter Θ, we can change the phase of the phase-coherent state as $\Theta(\rho^{\mu}_{\text{PC}}) = |\sqrt{\mu}e^{i\theta}\rangle\langle\sqrt{\mu}e^{i\theta}|$. So, by shifting the phase of ρ^{μ}_{PC} by ϕ_1 with probability $p(\phi_1)$, we can get our laser state $\rho^{\mu}_{\text{laser}}$, i.e.

$$\int_0^{2\pi} d\theta_1 p(\theta_1)\Theta_1(\rho^{\mu}_{\text{PC}}) = \int_0^{2\pi} d\theta_1 \, p(\theta_1) |\sqrt{\mu}\mathrm{e}^{\mathrm{i}\theta_1}\rangle\langle\sqrt{\mu}\mathrm{e}^{\mathrm{i}\theta_1}| \tag{4.3}$$

$$= \rho^{\mu}_{\text{laser}} \tag{4.4}$$

Since this transformation was enacted by using optical components, it is certainly a channel - we could, in principle, do this in the lab.

So, we can certainly replace the general laser pulse with a phase-coherent laser pulse followed by a channel. However, we can actually do better than this. First note that the phase shifter does not affect the fully phase-randomised state as stated in Section 2.4.3. We only need to characterise the minimum of the phase distribution

$$q := 2\pi \min_{\phi_1} p(\phi_1) \tag{4.5}$$

in order to construct our virtual laser. Note that since the minimum of $p(\phi_1)$ cannot be greater than $\frac{1}{2\Pi}$, q lies between 0 and 1. Thus, q can be thought of as a probability.

We can think of the laser state as a mixture of a fully phase-randomised state (with probability q), and a generic probability distribution (with probability $1-q$)

$$\rho_{\text{laser}}^{\mu} = q\,\rho_{\text{PR}}^{\mu} + (1-q)\int_0^{2\pi} d\phi_1 \frac{p(\phi_1) - \frac{q}{2\pi}}{1-q} |\sqrt{\mu}e^{i\phi_1}\rangle\langle\sqrt{\mu}e^{i\phi_1}|. \tag{4.6}$$

Intuitively, this is like taking the probability distribution of the phase and subtracting the flat distribution from it till the minimum is 0. Since the completely phase-randomised state has a flat probability distribution, this subtraction exactly corresponds to ρ_{PR}^{μ}.

Now, as described above we can probabilistically use a phase shifter to take us from the phase-coherent state to any generic probabilistic phase mixture of coherent states. In particular, we can probabilistically use the phase shifter to take us from a phase-coherent state to a state with the shifted probability distribution. Thus, the state outputted by the virtual laser for a single pulse is give by

$$\rho_{\text{model}}^{\mu} = q\,\rho_{\text{PR}}^{\mu} + (1-q)|\sqrt{\mu}\rangle\langle\sqrt{\mu}| \tag{4.7}$$

and we call it a **model laser state**.

The channel Ψ_1 such that $\rho_{\text{laser}}^{\mu} = \Psi_1(\rho_{\text{model}}^{\mu})$ can be described by applying a phase-shifter that shifts the phase by ϕ_1 with probability $\frac{p(\phi_1)-\frac{q}{2\pi}}{1-q}$. Note that to construct this model laser state, we do not need to characterise the entire probability distribution $p(\phi_1)$, we just need to characterise its minimum.

Pulse train

The general N laser pulse case is very similar to the single pulse case and carries the same intuition. Starting from the state of N laser pulses given in Eq. (4.1), we can rewrite the probability distribution in terms of conditional probabilities

$$p(\phi_1 \dots \phi_N) = p(\phi_1)p(\phi_2|\phi_1)\dots p(\phi_N|\phi_1\dots\phi_{N-1}). \tag{4.8}$$

The quantity that we need to characterise is the minimum of all the conditional probability distributions

$$q \coloneqq 2\pi \min_i \min_{\phi_i} p(\phi_i|\phi_1\dots\phi_{i-1}). \tag{4.9}$$

The iid state that our virtual laser will prepare is simply the tensor product of the model laser state $\rho_{\text{model}}^{\mu\otimes N}$. The map Ψ_N can be constructed from applying N phase shifters that shift the phase of the i^{th} laser pulse by ϕ_i with probability $\frac{p(\phi_i|\phi_1\ldots\phi_{i-1})-\frac{q}{2\pi}}{1-q}$. Note that we have obtained a correlated state from an iid state by applying a map that is correlated; the action of the i^{th} phase-shifter depends on the action of all the $(i-1)$ phase-shifters before it. Thus, we have $\rho_{\text{laser}}^{\mu} = \Psi_N\left(\rho_{\text{model}}^{\mu\otimes N}\right)$.

Encoding

Before we can use this to construct a source map, we need to know the encoding isometries V_i. Since this is protocol dependent, we do not claim that this would work for all protocols. However, most optical protocols have encoding unitaries $V_i \in \mathrm{U}(\mathcal{H}, \mathcal{H}\otimes\mathcal{H})$ that takes the laser pulse and encodes it into one of two modes for e.g.- polarisation or time-bin encoding[2].

So for any such protocol that does not use any phase correlations across rounds[3], we can construct the source map by applying the same phase shift to both modes. For the i^{th} pair of pulses, the source map Ψ is described by applying phase shifters that shift the phase of both modes by the same ϕ_i with probability $\frac{p(\phi_i|\phi_1\ldots\phi_{i-1})-\frac{q}{2\pi}}{1-q}$. This would ensure that $V_i^{\otimes N}\rho_{\text{laser}}^{\mu}V_i^{\dagger\otimes N} = \Psi\left(V_i^{\otimes N}\rho_{\text{model}}^{\mu\otimes N}V_i^{\dagger\otimes N}\right)$. Thus, we can replace the source that uses the general laser state to encode Alice's states with the virtual source that uses the model laser state to encode Alice's states to get a lower bound on the secret key rate as depicted in Fig. 2.1.

To summarise, in this chapter we have prescribed a general method to reduce signal states with a correlated and partially characterised phase distribution to iid signal states in terms of only the characterised parameter which works for most optical QKD protocols. Even though characterising this single parameter would be significantly easier than characterising the entire phase distribution, we remark that this might still pose significant challenges. For e.g. if we take the most general case with infinite correlation length $\lim N \to \infty$, no finite block of laser pulses are iid. We cannot use the measurement results to construct any meaningful probability distribution as each measurement is effectively on a different density operator. Thus, it might be hard for us to get the statistics to characterise the phase distribution, even partially to find q. So in practice, in order to collect

[2] We will see an example of a protocol that uses time-bin encoding in the next chapter.

[3] Some protocols like the COW [55] and DPS [56] protocols do use phase coherences across protocol rounds. However, such protocols would not attempt to phase-randomise their pulses and so these methods do not apply in such cases.

enough data to interpret the measurement results as probabilities, we might need to make some assumptions about the range of correlations even though the theoretical methods work even in the most general case.

4.3 Concluding remarks

In general, dealing with non-iid protocols is a challenge. However, the methods described in this chapter give a handle on dealing with a large class of protocols that are ideally iid, but have non-iid structure due to phase imperfections. This is achieved by using a non-iid source map with an iid virtual state, resulting in an iid virtual protocol whose key rate lower bounds the key rate for the non-iid real protocol. Combined with the tools developed in Chapter 3, this extends the known proof techniques to correlated lasers used for decoy-state QKD to achieve a high loss-tolerance.

Chapter 5

Three-state protocol

The three-state protocol is a variant of the BB84 protocol where Alice has three signal states instead of four by omitting one of the four signals. A recent optical implementation [2] was able to share secret keys over 421 km of optical fiber. This protocol is of particular interest since it has very few active components making it cheap and easy to implement. The security proof of the protocol used standard decoy-state analysis and so assumed that the states that Alice sent were completely phase-randomised. However, the laser emits pulses at a rate of 2.5 GHz. Thus, there was incomplete phase-randomisation [53]. We use the methods outlined in this thesis to analyse the security of the protocol and estimate the effect of the partial phase-randomisation on the key rate.

5.1 Protocol description

We shall first describe the protocol depicted in Fig. 5.1. This would include Alice's state preparation, Bob's measurements and the parameters that we have used for our security analysis of the protocol.

5.1.1 State preparation

Alice has a laser that emits a mixture of coherent states with the same intensity[1] as described in Eq. (4.1). Alice's encoding isometries that act on the laser pulses can be

[1] For the experiment in [2], this assumption does not entirely hold [53]. However, we only take into account the phase imperfections in this thesis.

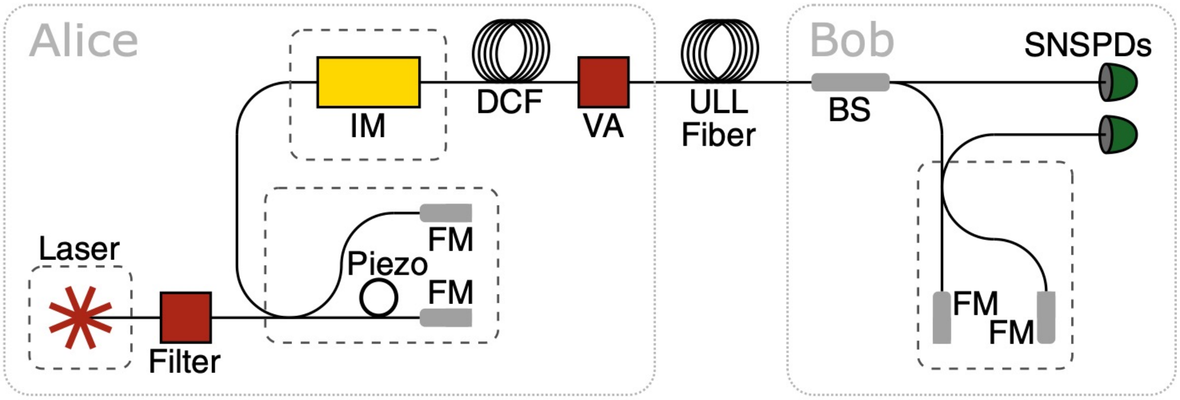


Figure 5.1: Schematic of the experimental setup taken from [2].

described by the following steps:

1. Pass the laser pulse through an unbalanced Michelson interferometer which transforms coherent states from $|\alpha\rangle \to |\alpha/2\rangle \otimes |\alpha/2\rangle \in \mathcal{H} \otimes \mathcal{H}$, where each Hilbert space $\mathcal{H}$ in the tensor product represents a different time-bin.

2. Alice chooses one of 0, 1 and + from her a priori probability distribution $p(i)$ and encodes the state based on the choice:

   0: Alice uses an intensity modulator to suppress the first pulse.

   1: Alice uses an intensity modulator to suppress the second pulse.

   +: Alice uses a variable attenuator to halve the intensity of each pulse so that the total mean photon number of both pulses in all 3 states are the same. We note here that this is an arbitrary choice that we have made. This is not necessary and we could just easily find the key rate without halving the intensity of each pulse.

3. Alice uses the variable attenuator to change the intensity of the pulse and send decoy states as well as signal states.

So given an intensity setting of μ, the three states that she prepares are a phase mixture of the following:

0: $|0\rangle \otimes |\sqrt{\mu}\rangle$,

1: $|\sqrt{\mu}\rangle \otimes |0\rangle$, and

+: $|\sqrt{\mu/2}\rangle \otimes |\sqrt{\mu/2}\rangle$.

Note that the three states are not orthogonal to each other in general, whereas if single photons are used, 0 and 1 states are orthogonal to each other and form a basis, called the Z basis, that spans the qubit space. We use the Z basis for key generation. The + state is an equal superposition of the 0 and 1 states in the qubit space and corresponds to the X basis. This + state is used only for parameter estimation to detect Eve's attacks.

5.1.2 Measurements

Bob's measurement setup has a t:1-t beam splitter that passively "chooses" the basis in which he measures. The Z basis measurement is performed by a threshold detector that records the time of arrival. A click in the first time-bin corresponds to state 1, while a click in the second time-bin corresponds to state 0.

For the X basis measurement, Bob uses a Mach-Zehnder interferometer which measures the phase coherence between the pulses as described in Section 2.4.3. Here only the detector corresponding to destructive interference is used for experimental simplicity. Note that the phase coherences that are measured here are between pulses that have been generated from the same laser pulse due to Alice's Michelson interferometer and so are guaranteed to be in phase if Alice sends a + state and Eve has not manipulated the states in any way.

5.1.3 Simulation parameters

We assume that the quantum channel is a loss-only channel with a loss of 0.16 dB/km based on the experimental implementation in [2]. We also assume Bob's threshold detectors to be ideal with no dark counts for simplicity. We apply a coarse-graining on all click patterns that Bob observes to only consider no-click events, single-click events, and grouping all multi-click events together. This is done to decrease the computation time of the key rate calculation. We denote the coarse-grained POVM $\{\Gamma_j\}$.

Ideally, we would want to optimise over all free parameters. However, this is not computationally feasible so we make some arbitrary choice for each of them. We choose the decoy intensities to be 0 and 0.25 while optimising the signal intensity for different distances. We consider Alice's state preparation choice to be based on an equal a priori distribution, $p(i) = 1/3$ for all $i \in \{0, 1, +\}$. We set Bob's beam splitter to be a 0.9/0.1 beam splitter with the 0.9 transmission side facing the upper detector shown in Fig. 5.1 which corresponds to the Z basis measurement.

Characterisation of the laser

To characterise the quantity of interest as defined in Eq. (4.9), we assume that the phase of each pulse can only be correlated to the phase of the pulse nearest to it. Two consecutive laser pulses are passed through an interferometer with a phase shifter in one arm as described in Section II. A. of [54]. The phase shifter can be modulated and adds a phase shift of ϕ. The intensity of the light arriving at the detector is measured for different values of ϕ.

The state of the two consecutive laser pulses $\rho^{\mu}_{\text{laser}}$ is given by Eq. (4.1) with the joint phase distribution $p(\theta_1, \theta_2) = p(\theta_1)p(\theta_2|\theta_1)$. Assuming that the intensity of both pulses is the same, from Eq. (3) of [54] we get the intensity $I(\phi) \propto 2\mu\,(1 + \langle\cos{(\theta+\phi)}\rangle)$ where $\theta \coloneqq \theta_2 - \theta_1$ is the phase difference between the two pulses and $\langle\cos{(\theta+\phi)}\rangle$ is the average of $\cos{(\theta+\phi)}$ over the phase distribution $p(\theta_1, \theta_2)$. So we have

$$\langle\cos{(\theta+\phi)}\rangle = \int_0^{2\pi} d\theta_1\, p(\theta_1) \int_0^{2\pi} d\theta_2\, p(\theta_2|\theta_1)\cos{(\theta_2 - \theta_1 + \phi)}. \tag{5.1}$$

The experimental quantity that is eventually calculated is termed the **visibility** V. This is given by

$$\begin{aligned} V =& \frac{I_{\text{max}} - I_{\text{min}}}{I_{\text{max}} + I_{\text{min}}} && (5.2)\\ =& \frac{\langle\cos{(\theta+\phi)}\rangle_{\text{max}} - \langle\cos{(\theta+\phi)}\rangle_{\text{min}}}{2 + \langle\cos{(\theta+\phi)}\rangle_{\text{max}} + \langle\cos{(\theta+\phi)}\rangle_{\text{min}}} && (5.3) \end{aligned}$$

where the maximum and minimum is over all ϕ.

Using a 5GHz laser, the authors of [53] measured the visibility of two adjacent laser pulses to be 0.0019 (See Eq. (5) from [53]). They assumed that the laser pulses had no correlation with probability $1 - p_c^*$, and were entirely correlated with probability p_c^*. This corresponds to a conditional probability distribution $p(\theta_2|\theta_1) = \frac{1-p_c^*}{2\pi} + p_c^*\delta(\theta_2 - \theta_1)$. Note that the minimum of this probability distribution is $\frac{1-p_c^*}{2\pi}$, and so from Eq. (4.9) we get $q = 1 - p_c^*$. Calculating the two terms separately, we can see that if $p(\theta_2|\theta_1) = \frac{1}{2\pi}$,

$$\langle\cos{(\theta+\phi)}\rangle = \int_0^{2\pi} d\theta_1\, p(\theta_1) \int_0^{2\pi} d\theta_2\, \frac{1}{2\pi}\cos{(\theta_2 - \theta_1 + \phi)} = 0. \tag{5.4}$$

Thus, the visibility for the completely phase-randomised state is 0, as we might expect.

For the second term, if $p(\theta_2|\theta_1) = \delta(\theta_2 - \theta_1)$,

$$\begin{aligned} \langle \cos{(\theta + \phi)} \rangle &= \int_0^{2\pi} d\theta_1 \, p(\theta_1) \int_0^{2\pi} d\theta_2 \, \delta(\theta_2 - \theta_1) \cos{(\theta_2 - \theta_1 + \phi)} && (5.5) \\ &= \int_0^{2\pi} d\theta_1 \, p(\theta_1) \cos{(\phi)} && (5.6) \\ &= \cos{(\phi)}. && (5.7) \end{aligned}$$

The minimum and maximum of $\cos{(\phi)}$ is 1 and -1 respectively. Thus, the visibility of the phase-coherent state is $V = 1$.

So, we can add these together to get $\langle \cos{(\theta + \phi)} \rangle = p_c^* \cos{(\phi)}$ giving us the visibility $V = p_c^*$. Thus, we can characterise the source as having $1 - q = 0.0019$, or $q = 0.9981$. Recall that q represented the degree of phase-randomisation in the laser pulse as described in Eq. (4.6) and Eq. (4.7). We reiterate here that this parameter value was got under a somewhat restrictive assumption made on the source. For instance, if instead we had assumed that the source phase distribution was a Gaussian centered about the phase of the previous pulse with standard deviation σ as in Eq. (5) from [54], we would get the visibility to be $V = \exp[-\frac{\sigma^2}{2}]$. Using the value for visibility $V = 0.0019$ measured in the experiment, we can get the variance $\sigma^2 \approx 12.53$. We can finally use this to completely characterise the wrapped normal[2] distribution and numerically find the minimum. The value we get for this is $q \approx 0.8782$.

We see that this is far smaller than the value assuming the more simplistic model. It is not entirely clear which model accurately describes our source and so it is not clear precisely how to characterise this parameter. Developing better techniques to characterise this parameter is an avenue for further research.

5.2 Application of generalised decoy-state analysis

We can represent Alice's prepared states by the action of Alice's isometries on the model laser state from Eq. (4.7) $\rho_i^\mu = V_i \rho_{\text{model}}^\mu V_i^\dagger$. Since we have a fully characterised iid source, we can follow the flowchart in Fig. 3.1 to apply the generalised decoy-state method to the three-state protocol.

[2] A wrapped normal distribution $g(\theta)$ is a normal distribution where every point $\theta = \theta + 2\pi$ is taken to be the same. Thus, the probability of these points are all added. It is the circular analogue of the normal distribution which is exactly what we deal with.

Let η be the signal intensity and the eigenvalues and eigenvectors of the model laser state $\rho^{\nu}_{\text{laser}}$ be $p_{\tilde{n}}$ and $|\tilde{n}\rangle$ respectively. Since it is hard to diagonalise this state, we take a finite projection Π_M on the $\leq M$ photon space to get $\rho = \frac{\Pi_M \rho^{\nu}_{\text{model}} \Pi_M}{\text{Tr}\left[\Pi_M \rho^{\nu}_{\text{model}} \Pi_M\right]}$. We choose M to be large enough so that the elements of the $\rho^{\nu}_{\text{model}}$ outside the subspace become smaller than machine precision. We emphasise that we do not ignore the elements smaller than machine precision, but use machine precision to choose the cutoff for the projection. Thus, the difference in one norm between the projected state ρ and the full state $\rho^{\nu}_{\text{model}}$ given by $\epsilon^{\nu}_{\text{proj}}$ is intimately linked with the machine precision. We can numerically diagonalise ρ to get its eigenvalues $\lambda_{\tilde{n}}(\rho)$ and eigenvectors $|v_{\tilde{n}}\rangle$ and can relate these to the eigenvalues and eigenvectors of $\rho^{\nu}_{\text{model}}$ as shown in Section 3.5; $|p_{\tilde{n}} - \lambda_{\tilde{n}}(\rho)| \leq \epsilon^{\nu}_{\text{proj}}$, and $\left\| |v_{\tilde{n}}\rangle\langle v_{\tilde{n}}| - |\tilde{n}\rangle\langle\tilde{n}| \right\|_1 \leq \epsilon_{\text{vec}}$.

We implement tagging as describe in Section 3.4.1 to break the key rates into components that correspond to the different $\tilde{n}$ as in Eq. (3.51). The state we diagonalise is close to (or is exactly) the fully phase-randomised state. Thus, the different $|\tilde{n}\rangle$ are close to the n-photon states. So we expect most of the key rate contribution to come from just the $|\tilde{1}\rangle$ state. For small distances (0-10 km), the partially phase-randomised case with $q = 0.8782$ does show small contributions to the key rate from the $\tilde{0}$ state. Besides this, we observe no other contributions to the key rate from other values of $\tilde{n}$.

So we need to use the generalised decoy-state method to obtain the statistics for the key rate component corresponding to the $|\tilde{1}\rangle$ and $|\tilde{0}\rangle$ states. The isometries add another identical space when described in the Fock basis, effectively doubling the dimension of the signal states as described in Step 1 of Alice's state preparation. Thus, we reduce the dimensions as described in Section 3.4.2 before applying our methods.

In order to solve the SDPs in Eq. (3.64) that we obtain after applying our methods, we can find $\epsilon^{\mu}_{\text{proj}}$ by projecting into the $\leq M$ photon subspace just as we did for diagonalisation. We find Bob's observed statistics γ^{μ}_{jk} for different distances by simulating the loss-only channel as described in Appendix B.1. The weight of the state Bob receives outside the projected subspacec W^{μ}_{k} is estimated from the cross-clicks as shown in Appendix B.2. Since the only time our states commute with the photon-number projections is when $q = 1$, we get $C\left(\rho^{\mu}, \Pi_M\right) = 1 - \delta_{q,1}$. Recall that $C\left(\rho^{\mu}, \Pi\right)$ is a function that checks whether or not ρ^{μ} commutes with Π. Using this data, we get the finite-dimensional SDPs in Eq. (3.64) with $\tilde{n} = \tilde{1}$ and $\tilde{0}$ that we can solve. Since the projection we used to diagonalise the signal state is the same as the projection we used to make the state space finite for the generalised decoy method, the correction term to the objective function in Eq. (3.48) is simply $\epsilon_i = 0$. Thus, after loosening the optimal values for the SDPs by ϵ_{vec} as shown in Eq. (3.67), we get the bound on the statistics we need. This has been summarised in Table 5.1.

<table>
<tr><td colspan="2">Decoy SDP Eq. (3.64)</td></tr>
<tr><td>$\lvert\tilde{n}\rangle$</td><td>$\lvert\tilde{1}\rangle$ and $\lvert\tilde{0}\rangle$</td></tr>
<tr><td>$\epsilon^{\mu}_{\text{proj}}$</td><td>Appendix A.1</td></tr>
<tr><td>γ^{μ}_{jk}</td><td>Appendix B.2</td></tr>
<tr><td>W^{μ}_{k}</td><td>Appendix B.2</td></tr>
<tr><td>$C\left(\rho^{\mu}, \Pi_M\right)$</td><td>$1 - \delta_{q,1}$</td></tr>
<tr><td colspan="2">Objective function correction Eq. (3.48) and (3.67)</td></tr>
<tr><td>ϵ_i</td><td>0</td></tr>
<tr><td>ϵ_{vec}</td><td>Appendix A.3</td></tr>
</table>

Table 5.1: Table listing the quantities that are important to use the generalised decoy-state method along with their value or the Appendix describing how they can be computed.

5.3 Results

Using these bounds on the $\lvert\tilde{1}\rangle$ and $\lvert\tilde{0}\rangle$ statistics, we solve the key rate SDP and plot our results in Fig. 5.2. The source code for our numerics is given in Appendix C. We compared the key rates for the different values of $q = 0.8782$ and $q = 0.9981$ that we get considering different models for the characterisation of the laser as described in Section 5.1.3 using the measurements made in [53]. We also compared these key rates with the fully-phase randomised state $q = 1$. We found that the key rate for the incomplete phase-randomisation with $q = 0.9981$ starts to diverge from the fully phase-randomised key rate from about 150 km. In contrast, the key rate for $q = 0.8782$ starts to diverge from the fully phase-randomised key rate from 0 km illustrating the importance of accurately characterising the laser.

We believe that at least part of the gap between the fully phase-randomised and partially phase-randomised key rates is due to various sources of looseness introduced in our proof techniques and the numerics. One such source is the value of ϵ_{vec} that we get from approximately diagonalising the states. This is a significant factor that only affects the key rate when $q \neq 1$. This can be tightened in two ways. The first way would be to simply increase machine precision that would enable us to take projections onto larger dimensional spaces and accurately diagonalise the corresponding states. This need not increase the size

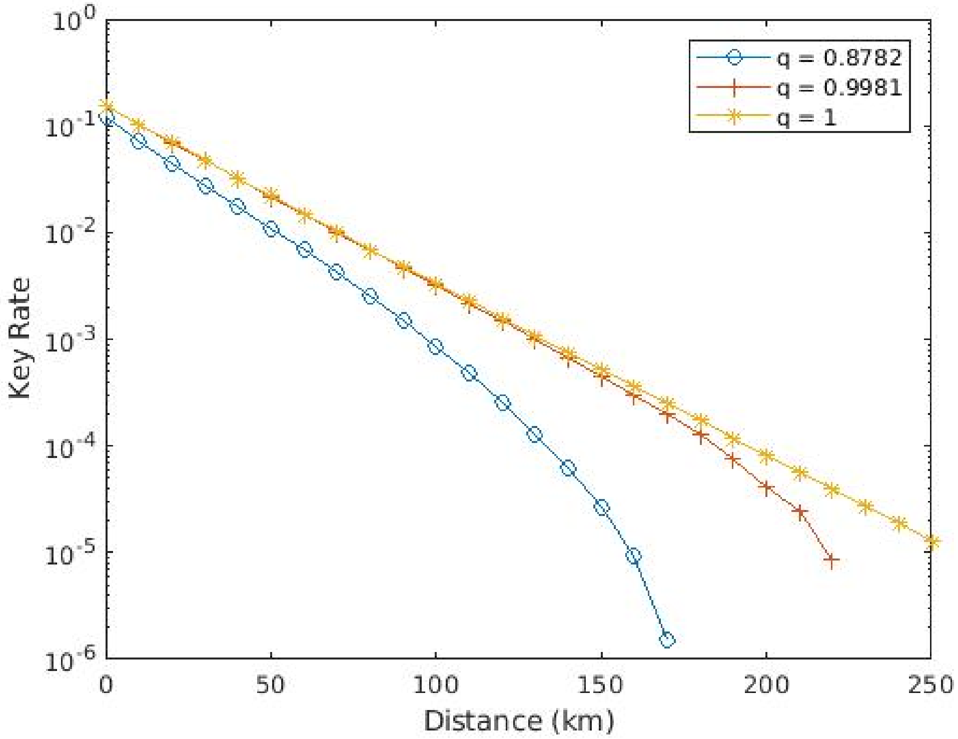


Figure 5.2: Comparison of the key rate of the 3 state protocol with partial ($q = 0.8782$ and $q = 0.9981$) and complete ($q = 1$) phase-randomisation. The partially phase-randomised key rate with $q = 0.9981$ starts to diverge from the fully phase-randomised key rate at 150 km. The key rate for the partially phase-randomised states with $q = 0.8782$ diverges from 0 km.

of the decoy-state SDPs as we could always project once again into a subspace of Π_M, so keeping the dimensions of the SDPs low.

The second way would be to develop analytical methods to obtain better bounds when we approximately diagonalise operators that have small off-diagonal components. Currently we do not use any information about the off-diagonal elements. So if we naively use the same approximate diagonalisation method to diagonalise even the fully phase-randomised state, we would not get $\epsilon_{\text{vec}} = 0$ even though the state is already diagonal. We intuitively expect there to be some way to use the fact that the off-diagonal elements are small in magnitude to get better bounds on the norm of the difference in eigenvectors. However, we do not currently know of any such proof.

Yet another source of looseness comes from $C\left(\rho^{\mu}_{\text{model}}, \Pi_M\right)$ as discussed in Section 3.3.1 where when we go from the fully phase-randomised state to an almost fully phase-randomised state, $C\left(\rho^{\mu}_{\text{model}}, \Pi_M\right)$ goes directly from 0 to 1. We believe that this is too loose and that there should be a way to have a tighter bound with intermediate values of $C\left(\rho^{\mu}_{\text{model}}, \Pi_M\right)$ that depend on the magnitude of the off-diagonal components. Finally, our current bound on $\epsilon^{\mu}_{\text{proj}}$ is once again limited by machine precision which affects the loosening of the constraints in Eq. (3.49), as well as the loosening of the eigenvalue prefactor for the block key rate in Eq. (3.51).

5.4 Concluding remarks

To summarise, in this chapter we have shown how our methods can be used to find the key rate of a practical QKD experiment with phase correlations that arise due to high clock rates. We see that the further the correlated state is from the completely phase-randomised state, less is the loss-tolerance of the protocol. Finally, we discuss the limitations of our methods, and conjecture that the loss in key rate that we get in the presence of phase imperfections might be in part due to our imperfect proof techniques. Thus, an interesting avenue of future research would be to tighten these bounds to make it easier to achieve practical QKD with high clock rates.

Chapter 6

Conclusion

In summary, we formulated a general framework that we call the generalised decoy-state method. This method needs the statistics of some set of states passing through an unknown channel before being measured. It uses these statistics to then bound the statistics of a state outside this set passing through the same unknown channel. We show how to loosen the problem to find reliable bounds if the states and measurements we use act on infinite-dimensional Hilbert spaces.

We also apply this to decoy-state QKD and show how our general framework reduces to standard decoy-state QKD when we use fully phase-randomised states. The application to QKD involved using a process called tagging which allows us to break up the key rate optimisation into smaller blocks at the cost of some looseness to our results, although we believe that this looseness is not significant in most cases as it gives Eve information that is irrelevant to the key bits. We also show a general method to reduce the dimensions of the decoy-state SDPs at the cost of some further looseness that might help speed up computation times in a wide range of protocols.

In order to accommodate infinite-dimensional states that might be hard to diagonalise, we found bounds on the distance between the eigenvalues and eigenvectors of such states with their finite projections, which can be numerically diagonalised. We believe that our methods here are suboptimal and can be significantly improved upon by considering the magnitude of the off-diagonal elements in the state.

We then discuss phase imperfections in the form of correlations and imperfect phase-randomisation. We considered a general probabilistic phase mixture of coherent states of the same intensity with arbitrary correlations and developed a general method to reduce this to a characterised iid model laser state that required some minimal characterisation of the original state. We showed that this simpler model laser state would lower bound the key rate of a wide range of protocols that in reality use the general uncharacterised laser state.

As a concrete example, we applied our methods to the three-state protocol. This is an interesting example since it uses very few active components and is easy to implement. We discuss the problem of partially characterising the laser source, and state clearly the assumptions made for this partial characterisation. We then describe the effect of implementing this protocol at high clock rates on the key rates due to imperfect phase-randomisation. We finally present our results where we found a gap in the fully phase-randomised and partially phase-randomised key rates after 10 km or 150 km depending on our characterisation of the laser. We state that we expect at least some of this gap to be due to our proof techniques and point out specific sources of looseness that we believe can be improved upon with further research.

Our work adds to the existing research on source imperfections and correlations [9, 10]. These works are more general in their consideration of source imperfections and correlations, but were unable to calculate key rates with the decoy-state analysis. In this limit with only signal states, it turns out that the results of the generalised decoy-state SDPs that we obtain are too loose to give us any key rate. So while considering protocols that do not use any decoy states, their methods are better than ours in both generality and key rates. However, for protocols that use decoy states, although our method only considers phase imperfections and correlations, their methods cannot be used at all. There has been recent work in accommodating intensity correlations in decoy-state analysis with fully-phase randomised states [12]. We hope to see our methods combined with theirs to perform decoy-state analysis with a wider range of source imperfections leading us closer to a secure realistic implementation of QKD.

References

[1] Twesh Upadhyaya, Thomas van Himbeeck, Jie Lin, and Norbert Lütkenhaus. Dimension reduction in quantum key distribution for continuous-and discrete-variable protocols. *PRX Quantum*, 2(2):020325, 2021.

[2] Alberto Boaron, Gianluca Boso, Davide Rusca, Cédric Vulliez, Claire Autebert, Misael Caloz, Matthieu Perrenoud, Gaëtan Gras, Félix Bussières, Ming-Jun Li, et al. Secure quantum key distribution over 421 km of optical fiber. *Physical Review Letters*, 121(19):190502, 2018.

[3] Hoi-Kwong Lo, Marcos Curty, and Bing Qi. Measurement-device-independent quantum key distribution. *Physical Review Letters*, 108(13):130503, 2012.

[4] Feihu Xu, Marcos Curty, Bing Qi, and Hoi-Kwong Lo. Measurement-device-independent quantum cryptography. *IEEE Journal of Selected Topics in Quantum Electronics*, 21(3):148–158, 2014.

[5] Zhiyuan Tang, Zhongfa Liao, Feihu Xu, Bing Qi, Li Qian, and Hoi-Kwong Lo. Experimental demonstration of polarization encoding measurement-device-independent quantum key distribution. *Physical Review Letters*, 112(19):190503, 2014.

[6] Hua-Lei Yin, Teng-Yun Chen, Zong-Wen Yu, Hui Liu, Li-Xing You, Yi-Heng Zhou, Si-Jing Chen, Yingqiu Mao, Ming-Qi Huang, Wei-Jun Zhang, et al. Measurement-device-independent quantum key distribution over a 404 km optical fiber. *Physical Review Letters*, 117(19):190501, 2016.

[7] GL Roberts, M Lucamarini, ZL Yuan, JF Dynes, LC Comandar, AW Sharpe, AJ Shields, M Curty, IV Puthoor, and E Andersson. Experimental measurement-device-independent quantum digital signatures. *Nature Communications*, 8(1):1–7, 2017.

[8] Margarida Pereira, Marcos Curty, and Kiyoshi Tamaki. Quantum key distribution with flawed and leaky sources. *npj Quantum Information*, 5(1):1–12, 2019.

[9] Margarida Pereira, Go Kato, Akihiro Mizutani, Marcos Curty, and Kiyoshi Tamaki. Quantum key distribution with correlated sources. *Science Advances*, 6(37):eaaz4487, 2020.

[10] Álvaro Navarrete, Margarida Pereira, Marcos Curty, and Kiyoshi Tamaki. Practical quantum key distribution that is secure against side channels. *Physical Review Applied*, 15(3):034072, 2021.

[11] Hoi-Kwong Lo, Xiongfeng Ma, and Kai Chen. Decoy state quantum key distribution. *Physical Review Letters*, 94(23):230504, 2005.

[12] Víctor Zapatero, Álvaro Navarrete, Kiyoshi Tamaki, and Marcos Curty. Security of quantum key distribution with intensity correlations. *arXiv preprint arXiv:2105.11165*, 2021.

[13] Jason Crann, David W Kribs, and Vern I Paulsen. *Functional Analysis for Quantum Information.* Unpublished course notes for QIC 890 taught at the University of Waterloo, 2019.

[14] Joseph Emerson. *Theory of Quantum Systems: From Mathematical Foundations to Experimental Methods.* Unpublished course notes for QIC 845 taught at the University of Waterloo.

[15] Charles H Bennett and Stephen J Wiesner. Communication via one-and two-particle operators on einstein-podolsky-rosen states. *Physical Review Letters*, 69(20):2881, 1992.

[16] Shmuel Friedland. Infinite dimensional generalizations of choi's theorem. *Special Matrices*, 7(1):67–77, 2019.

[17] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[18] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

[19] Carl W Helstrom. Quantum detection and estimation theory. *Journal of Statistical Physics*, 1(2):231–252, 1969.

[20] Alexander S Holevo. Statistical decision theory for quantum systems. *Journal of multivariate analysis*, 3(4):337–394, 1973.

[21] Christopher A Fuchs and Jeroen Van De Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, 45(4):1216–1227, 1999.

[22] Tomohiro Ogawa and Hiroshi Nagaoka. A new proof of the channel coding theorem via hypothesis testing in quantum information theory. In *Proceedings IEEE International Symposium on Information Theory,*, page 73. IEEE, 2002.

[23] Jamie Sikora. *Analyzing quantum cryptographic protocols using optimization techniques.* PhD thesis, University of Waterloo, 2012.

[24] Ruoheng Liu and Wade Trappe. *Securing wireless communications at the physical layer*, volume 7. Springer, 2010.

[25] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[26] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, January 1983.

[27] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *International Conference on Computers, Systems & Signal Processing*, 1:pp. 175–179, 1984.

[28] Ian George. Numerical finite key analysis. Master's thesis, University of Waterloo, 2020.

[29] Renato Renner. Security of quantum key distribution. *International Journal of Quantum Information*, 6(01):1–127, 2008.

[30] Agnes Ferenczi and Norbert Lütkenhaus. Symmetries in quantum key distribution and the connection between optimal attacks and optimal cloning. *Physical Review A*, 85(5):052310, 2012.

[31] Nicky Kai Hong Li and Norbert Lütkenhaus. Improving key rates of the unbalanced phase-encoded BB84 protocol using the flag-state squashing model. *Physical Review Research*, 2(4):043172, 2020.

[32] Karol Horodecki, Michal Horodecki, Pawel Horodecki, and Jonathan Oppenheim. General paradigm for distilling classical key from quantum states. *IEEE Transactions on Information Theory*, 55(4):1898–1929, 2009.

[33] Oleg Gittsovich, Normand J Beaudry, Varun Narasimhachar, R Romero Alvarez, Tobias Moroder, and Norbert Lütkenhaus. Squashing model for detectors and applications to quantum-key-distribution protocols. *Physical Review A*, 89(1):012325, 2014.

[34] Nicky Kai Hong Li. Application of the flag-state squashing model to numerical quantum key distribution security analysis. Master's thesis, University of Waterloo, 2020.

[35] Renato Renner. *Security of Quantum Key Distribution*. PhD thesis, ETH Zurich, 9 2005. Available at http://arxiv.org/abs/quant-ph/0512258.

[36] Marco Tomamichel, Charles Ci Wen Lim, Nicolas Gisin, and Renato Renner. Tight finite-key analysis for quantum cryptography. *Nature Communications*, 3(1):1–6, 2012.

[37] Matthias Christandl, Robert König, and Renato Renner. Postselection technique for quantum channels with applications to quantum cryptography. *Physical Review Letters*, 102(2):020504, 2009.

[38] Igor Devetak and Andreas Winter. Distillation of secret key and entanglement from quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and engineering sciences*, 461(2053):207–235, 2005.

[39] Adam Winick, Norbert Lütkenhaus, and Patrick J Coles. Reliable numerical key rates for quantum key distribution. *Quantum*, 2:77, 2018.

[40] Patrick J Coles. Unification of different views of decoherence and discord. *Physical Review A*, 85(4):042103, 2012.

[41] S.J. van Enk. Coherent states, beam splitters and photons. https://pages.uoregon.edu/svanenk/solutions/NotesBS.pdf, September 2011.

[42] Christopher Gerry and Peter Knight. *Introductory Quantum Optics*. Cambridge university press, 2005.

[43] Daniel F Walls and Gerard J Milburn. *Quantum optics*. Springer Science & Business Media, 2007.

[44] Varun Narasimhachar. Study of realistic devices for quantum key-distribution. Master's thesis, University of Waterloo, 2011.

[45] Won-Young Hwang. Quantum key distribution with high loss: toward global secure communication. *Physical Review Letters*, 91(5):057901, 2003.

[46] Norbert Lütkenhaus and Mika Jahma. Quantum key distribution with realistic states: photon-number statistics in the photon-number splitting attack. *New Journal of Physics*, 4(1):44, 2002.

[47] Bruno Huttner, Nobuyuki Imoto, Nicolas Gisin, and Tsafrir Mor. Quantum cryptography with coherent states. *Physical Review A*, 51(3):1863, 1995.

[48] Marcos Curty, Koji Azuma, and Hoi-Kwong Lo. Simple security proof of twin-field type quantum key distribution protocol. *npj Quantum Information*, 5(1):1–6, 2019.

[49] Twesh Upadhyaya. Tools for the security analysis of quantum key distribution in infinite dimensions. Master's thesis, University of Waterloo, 2021.

[50] Eleni Diamanti, Hoi-Kwong Lo, Bing Qi, and Zhiliang Yuan. Practical challenges in quantum key distribution. *npj Quantum Information*, 2(1):1–12, 2016.

[51] Arthur L Schawlow and Charles H Townes. Infrared and optical masers. *Physical Review*, 112(6):1940, 1958.

[52] Klaus Mølmer. Optical coherence: A convenient fiction. *Physical Review A*, 55(4):3195, 1997.

[53] Fadri Grünenfelder, Alberto Boaron, Davide Rusca, Anthony Martin, and Hugo Zbinden. Performance and security of 5 ghz repetition rate polarization-based quantum key distribution. *Applied Physics Letters*, 117(14):144003, 2020.

[54] Toshiya Kobayashi, Akihisa Tomita, and Atsushi Okamoto. Evaluation of the phase randomness of a light source in quantum-key-distribution systems with an attenuated laser. *Physical Review A*, 90(3):032320, 2014.

[55] Damien Stucki, Nicolas Brunner, Nicolas Gisin, Valerio Scarani, and Hugo Zbinden. Fast and simple one-way quantum key distribution. *Applied Physics Letters*, 87(19):194108, 2005.

[56] Kyo Inoue, Edo Waks, and Yoshihisa Yamamoto. Differential phase shift quantum key distribution. *Physical Review Letters*, 89(3):037902, 2002.

[57] Gerald Teschl. Mathematical methods in quantum mechanics. *Graduate Studies in Mathematics*, 99:106, 2009.

[58] Daniel Hsu. Notes on matrix perturbation and davis-kahan $\sin\theta$ theorem. https://www.cs.columbia.edu/~djhsu/coms4772-f16/lectures/davis-kahan.pdf, Fall 2016.

[59] John Watrous. *The theory of quantum information*. Cambridge university press, 2018.

[60] Lütkenhaus Group. Qkd software. https://github.com/nlutkenhaus/openQKDsecurity, August 2021.

APPENDICES

Appendix A

Bounds on projected operators

In this appendix we give the derivations for various results on projected operators that we use in Chapter 3 of the main thesis.

A.1 Bounds on one-norm

Let ρ be a density matrix, Π be a projection and $\rho^{\Pi} = \Pi\rho\Pi$.

$$\begin{aligned} \mathrm{F}(\rho, \rho^{\Pi}) &= \mathrm{Tr}\left[\sqrt{\sqrt{\rho}\rho^{\Pi}\sqrt{\rho}}\right] && \text{(A.1)} \\ &= \mathrm{Tr}\left[\sqrt{\sqrt{\rho}\Pi\rho\Pi\sqrt{\rho}}\right] && \text{(A.2)} \\ &= \mathrm{Tr}\left[\sqrt{\sqrt{\rho}\Pi\sqrt{\rho}\sqrt{\rho}\Pi\sqrt{\rho}}\right] && \text{(A.3)} \\ &= \mathrm{Tr}\left[\sqrt{\rho}\Pi\sqrt{\rho}\right] && \text{(A.4)} \\ &= \mathrm{Tr}\left[\Pi\rho\Pi\right] && \text{(A.5)} \\ &= \mathrm{Tr}\left[\rho^{\Pi}\right] && \text{(A.6)} \end{aligned}$$

Using Lemma 5 from [22], we can relate this to the one-norm of the difference: $\left\|\rho - \rho^{\Pi}\right\|_1 \leq 2\sqrt{1 - \mathrm{Tr}\left[\rho^{\Pi}\right]}$.

When $[\Pi, \rho] = 0$, this can tightened as follows. First, we show that $\Pi A \Pi \geq 0$ for any $A \geq 0$ and projection Π. Equivalently, we show that $\langle v|\Pi A \Pi|v\rangle \geq 0 \,\forall v$. Let $|w\rangle = \Pi|v\rangle$.

Thus, we need to show that $\langle w|A|w\rangle \geq 0$. This follows from the positive semidefiniteness of A, thus showing that

$$\Pi A \Pi \geq 0. \tag{A.7}$$

We use this fact to show that $(A - \Pi A \Pi) \geq 0$ when $[A, \Pi] = 0$. Let $\overline{\Pi} = \mathbb{I} - \Pi$ be the complementary projection, then

$$\begin{aligned}
(A - \Pi A \Pi) &= (\Pi + \overline{\Pi}) A (\Pi + \overline{\Pi}) - \Pi A \Pi && \text{(A.8)}\\
&= \overline{\Pi} A \overline{\Pi} + \Pi A \overline{\Pi} + \overline{\Pi} A \Pi && \text{(A.9)}\\
&= \overline{\Pi} A \overline{\Pi} + A \Pi \overline{\Pi} + \overline{\Pi} \Pi A && \text{(A.10)}\\
&= \overline{\Pi} A \overline{\Pi} && \text{(A.11)}\\
&\geq 0. && \text{(A.12)}
\end{aligned}$$

As a special case, we see that $\rho - \rho^{\Pi} \geq 0$. Finally, we get that

$$\begin{aligned}
\left\| \rho - \rho^{\Pi} \right\|_1 &= \mathrm{Tr}\left[\rho - \rho^{\Pi}\right] && \text{(A.13)}\\
&= 1 - \mathrm{Tr}\left[\rho^{\Pi}\right]. && \text{(A.14)}
\end{aligned}$$

A.2 Bounds on expectation value

Let A and B be positive semidefinite operators, Π be a projection, $\|A - \Pi A \Pi\|_1 \leq \epsilon$, and $\|B\|_\infty \leq 1$. Then by matrix Hölder's inequality we have that

$$|\mathrm{Tr}\left[(A - \Pi A \Pi) B\right]| \leq \|A - \Pi A \Pi\|_1 \|B\|_\infty \leq \epsilon \tag{A.15}$$

Rearranging the terms in the trace we get

$$|\mathrm{Tr}\left[(A - \Pi A \Pi) B\right]| = |\mathrm{Tr}\left[AB\right] - \mathrm{Tr}\left[\Pi A \Pi B\right]| \leq \epsilon. \tag{A.16}$$

Using the fact that $|c| = |-c| \leq k$ implies that $-k \leq -c \leq k$,

$$\begin{aligned}
&-\epsilon \leq \mathrm{Tr}\left[\Pi A \Pi B\right] - \mathrm{Tr}\left[AB\right] \leq \epsilon && \text{(A.17)}\\
\implies &\mathrm{Tr}\left[AB\right] - \epsilon \leq \mathrm{Tr}\left[\Pi A \Pi B\right] \leq \mathrm{Tr}\left[AB\right] + \epsilon && \text{(A.18)}
\end{aligned}$$

If we know that $[A, \Pi] = 0$, we can tighten the upper bounds as shown on pages 56-57 of [49]. Let $C = A - \Pi A \Pi$. We know that $C \geq 0$ from Eq. (A.12) so

$$\operatorname{Tr}\left[(A - \Pi A \Pi)B\right] = \operatorname{Tr}\left[CB\right] \geq 0 \tag{A.19}$$

$$\implies \operatorname{Tr}\left[\Pi A \Pi B\right] \leq \operatorname{Tr}\left[AB\right] \tag{A.20}$$

when $[A, \Pi] = 0$.

A.3 Closeness of eigenvectors

As in this thesis, one might run into a situation where one might want to find the eigenvectors of a density matrix ρ and the eigenvectors of another density matrix $\sigma = \rho + H$ is known where $\|H\|_1 \leq \epsilon$. In this appendix we explain how and when one can approximate the eigenvectors of ρ with the eigenvectors of σ.

Let $\lambda_i(S)$ be the i^{th} largest eigenvalue of a compact, self-adjoint operator S. From Theorem 4.10 of [57], we can write the eigenvalues as

$$\lambda_n(S) = \min_{|\psi_1\rangle \ldots |\psi_{n-1}\rangle} \max_{|\psi\rangle \in P^\perp(|\psi_1\rangle \ldots |\psi_{n-1}\rangle)} \langle\psi|S|\psi\rangle$$

where $P^\perp(|\psi_1\rangle \ldots |\psi_{n-1}\rangle) \coloneqq \{|\psi\rangle | \, \|\psi\| = 1, \, \psi \in \operatorname{span}\{|\psi_1\rangle \ldots |\psi_{n-1}\rangle\}^\perp\}$ is the space perpendicular to the vectors $|\psi_1\rangle \ldots |\psi_{n-1}\rangle$. From this we can bound the change in eigenvalues due to the perturbation.

Theorem 2. Let $\mathcal{H}$ be a Hilbert space. Given $\rho \in \mathrm{D}(\mathcal{H})$, $\sigma \in \mathrm{D}(\mathcal{H}$ and $H = \sigma - \rho$ with $\|H\|_1 \leq \epsilon$ as defined above,

$$|\lambda_i(\rho) - \lambda_i(\sigma)| \leq \epsilon$$

for all eigenvalues indexed by i.

Proof. The proof follows similarly to the proof of Weyl's inequality which is for finite dimensions.

$$\lambda_i(\sigma) = \min_{|\psi_1\rangle\ldots|\psi_{i-1}\rangle} \max_{|\psi\rangle\in P^\perp(|\psi_1\rangle\ldots|\psi_{i-1}\rangle)} \langle\psi|\sigma|\psi\rangle \tag{A.21}$$

$$= \min_{|\psi_1\rangle\ldots|\psi_{i-1}\rangle} \max_{|\psi\rangle\in P^\perp(|\psi_1\rangle\ldots|\psi_{i-1}\rangle)} \left(\langle\psi|\rho|\psi\rangle + \langle\psi|H|\psi\rangle\right) \tag{A.22}$$

$$\leq \min_{|\psi_1\rangle\ldots|\psi_{i-1}\rangle} \left(\max_{|\psi\rangle\in P^\perp(|\psi_1\rangle\ldots|\psi_{i-1}\rangle)} \langle\psi|\rho|\psi\rangle + \|H\|_\infty \right) \tag{A.23}$$

$$= \lambda_i(\rho) + \|H\|_\infty \tag{A.24}$$

$$\leq \lambda_i(\rho) + \|H\|_1 \tag{A.25}$$

$$= \lambda_i(\rho) + \epsilon \tag{A.26}$$

Starting with ρ instead of σ in the first line and following the same steps while replacing H with $-H$ gives us $\lambda_i(\rho) \leq \lambda_i(\rho) + \epsilon$. Combining both together, we get $|\lambda_i(\rho) - \lambda_i(\sigma)| \leq \epsilon$ as stated. □

Before talking about the individual eigenvectors, we shall introduce the Davis-Kahan $\sin(\theta)$ theorem [58] about the closeness of the eigenspaces after perturbation. Although the proof in [58] is for finite dimensions, the proof for infinite dimensional density operators is exactly the same.

Theorem 3 (Davis-Kahan $\sin(\theta)$ theorem)**.** Let $\rho = U\rho_D U^\dagger + U'\rho'_D U'^\dagger$ and $\sigma = V\sigma_D V^\dagger + V'\sigma'_D V'^\dagger$ be density operators as defined above where the block matrices $[U \quad U']$ and $[V \quad V']$ are orthogonal. Let their difference be $H = \sigma - \rho$. If the eigenvalues of ρ_D are contained in an interval (a, b), and the eigenvalues of σ'_D are excluded from the interval $(a - \delta, b + \delta)$ for some $\delta > 0$, then

$$\left\|V'^\dagger U\right\| \leq \frac{\left\|V'^\dagger H U\right\|}{\delta} \tag{A.27}$$

for any unitarily invariant norm $\|\cdot\|$.

Intuitively, the δ represents how separated the eigenspaces of σ are relative to the perturbation ϵ. If this δ is too small, the corresponding eigenspaces could be quite different. As an extreme example, consider

$$\rho = \begin{pmatrix} 1/2 - \epsilon/2 & 0 \\ 0 & 1/2 + \epsilon/2 \end{pmatrix}, \; H = \begin{pmatrix} \epsilon/2 & 0 \\ 0 & -\epsilon/2 \end{pmatrix}, \; \sigma = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}. \tag{A.28}$$

$\|H\|_1 \leq \epsilon$ as required. Since $\delta = \epsilon$, the perturbation might be arbitrarily small compared to the eigenvalues, but is not small compared to the separation of the eigenvalues. Now, the eigenvectors of ρ are $\binom{1}{0}$ and $\binom{0}{1}$. However, since σ has degenerate eigenvalues, any vectors lying in the degenerate space are eigenvectors. So if we chose the "wrong" eignvectors of σ to start with, we might get very far from the eigenvectors of ρ even though the perturbation is small, for e.g. $\binom{1/\sqrt{2}}{1/\sqrt{2}}$ has a fidelity of $1/\sqrt{2}$ with both the eigenvectors. We encourage the interested reader to refer to [58] for instructive examples and further intuition about this theorem.

We shall now use these theorems repeatedly to talk about individual eigenvectors. Let $U_i(V_i)$ be the i^{th} eigenvector of $\rho(\sigma)$, and $U_i'(V_i')$ be the matrix of eigenvectors of $\rho(\sigma)$ spanning the space perpendicular to $U_i(V_i)$. Further, define $\rho_i(\sigma_i)$ be the diagonal matrix containing all the eigenvalues of $\rho(\sigma)$ except the i^{th} eigenvector ordered so that we can write

$$\rho = U_i \lambda_i(\rho) U_i^\dagger + U_i' \rho_i U_i'^\dagger.$$

From Theorem 2 we know that $\lambda_i(\rho)$ lies in the interval (a_i, b_i) with $a_i = \lambda_i(\sigma) - \epsilon$ and $b_i = \lambda_i(\sigma) + \epsilon$. For $i = 1$, define $\delta_1 = \lambda_2(\sigma) - \lambda_1(\sigma) - \epsilon$. For $i \geq 2$, define $\delta_i = \min\{\lambda_i(\sigma) - \lambda_{i-1}(\sigma) - \epsilon, \lambda_{i+1}(\sigma) - \lambda_i(\sigma) - \epsilon\}$. So we know that all the eigenvalues of σ_i lie outside $(a_i - \delta_i, b_i + \delta_i)$.

We can now use Theorem 3 to get

$$\begin{aligned}
\left\| V_i'^\dagger U_i \right\|_\infty &\leq \frac{\left\| V_i'^\dagger H U_i \right\|_\infty}{\delta_i} && \text{(A.29)} \\
&\leq \frac{\left\| V_i'^\dagger \right\|_\infty \|H\|_\infty \|U_i\|_\infty}{\delta_i} && \text{(A.30)} \\
&= \frac{\|H\|_\infty}{\delta_i} && \text{(A.31)} \\
&\leq \frac{\|H\|_1}{\delta_i} && \text{(A.32)} \\
&\leq \frac{\epsilon}{\delta_i} && \text{(A.33)}
\end{aligned}$$

where the second inequality follows from the fact that the ∞-norm is submultiplicative $\|AB\|_\infty \leq \|A\|_\infty \|B\|_\infty$, and the succeeding equality follows since all U_i' and V_i' have maximum singular value 1.

Consider the block matrices

$$R_i = \begin{pmatrix} U_i & U_i' \end{pmatrix}, \text{and } S_i = \begin{pmatrix} V_i & V_i' \end{pmatrix}.$$

These are unitaries as they are the collection of all the orthonormal eigenvectors of ρ and σ respectively. Note that $U_i^\dagger U_i (V_i^\dagger V_i)$ are 1-dimensional as $U_i(V_i)$ are the i^{th} eigenvectors of $\rho(\sigma)$. Thus,

$$W_i := R_i^\dagger S_i = \begin{pmatrix} U_i^\dagger V_i & U_i^\dagger V_i' \\ U_i'^\dagger V_i & U_i'^\dagger V_i' \end{pmatrix}$$

must also be unitary. So $W_i W_i^\dagger = \mathbb{I}$. Looking at the first block which is 1-dimensional,

$$U_i^\dagger V_i V_i^\dagger U_i + U_i^\dagger V_i' V_i'^\dagger U_i = 1. \tag{A.34}$$

From Eq. (A.33) we know that $\left\| V_i'^\dagger U_i \right\|_\infty = \left\| U_i^\dagger V_i' \right\|_\infty \leq \frac{\epsilon}{\delta_i}$. Thus,

$$\left\| U_i^\dagger V_i' V_i'^\dagger U_i \right\|_\infty \leq \left\| U_i^\dagger V_i' \right\|_\infty \left\| V_i'^\dagger U_i \right\|_\infty \leq \frac{\epsilon^2}{\delta_i^2}$$

Finally, we can use this in Eq. (A.34) to get

$$U_i^\dagger V_i' V_i'^\dagger U_i = 1 - U_i^\dagger V_i V_i^\dagger U_i \tag{A.35}$$

$$\left\| U_i^\dagger V_i' V_i'^\dagger U_i \right\|_\infty = \left\| 1 - U_i^\dagger V_i V_i^\dagger U_i \right\|_\infty \tag{A.36}$$

$$\left| 1 - U_i^\dagger V_i V_i^\dagger U_i \right| \leq \frac{\epsilon^2}{\delta_i^2} \tag{A.37}$$

Observing that the fidelity between these eigenvectors is $\left| U_i^\dagger V_i \right|$, Eq. (A.37) directly gives us a bound on the fidelity,

$$F(U_i, V_i)^2 \leq 1 - \frac{\epsilon^2}{\delta_i^2}. \tag{A.38}$$

Finally, we can use Fuchs-van de Graaf inequality [59] to get

$$\left\| U_i U_i^\dagger - V_i V_i^\dagger \right\|_1 \leq 2 \frac{\epsilon}{\delta_i}. \tag{A.39}$$

So we have found a bound on how close the eigenvectors of ρ are to σ.

Appendix B

Three-state protocol statistics

In this appendix we first simulate a loss-only channel and explain how to find all the statistics from Bob's measurements for the three-state protocol described in Chapter 5. We then detail a method to bound the weight outside the $\leq N$ photon subspace. Bob's measurement apparatus is depicted in Fig. B.1 and we shall refer to it throughout this appendix.

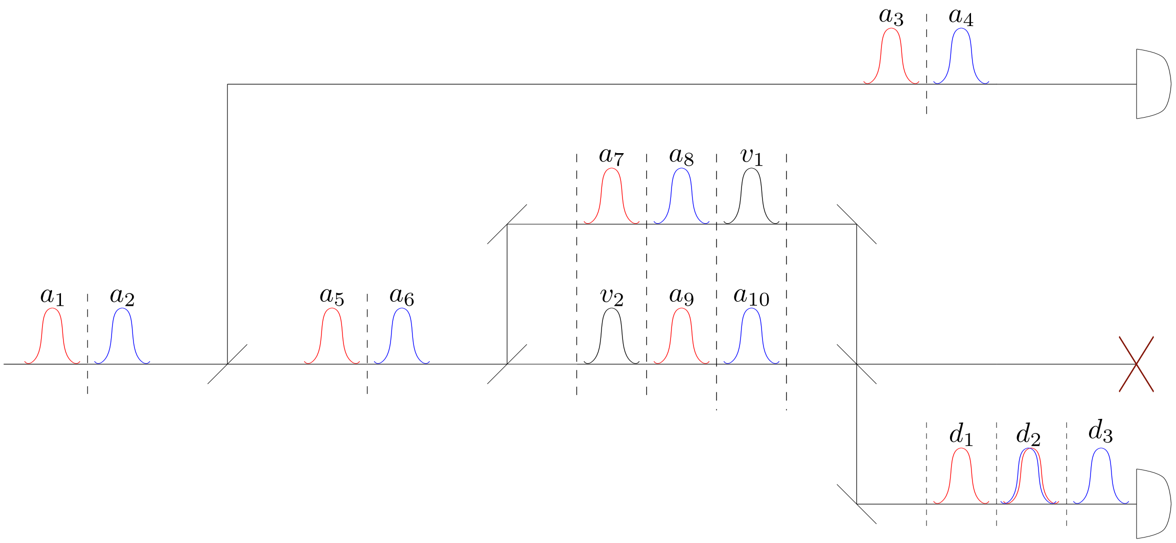
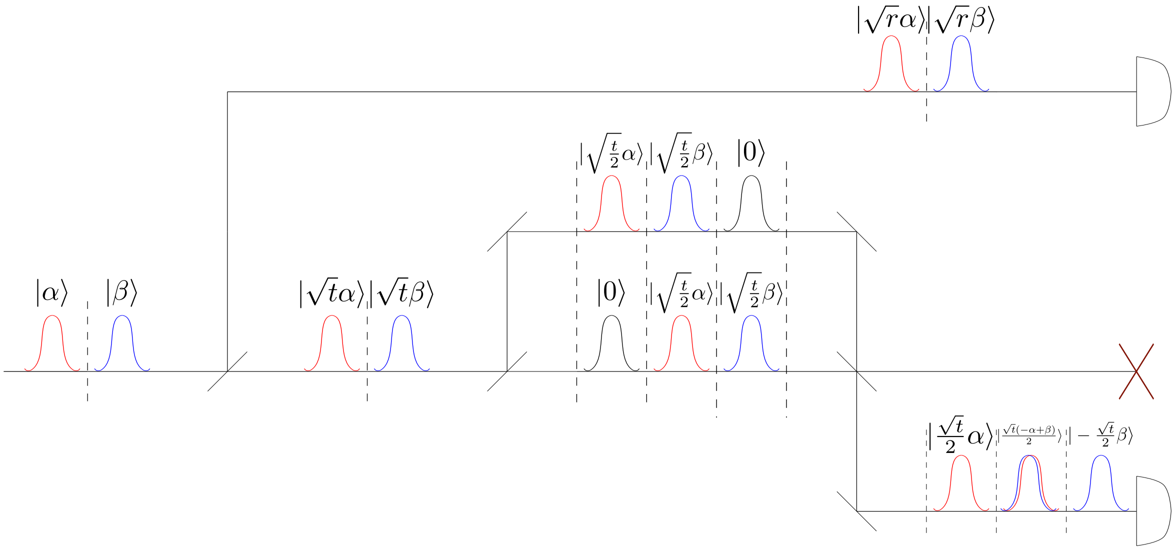


Figure B.1: Schematic of Bob's measurement setup where we have labelled the annihilation operators of each mode. The detector corresponding to constructive interference is not used for experimental simplicity.

B.1 Channel simulation statistics

We consider a loss-only quantum channel in our analysis of the protocol. This can be modelled as a beam splitter whose transmittance η depends on the distance l as $\eta = 10^{-\frac{\alpha l}{10}}$. As mentioned in Section 5.1.3, we consider the channel attenuation $\alpha = 0.16$ dB/km.

Alice randomises the phase of her laser states before encoding the state she sends them to Bob. However, note that Bob does not record measurements that measure the phase coherences across different protocol rounds. Thus, his measurement results will be independent of phase. For simplicity, we assume that Alice sends phase-coherent states while simulating the detection statistics that Bob would record. Under this special case where Bob receives coherent states, we can use Eq. (2.44) and Fig. 2.6 to compute the states that would arrive at Bob's detectors as depicted in Fig. B.2.



$|\sqrt{r}\alpha\rangle|\sqrt{r}\beta\rangle$
$|\sqrt{\frac{t}{2}}\alpha\rangle|\sqrt{\frac{t}{2}}\beta\rangle$
$|0\rangle$
$|\alpha\rangle$
$|\beta\rangle$
$|\sqrt{t}\alpha\rangle|\sqrt{t}\beta\rangle$
$|0\rangle$
$|\sqrt{\frac{t}{2}}\alpha\rangle|\sqrt{\frac{t}{2}}\beta\rangle$
$|\frac{\sqrt{t}}{2}\alpha\rangle|\frac{\sqrt{t}(-\alpha+\beta)}{2}\rangle|-\frac{\sqrt{t}}{2}\beta\rangle$

Figure B.2: The action of Bob's measurement setup on coherent states.

We first consider Alice encoding the 0 state $|0\rangle \otimes |\sqrt{\mu}\rangle$. After passing through the loss-only channel, Bob would receive the state $|0\rangle \otimes |\sqrt{\eta\mu}\rangle$. Thus, we have the case with $\alpha = 0$ and $\beta = \sqrt{\eta\mu}$. Thus, we can calculate the probability of a detector clicking in a particular time-slot from Eq. (2.51) as

$$\begin{aligned}
p_{a_4} &= 1 - e^{-r\eta\mu} \\
p_{a_3} &= 0 \\
p_{d_3} &= 1 - e^{-\frac{t}{4}\eta\mu} \\
p_{d_2} &= 1 - e^{-\frac{t}{4}\eta\mu} \\
p_{d_1} &= 0.
\end{aligned} \tag{B.1}$$

When Alice encodes the 1 state, Bob would receive the state $|\sqrt{\eta\mu}\rangle \otimes |0\rangle$ leading to very similar detector click probabilities

$$\begin{aligned}
p_{a_4} &= 0 \\
p_{a_3} &= 1 - e^{-r\eta\mu} \\
p_{d_3} &= 0 \\
p_{d_2} &= 1 - e^{-\frac{t}{4}\eta\mu} \\
p_{d_1} &= 1 - e^{-\frac{t}{4}\eta\mu}.
\end{aligned} \tag{B.2}$$

When Alice encodes the + state, Bob would receive the state $|\sqrt{\eta\mu/2}\rangle \otimes |\sqrt{\eta\mu/2}\rangle$ giving us the detector click probabilities

$$\begin{aligned}
p_{a_4} &= 1 - e^{-\frac{r}{2}\eta\mu} \\
p_{a_3} &= 1 - e^{-\frac{r}{2}\eta\mu} \\
p_{d_3} &= 1 - e^{-\frac{t}{8}\eta\mu} \\
p_{d_2} &= 0 \\
p_{d_1} &= 1 - e^{-\frac{t}{8}\eta\mu}.
\end{aligned} \tag{B.3}$$

We further assume that the detector clicks in each time bin are all independent of each other. Thus, we can get any click pattern by simply multiplying the corresponding probabilities. For e.g., the probability of getting a single-click in mode a_4 given that Alice encoded the 0 state is

$$p_0(a_4 = \text{click}; a_3, d_1, d_2, d_3 = \text{no click}) = p_{a_4}(1 - p_{a_3})(1 - p_{d_1})(1 - p_{d_2})(1 - p_{d_3}) \tag{B.4}$$

$$= (1 - e^{-r\eta\mu})e^{-\frac{t}{4}\eta\mu}e^{-\frac{t}{4}\eta\mu}. \tag{B.5}$$

We can repeat this process for all possible click patterns to obtain the full statistics. For the decoy-state SDPs and the key rate optimisation, we use coarse-grained statistics where we combine all click patterns that have more than one time-slot clicking in the same round.

B.2 Bound on weight outside projected subspace

The general method of using cross-clicks to bound the weight outside the projected subspace is taken from Chapter 2 of [44]. As stated in Appendix B.1, the probability of getting any click pattern does not depend on the absolute phase of the two pulses, just on the relative phase between them. Thus, phase-randomised states would give the same statistics as phase-coherent states. So we can conclude that Bob's measurements are block-diagonal in the total photon number of the two pulses. Given the probability that Bob received a state with n photons $p(n)$, the probability of getting a cross-click can then be written as

$$\begin{aligned} p(\mathrm{cc}) &= \sum_{n=0}^{\infty} p(n)p(\mathrm{cc}|n) && \text{(B.6)} \\ &= \sum_{n=0}^{N} p(n)p(\mathrm{cc}|n) + \sum_{n=N+1}^{\infty} p(n)p(\mathrm{cc}|n) && \text{(B.7)} \\ &\geq p(\leq N)p^{\min}(\mathrm{cc}| \leq N) + p(> N)p^{\min}(\mathrm{cc}| > N) && \text{(B.8)} \end{aligned}$$

where $p^{\min}(\mathrm{cc}| \leq N)$ denotes the minimum probability of a cross-click given the state has $\leq N$ photons. Using the fact that $p(\leq N) + p(> N) = 1$ and rearranging we get

$$p(> N) \leq \frac{p(\mathrm{cc}) - p^{\min}(\mathrm{cc}| \leq N)}{p^{\min}(\mathrm{cc}| > N) - p^{\min}(\mathrm{cc}| \leq N)}. \tag{B.9}$$

So in order to bound the weight outside the $\leq N$ subspace, we need to find the $p(\mathrm{cc}|n)$.

We have some choice when defining what to call cross-clicks. Here, we define a cross-click to be any click pattern that records a click in both the detectors while ignoring all clicks in mode d_2. We make this choice because it makes the calculations simpler as shall become apparent. Before we calculate the cross-click probability given that Bob received an n-photon state $p(\mathrm{cc}|n)$, we shall work out a simple example that will be useful later.

B.2.1 Statistics of a fock state through a beam splitter

Consider the setup shown in Fig. B.3. We send an n-photon state $|n\rangle$ at one of its input

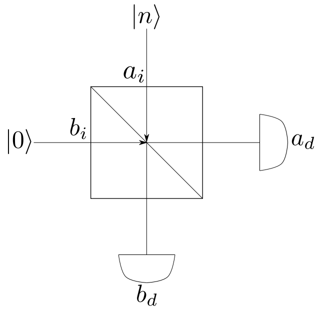


Figure B.3: Schematic of setup with detectors at the end of both the output ports of a beam splitter.

ports and the vacuum state at the other, and wish to find the detection probabilities. We can first transform the state using the beam splitter relations

$$|n,0\rangle_i = \frac{1}{\sqrt{n!}} a_i^{n\dagger}|0\rangle \tag{B.10}$$

$$\xrightarrow[\text{splitter}]{\text{beam}} \frac{1}{\sqrt{n!}}(\sqrt{r}a_d^\dagger + \sqrt{t}b_d^\dagger)^n|0\rangle \tag{B.11}$$

$$= \sum_{i=0}^{n} \frac{\sqrt{n!r^i t^{n-i}}}{i!(n-i)!} {a_d^i}^\dagger b_d^{n-i\dagger}|0\rangle \tag{B.12}$$

$$= \sum_{i=0}^{n} \sqrt{\frac{n!r^i t^{n-i}}{i!(n-i)!}}|i,n-i\rangle_d. \tag{B.13}$$

The probability of each of the detectors not clicking is given by the overlap of this state with the vacuum state in the corresponding space

$$p(a_d = \text{click}, b_d = \text{no-click}) = r^n \tag{B.14}$$

$$p(b_d = \text{click}, a_d = \text{no-click}) = t^n \tag{B.15}$$

Since the probability of neither detector clicking is 0, the probability of both detectors clicking together is given by $1-r^n-t^n$. Intuitively, we can think of the probability that only one detector clicks as the probability that all n photons get reflected (transmitted) where the probability of each photon getting reflected (transmitted) is given by the reflectance (transmittance) r (t) of the beam splitter. This intuition does not work in general when there is any kind of interference and so must be used with caution. However, due to our definition of cross-clicks we just have a single pulse going through multiple beam splitters before being measured.

B.2.2 Cross-click probabilities

We observe from the statistics calculated in Appendix B.1 that if we ignore the clicks in mode d_2, the statistics do not depends on either the phase or the relative phase of the two pulses. Thus, without loss of generality, we can always phase-randomise each pulse individually without changing the statistics. In other words, we can assume that our input state is a probabilistic mixture of $|m, n-m\rangle\langle m, n-m|$ where the total photon number is n.

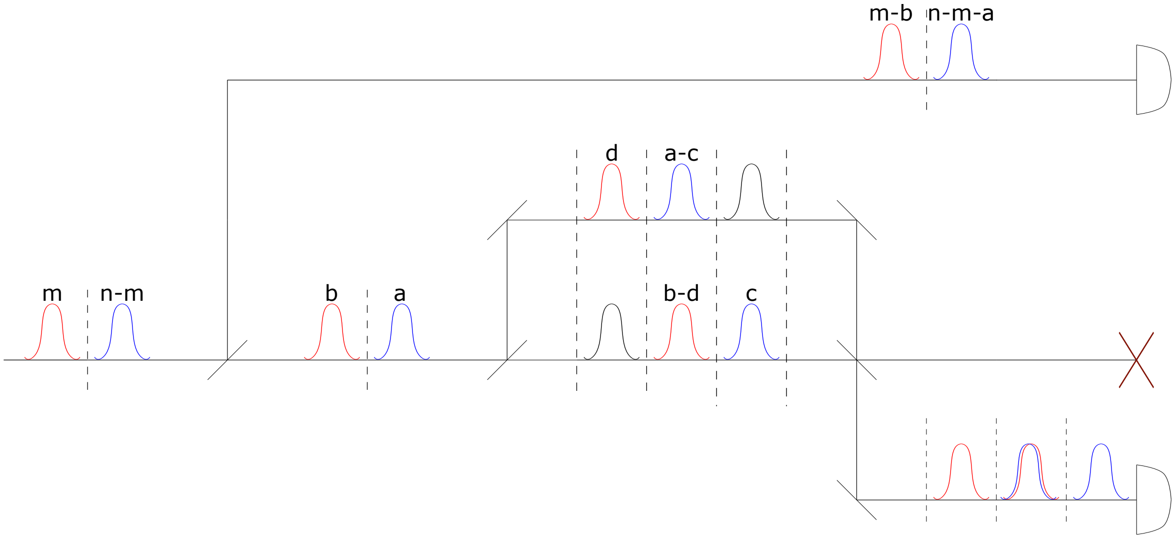


Figure B.4: Bob receives m and $n-m$ photons in the two pulses. Of that $a+b$ photons go into the X basis measurement line, and $n-a-b$ go to the Z basis detector. c and d photons go into the outside time-bins of the interferometer with the rest going into the middle time-bin.

As depicted in Fig. B.4, we start with m and $n-m$ photons in the 2 pulses. $a+b$ photons go into the X basis part of the beam splitter, b from the first pulse and a from the second. Thus, $n-a-b$ photons go to the Z basis measurement. The probability of this is $\binom{m}{b}\binom{n-m}{a}t^{a+b}(1-t)^{n-a-b}$. Of the b photons in the first pulse, d go to the long arm of the interferometer, and $b-d$ go to the short arm. From the second pulse, c go to the short arm and $a-c$ go to the long arm of the interferometer. The probability for this is

$\binom{a}{c}\binom{b}{d}\left(\frac{1}{2}\right)^{a+b}$. Finally, the probability that all $c+d$ photons go to the top detector of the Mach-Zehnder interferometer, denoted by a cross in Fig. (B.4), is $\left(\frac{1}{2}\right)^{c+d}$.

Taking all of this into account, the probability of a cross-click given an input state containing m and $n-m$ photons in the two pulses is

$$p(\mathrm{cc}|m,n-m) = \sum_{\substack{a+b\neq 0\\ a+b\neq n}} \binom{m}{b}\binom{n-m}{a} t^{a+b}(1-t)^{n-a-b}\left(\frac{1}{2}\right)^{a+b} \sum_{c+d\neq 0}\binom{a}{c}\binom{b}{d}\left(1-\left(\frac{1}{2}\right)^{c+d}\right) \tag{B.16}$$

where the last $\left(1-\left(\frac{1}{2}\right)^{c+d}\right)$ factor is to subtract the case when all $c+d$ photons go into the line with the detector we do not use.

We first calculate the second summation,

$$S(a,b) = \sum_{c+d\neq 0}\binom{a}{c}\binom{b}{d}\left(1-\left(\frac{1}{2}\right)^{c+d}\right) \tag{B.17}$$

$$= \sum_{c=0}^{a}\sum_{d=0}^{b}\binom{a}{c}\binom{b}{d}\left(1-\left(\frac{1}{2}\right)^{c+d}\right) - 0 \tag{B.18}$$

$$= 2^{a+b} - \left(\frac{3}{2}\right)^{a+b}. \tag{B.19}$$

Thus, the cross-click probability can be simplified as

$$p(\mathrm{cc}|m,n-m) = \sum_{\substack{a+b\neq 0\\ a+b\neq n}} \binom{m}{b}\binom{n-m}{a} t^{a+b}(1-t)^{n-a-b}\left(\frac{1}{2}\right)^{a+b} S(a,b) \tag{B.20}$$

$$= \sum_{\substack{a+b\neq 0\\ a+b\neq n}} \binom{m}{b}\left(\frac{t}{2}\right)^{b}(1-t)^{m-b}\binom{n-m}{a}\left(\frac{t}{2}\right)^{a}(1-t)^{n-m-a}\, S(a,b). \tag{B.21}$$

In order to simplify this, we compute

$$f(x,m) = \sum_{b=0}^{m} \binom{m}{b} \left(\frac{t}{2}\right)^{b} (1-t)^{m-b} x^{b} \tag{B.22}$$

$$= \left(1 - t + \frac{xt}{2}\right)^{m}. \tag{B.23}$$

So using Eq. (B.23) in Eq. (B.21) we get

$$p(\mathrm{cc}|m, n-m) = f(2,m)f(2,n-m) - f\left(\frac{3}{2}, m\right) f\left(\frac{3}{2}, n-m\right) - \left(\frac{2t}{2}\right)^{n} + \left(\frac{3t}{4}\right)^{n} \tag{B.24}$$

$$= \left(1 - t + \frac{2t}{2}\right)^{n} - \left(1 - t + \frac{3t}{4}\right)^{n} - t^{n} + \left(\frac{3t}{4}\right)^{n} \tag{B.25}$$

$$= 1 - t^{n} - \left(1 - \frac{t}{4}\right)^{n} + \left(\frac{3t}{4}\right)^{n}. \tag{B.26}$$

We observe that the cross-click probability does not depend on m which intuitively follows from the symmetry of our definition of cross-clicks.

Viewing the cross-click probability as a function of n

$$f(n) = 1 - t^{n} - \left(1 - \frac{t}{4}\right)^{n} + \left(\frac{3t}{4}\right)^{n}, \tag{B.27}$$

we look to show that the function is monotonically increasing. This would make it easy to find $p^{\min}(\mathrm{cc}| \leq N)$. We do this by considering

$$f(n+1) - f(n) = t^{n}(1-t) + \left(1 - \frac{t}{4}\right)^{n} \left(\frac{t}{4}\right) - \left(\frac{3t}{4}\right)^{n} \left(1 - \frac{3t}{4}\right) \tag{B.28}$$

and showing that this is positive for all positive integers n. We first note that $0 \leq t \leq 1$

which gives us

$$\begin{aligned} t &\leq 1 && \text{(B.29)} \\ 1 - t &\geq 0 && \text{(B.30)} \\ 1 - \frac{t}{4} - \frac{3t}{4} &\geq 0 && \text{(B.31)} \\ 1 - \frac{t}{4} &\geq \frac{3t}{4}. && \text{(B.32)} \end{aligned}$$

Raising both sides to the n^{th} power and multiplying both sides of the inequality by $\frac{t}{4}$,

$$\begin{aligned} \left(1 - \frac{t}{4}\right)^n \frac{t}{4} &\geq \left(\frac{3t}{4}\right)^n \frac{t}{4} && \text{(B.33)} \\ \left(1 - \frac{t}{4}\right)^n \frac{t}{4} - \left(\frac{3t}{4}\right)^n \left(1 - \frac{3t}{4}\right) &\geq \left(\frac{3t}{4}\right)^n (t - 1). && \text{(B.34)} \end{aligned}$$

Finally, adding $t^n(1-t)$ to both sides of the equation,

$$\begin{aligned} t^n(1-t) + \left(1 - \frac{t}{4}\right)^n \frac{t}{4} - \left(\frac{3t}{4}\right)^n \left(1 - \frac{3t}{4}\right) &\geq t^n(1-t) + \left(\frac{3t}{4}\right)^n (t-1) && \text{(B.35)} \\ f(n+1) - f(n) &\geq \left(t^n - \left(\frac{3t}{4}\right)^n\right)(1-t) \geq 0 && \text{(B.36)} \end{aligned}$$

where the last inequality follows from the fact that $t \geq \frac{3t}{4}$. Thus, $p^{\text{min}}(\text{cc}| \leq N) = p(\text{cc}|0) = 0$, and $p^{\text{min}}(\text{cc}| > N) = p(\text{cc}|N+1)$. Using this in Eq. (B.9),

$$p(> N) \leq \frac{p(\text{cc})}{1 - t^{N+1} - \left(1 - \frac{t}{4}\right)^{N+1} + \left(\frac{3t}{4}\right)^{N+1}} \tag{B.37}$$

where we can simulate the cross-click probability $p(\text{cc})$ as shown in Appendix B.1.

Appendix C

Source code for numerics

This appendix contains the source code used to generate the plots for the three-state protocol seen in Fig. 5.2. The primary function that bounds Eve's information about the key register is given below.

```
%% FUNCTION NAME: ThreeState_Eves_info
% This function allows you to bound Eve's information about the key register for the 3-state protocol using the generalised decoy-state method.
%
% Input:
%
% * N : Preserving up to N-photon subspace on Bob's side
%
% * ntilde : The value of $\tilde{n}$ that indicates the key rate block
% that we optimise.
%
% * q : The probability of Alice's sending a completely phase randomised state.
%
% * alpha : The vector containing the list of amplitudes $\sqrt{\mu}$ of the signal and decoy states produced by the laser with the first element being the signal intensity.
%
% * eta : Channel loss
%
```

```
% * t  : The fraction of photons going into the X basis.
%
% * iter: The maximum number of iterations in primalSolver.
%
% * rho0: Initial guess for optimisation in primalSolver.
%
% Output:
%
% * Eves_info: Lowerbound on the optimal key rate term  H($Z|E$) conditioned on Alice sending out the $|\tilde{n}\rangle$ signal.
%
function Eves_info = ThreeState_Eves_info(N, ntilde, q, alpha, eta, t, iter, rho0)


    %% Primal CVX Settings

    if nargin == 8 % if no input rho0
        rho0 = eye(dim)/dim;
        options.initmethod = 2;

    elseif nargin == 9 % specified initial guess for rho0
        options.initmethod = 1;
    end

    options.verbose = 1;
    options.linearconstrainttolerance = 1e-10;

    options.maxgap = 1e-7;
    options.maxiter = iter;

    %% POVMs, Krauss operators, and Key map
    dimA = 3; % dimension of Alice's system in the PM scheme (3 states)
    dimB = (N+1)*(N+2)/2;  % Bob's <=N-photon subspace dimension
    dim = dimA * dimB; % Total dimension of $\rho_{AB}^N$

    vecZero = zket(2,1); %|0>
    vecOne = zket(2,2); %|1>

    % Using the code that generates POVM for the COW protocol
    BPOVMs = COWPOVM(N, 1, t);
    % Coarse-graining the POVM elements to ignore all clicks in the "+" detector
    BPOVM = CoarseGrainedPOVMs(BPOVMs);
    % Coarse-graining the POVM elements to group all the multi-click events into one POVM element to save computational time
    BobPOVM = MultiCoarseGrainedPOVMs(BPOVM);

    KraussBob = MultiBobKrauss(BobPOVM);
    KraussAlice = kron(vecZero*zket(3,1)') + ...
                  kron(vecOne*zket(3,2)');
    krausOp = {kron(KraussAlice, KraussBob)};

    keyProj1 = kron(vecZero*vecZero', eye(dimB));
    keyProj2 = kron(vecOne*vecOne', eye(dimB));
    keyMap = {keyProj1, keyProj2};

    AlicePOVM = {diag(zket(3,1)), ...
                 diag(zket(3,2)), ...
                 diag(zket(3,3))};
    pA = [1/3; 1/3; 1/3]; % Probability of Alice sending each signal state
    coarsestats = [];

    % Finding the cut-off M on the signal states that Alice sends
    M = 17;
    for signalstateindex = 0:1:10
        if(poisspdf(signalstateindex,alpha(1)^2) <= eps)
            M = signalstateindex-1;
            break;
        end
    end

```

```
%% Simulating the loss-only channel and finding the different $\gamma_{ij}$ for the different signal and decoy states.

for ampindex = 1:1:length(alpha)

% Fine-grained statistics for the protocol
[finestats{3*ampindex-2}, finestats{3*ampindex-1}, finestats{3*ampindex}] = ChannelSimulationStatistics(sqrt(eta)*alpha(ampindex), t);
%Coarse-grained statistics with all multi-clicks added up
coarsestats = [coarsestats; MultiChannelSimulationStatistics(finestats{3*ampindex-2}, finestats{3*ampindex-1}, finestats{3*ampindex})];

mu = alpha(ampindex)^2;
% Projected state
rho{ampindex} = zeros(M+1);
for signalstateindex = 0:1:M
    for j = 0:1:M
        if poisspdf(signalstateindex,mu) > eps
            if signalstateindex == j
                rho{ampindex}(signalstateindex+1,j+1) = poisspdf(signalstateindex,mu);
            else
                rho{ampindex}(signalstateindex+1,j+1) = (1-q)*sqrt(poisspdf(signalstateindex,mu)*poisspdf(j,mu));
            end
        end
    end
end
end
Fproj = trace(rho{1}); % Fidelity between approx. rho and infinite rho
if Fproj>1
    Fproj = 1;
end
```

```
    epsilonproj = 2*sqrt(1-Fproj) + 2*sqrt(eps); % Accounting for machine error
    if q==1
        epsilonproj = 1-Fproj+eps; % Tighter bound if rho commutes with projector
    end

    % Eigenvector/value of the projected signal state with the maximum distance between this eigenvector and the infinite dimensional states eigenvector
    [psi,lambda,epsilonvec] = closestEigenvectors(rho{1}, ...
                                                  epsilonproj, ntilde+1);
    if q == 1
        epsilonvec = options.linearconstrainttolerance; % Already diagonal case
    end

    for signalstateindex = 1:1:3
        expect = [];
        for ampindex = 1:1:length(alpha)
            % Upper bound on greater than equal to N subspace
            PgeqNx = (1-ThreeStateleqN (N, t, finestats{3*ampindex-3+signalstateindex}));
            Wb(ampindex) = PgeqNx;
            %Constraining statistics and POVMs for decoy-state
            consStat = coarsestats(3*ampindex - 3 + ...
                                  signalstateindex,:);
            consPOVM = BobPOVM;
            expect = [expect, consStat'];
        end
        % We take the same objective and constraining POVMs
        [YL{signalstateindex}, YU{signalstateindex}] = decoyBoundsSDP(consPOVM,consPOVM, rho, {psi*psi'}, expect, Wb, q);
    end

    % POVM and statistic bounds that will go into the primal solver
    Gamma = {};
    gammaL = [];
    gammaU = [];

    for signalstateindex = 1:1:3
        for j = 1:length(consPOVM)
            Gamma = [Gamma; kron(AlicePOVM{signalstateindex}, consPOVM{j})];
            gammaL = [gammaL; max(pA(signalstateindex)*YL{signalstateindex}(j)-epsilonvec,0)];
            gammaU = [gammaU; min(pA(signalstateindex)*YU{signalstateindex}(j)+epsilonvec,pA(signalstateindex))];
        end
    end

    % Encoding isometries to obtain the partial trace constraint
    V{1} = kron(zket(M+1,1),eye(M+1));
    V{2} = kron(eye(M+1),zket(M+1,1));
    V{3} = 0;
    for signalstateindex = 0:1:M
        for j = 0:1:signalstateindex
            V{3} = V{3} + sqrt(nchoosek(signalstateindex,j)/2^signalstateindex)*kron(zket(M+1,j+1),zket(M+1,signalstateindex-j+1))*zket(M+1,signalstateindex+1)';
        end
    end
    knownStates = {};
    for ampindex = 1:1:length(alpha)
        knownStates{3*ampindex-2} = V{1}*rho{ampindex}*V{1}';
        knownStates{3*ampindex-1} = V{2}*rho{ampindex}*V{2}';
        knownStates{3*ampindex} = V{3}*rho{ampindex}*V{3}';
    end
    W=0;
    rhoA = zeros(3);
    for signalstateindex = 1:1:3
        for j = 1:1:3
```

```
            rhoA(signalstateindex,j) = sqrt(pA(signalstateindex)*pA(j))*psi'*V{signalstateindex}'*V{j}*psi;
        end
    end

    %% Primal CVX
    [fvalvec,rho,primalfvals,gaps,flagprimal] = primalSolverAB(rho0, keyMap, rhoA,...
                            epsilonvec, Gamma, gammaL, gammaU, W, krausOp);

    % Perturb the suboptimal rho if its eigenvalues are not all positive to avoid the 'zetaEp' in 'step2Solver.m' being too big
    eigMin = lambda_min(rho);
    if eigMin < 0
        epsilon = -eigMin*dim;
        if epsilon < 1
            rho = (1-epsilon)*rho + epsilon *eye(dim)/dim;
            disp('lambda_min(Rho) with lambda_min before perturbation = '+string(eigMin));
            disp('lambda_min(Rho) with lambda_min = '+string(lambda_min(rho)));
        else
           disp('Warning: epsilon >= 1');
        end
    end
    Rho = rho;

    %% Dual CVX
    disp('running dual cvx...');
    tic
    cons = zeros(1, numel(Gamma));
```

```
        for iBasisElm = 1:numel(Gamma)
                if real(trace(rho * Gamma{iBasisElm})) > gammaU(iBasisElm)
                    cons(iBasisElm) = abs(real(trace(rho * Gamma{iBasisElm}) - gammaU(iBasisElm)));
                end
        end
        option2 = {};
        option2.epsilonprime = max(cons); % maximum constraint violation
        disp('max cons = '+string(max(cons)));
        option2.epsilonprimeprime = 1e-10;

        [dualfvals, debugging, flagdual] = step2SolverAB(Rho, keyMap, rhoA, epsilonvec, Gamma, gammaL, gammaU, sum(W), krausOp, option2);

       % Multiplying the optimal value by the eigenvalue to get the key rate contribution
        if dualfvals <= 0 || (lambda-epsilonproj) <= 0
            keyrate = 0;
        else
            keyrate = dualfvals*(lambda-epsilonproj);
        end
end
```

The primal and dual solvers can be found in [60]. The rest of the functions used to generate these key rate contributions have been given below.

C.1 Protocol description

Code to generate the POVMs

```
%% Bob's Flag-State Squashed POVM for COW protocol in <=N subspace
%
% Input:
```

```
%
% * N   : photon number cut-off corresponding to <=N-photon subspace
%
% * n   : number of bits in a block. For 3-State Protocol, this can be set to 1.
%
% * t   : fraction of photons going into the X basis. For DPS protocol, this can be set to 1.
%
% Output:
%
% * POVM : a cell of the POVM elements
%
function POVM = COWPOVM(N, n, t)

    dim = (N+1) * nchoosek(2*n+N, 2*n-1)/(2*n); % Distributing <= N photons in $2n$ modes $\sum_{i=0}^{N} \binom{(2n+i-1)}{(2n-1)}$

    % The next lines give the projector onto the <= N photon subspace with block diagonal structure corresponding to total photon number in the n-bit block.
    projector = zeros(dim, (N+1)^(2*n));
    projector(1,1) = 1;

    for photonnumber = 1:1:N
        % Forming vector of all the ways that $x_1 + .... + x_{2n} = photonnumber$
        temp = sym2cell(feval(symengine, 'combinat::compositions', photonnumber, strcat('Length = ',int2str(2*n)), 'MinPart = 0'));

        l = size(temp,2); % Count number of such combinations that are there.
            for i = 1:1:l % Loop over all combinations
                combtemp = sym2cell(temp{i});
                index = 1; % Column that will be non-zero in our
```

```
               projection matrix
            for j = 1:1:2*n % Loop over all the elements in the combination
                index = index + combtemp{j}*(N+1)^(j-1); % |....,i,j,k> has only the 1+(N+1)*k+(N+1)^2*j+(N+1)^3*i+... element non-zero. Thus, this is the column of the projector that will be non-zero
            end
            projector((photonnumber) * nchoosek(2*n+photonnumber-1, 2*n-1)/(2*n)+i, index) = 1;
        end
end
% Generic annihilation operator restricted to <=N-photon subspace
a = zeros(N+1);

for k = 1:1:N
    a(k,k+1) = sqrt(k);
end

% Input annihilation operators $u_i$ where the different spaces in the tensor product represent different time-bins

u = cell(2*n,1);
for i = 1:1:2*n
    u{i} = a;
    for j=1:1:i-1
    u{i}=kron(u{i},eye(N+1)); % $u_i = a \otimes \mathbb{I}^{\otimes(i-1)}$
    end
    for j=1:1:(2*n-i)
        u{i}=kron(eye(N+1),u{i}); % $u_i = \mathbb{I}^{\otimes(2n-i)} \otimes a \otimes \mathbb{I}^{\otimes(i-1)}$
    end
end
% Project these operators into the <= N photon subspace
for i = 1:1:2*n
    u{i} = projector * u{i} * projector';
    end

    % Vacuum vector |0,0,0,.. 0 2n times> in the <= N photon subspace
    vac = projector * zket((N+1)^(2*n),1);

    % Annihilation operators that represent the destructive interference of the adjacent time-bins.
    d = cell(2*n+1,1);
    % Annihilation operators that represent the constructive interference of the adjacent time-bins.
    p = cell(2*n+1,1);

    % Note that while writing these annihilation operators we have ignored the terms that would come from the vacuum components of the beam splitter as when writing the POVMs and imposing the condition that said state was the vacuum state, these annihilation operators would not play a role. See Varun Narasimhachar's masters thesis, equations A.5 and A.6 for a more detailed explanation.


    % Annihilation operators for the first time slot
    d{1} = -0.5*sqrt(t)*u{1};
    c{1} = 0.5*sqrt(t)*u{1};

    % Annihilation operators for the last time slot
    d{2*n+1} = 0.5*sqrt(t)*u{2*n};
    c{2*n+1} = 0.5*sqrt(t)*u{2*n};

    % Annihilation operators for middle-slots
    for i = 2:1:2*n
        d{i} = 0.5*sqrt(t)*(u{i-1}-u{i});
        c{i} = 0.5*sqrt(t)*(u{i-1}+u{i});
    end

    % Creating a cell that contains all the annihilation operators including the basis choice beam splitter
    w = cell(6*n+2,1);
    for i = 1:1:2*n
        w{i} = d{i}; % destructive interference operators in the X-basis
        w{2*n+1+i} = c{i}; % constructive interference operators in X-basis
        w{4*n+2+i} = sqrt(1-t)*u{i}; % operators for the Z-basis
    end
    w{2*n+1} = d{2*n+1};
    w{4*n+2} = c{2*n+1};

    % Number of detectors in the protocol. Here we think of clicks in different time slots as different detectors.
    numberofdetectors = 6*n+2;

    POVM{1,1} = vac*vac'; % no-click event (index2-1 = 0)

    % The second index refers to the number of detectors that click + 1
    % Within each block of click patterns (the first index), the POVMs are arranged as per how nchoosek orders them. To know which POVM corresponds to which detection event, look at how the function nchoosek orders them.

    for detectionevents = 1:1:numberofdetectors

        annOp = nchoosek(w, detectionevents); % Creates an array of all possible combinations of i annihilation operators for the i click event.
        l = size(annOp, 1); % Find number of combinations of i annihilation operators that exist.

        % Taking all possible combinations of multi-clicks and adding them to the set of POVMs.
        for j = 1:1:l
            temp = annOp(j,:);
            tempPOVM = zeros(dim);
            for photonnumber = 1:1:N
```

```
                % Forms the POVM for the combination stored in temp for photonnumber and sums over these POVMs till cut-off
                tempPOVM = tempPOVM + Sum({0}, temp, photonnumber, vac);
            end
              POVM{j,detectionevents+1} = tempPOVM; % The second index is the number of detectors that click + 1
        end
    end

end
```

This code forms the annihilation operators for the different click events and uses the function 'Sum' given below to use those to get the final POVMs.

```
%% Forming the POVM from a specified set of annihilation operators assuming N photons
%
% We have to sum over all photon numbers for the different detectors and time-slots with the constraint that the sum of all the photon numbers in the detector is N. Thus, we have multiple summations that are dependent on each other through the constraint.
%
% This code can be used to find fine-grained POVM elements for any protocol if you know the annihilation operators for each detector.
%
% Input:
%
% * indices   : Vector of numbers that signify the indices for the previous summations. When calling the function for the first time, this should be initialised to 0.
%
% * annOp : Cell containing all the annihilation operators for the different detectors and time slots.
%
```

```
% * N   : Total number of photons that are detected.
%
% * vac : The vacuum state |0> which tells us the dimension of the space.
%
% Output:
%
% * P : a (2n,1) cell which will be the POVMs after adding the contribution due to the detector and time-slot represented by the annihilation operator m

function tempstorage = Sum(indices, annOp, N, vac)

    if N >= length(annOp) % Need to have more photons than click events
        remainingphotons = N-sum([indices{:}])-length(annOp)+1; % The total number of photons left for the rest of the summations

        if(length(annOp) > 1)
            tempstorage = 0; % initialising sum as 0
            for i = 1:1:remainingphotons
                v = annOp;
                v(1) = []; % Deleting the annihilation operator just used
                tempindices = indices;
                tempindices{end + 1} = i; % Adding the number of photons detected in this operator to the indices vector
                tempstorage = tempstorage + annOp{1}'^i * ...
                    Sum(tempindices, v, N, vac) * annOp{1}^i/factorial(i);
            end
        else % Base case when we have just one annihilation operator
            tempstorage = annOp{1}'^remainingphotons * vac * vac' * ...
                annOp{1}^remainingphotons/factorial(
                    remainingphotons);
        end
    else % Zero matrix of whatever dimension space we have
        tempstorage = vac*vac'-vac*vac';
    end
end
```

These POVMs include the '+' detector which we do not have in the experimental setup we consider. So we need to coarse-grain these to get the POVMs for the experimental setup. This is done by the function 'CoarseGrainedPOVMs' shown below.

```
%% Bob's coarse-grained POVM operators  for the 3-State protocol
% We do not actually have the "plus" detector in the experimental implementation of the protocol. So any detection events with clicks in the "plus" detector and clicks in any other detector would be coarse-grained to clicks in just the other detector and all clicks in just the "plus" detector would be coarse grained to the no-click event.
%
% Input:
%
% * POVM   : POVMs as outputted by the COWPOVM function with n=1
%
% Output:
%
% * CPOVM : Coarse grained POVMs


function CPOVM = CoarseGrainedPOVMs(POVM)

   CPOVM = cell(10, 6); % The second index i is for the number of detectors that click and the first will tell us exactly what (i-1)-click event has occured.

   %% The coarse grained no-click event

   CPOVM{1,1} = POVM{1,1} + ...                              % Vacuum event
                    POVM{4,2} + POVM{5,2} + POVM{6,2} + ...   %
                    Single-click event
                POVM{19,3} + POVM{20,3} + POVM{23,3} + ...% Double-click event
                POVM{47,4};                                % Triple-click event



%% Coarse grained single-click events

% The "-" detector in the first time slot (1)
CPOVM{1,2} = POVM{1,2} + ...                              % Single-click event
                POVM{3,3} + POVM{4,3} + POVM{5,3} + ...   % Double-click event
                POVM{12,4} + POVM{13,4} + POVM{16,4} + ...% Triple-click event
                POVM{26,5};                                % Quadruple-click event

% The "-" detector in the middle time slot (2)
CPOVM{2,2} = POVM{2,2} + ...                              % Single-click event
                POVM{9,3} + POVM{10,3} + POVM{11,3} + ... % Double-click event
                POVM{27,4} + POVM{28,4} + POVM{31,4} + ...% Triple-click event
                POVM{46,5};                                % Quadruple-click event

% The "-" detector in the last time slot (3)
CPOVM{3,2} = POVM{3,2} + ...                              % Single-click event
                POVM{14,3} + POVM{15,3} + POVM{16,3} + ...% Double-click event
                POVM{37,4} + POVM{38,4} + POVM{41,4} + ...% Triple-click event
                POVM{56,5};                                %
```

```
                    Quadruple-click event

% The "0/1" detector in the first time slot (7)
CPOVM{4,2} = POVM{7,2} + ...                             %
   Single-click event
             POVM{21,3} + POVM{24,3} + POVM{26,3} + ...%
                Double-click event
             POVM{48,4} + POVM{50,4} + POVM{53,4} + ...%
                Triple-click event
             POVM{66,5};                                %
                Quadruple-click event

% The "0/1" detector in the last time slot (8)
CPOVM{5,2} = POVM{8,2} + ...                             %
   Single-click event
             POVM{22,3} + POVM{25,3} + POVM{27,3} + ...%
                Double-click event
             POVM{49,4} + POVM{51,4} + POVM{54,4} + ...%
                Triple-click event
             POVM{67,5};                                %
                Quadruple-click event

%% Coarse-grained double-click events

% Double-click in 1 and 2
CPOVM{1,3} = POVM{1,3} + ...                             %
   Double-click event
             POVM{2,4} + POVM{3,4} + POVM{4,4} + ...    %
                Triple-click event
             POVM{6,5} + POVM{7,5} + POVM{10,5} + ...   %
                Quadruple-click event
             POVM{11,6};                                %
                Quintuple-click event

% Double-click in 1 and 3
CPOVM{2,3} = POVM{2,3} + ...                             %
   Double-click event
             POVM{7,4} + POVM{8,4} + POVM{9,4} + ...    %
```

```
                    Triple−click events
                POVM{16,5} + POVM{17,5} + POVM{20,5} + ...%
                    Quadruple−click events
                POVM{21,6};                               %
                    Quintuple−click event

% Double−click in 1 and 7
CPOVM{3,3} = POVM{6,3} + ...                              %
    Double−click event
                POVM{14,4} + POVM{17,4} + POVM{19,4} + ...%
                    Triple−click event
                POVM{27,5} + POVM{29,5} + POVM{32,5} + ...%
                    Quadruple−click event
                POVM{31,6};                               %
                    Quintuple−click event

% Double−click in 1 and 8
CPOVM{4,3} = POVM{7,3} + ...                              %
    Double−click event
                POVM{15,4} + POVM{18,4} + POVM{20,4} + ...%
                    Triple−click event
                POVM{28,5} + POVM{30,5} + POVM{33,5} + ...%
                    Quadruple−click event
                POVM{32,6};                               %
                    Quintuple−click event

% Double−click in 2 and 3
CPOVM{5,3} = POVM{8,3} + ...                              %
    Double−click event
                POVM{22,4} + POVM{23,4} + POVM{24,4} + ...%
                    Triple−click event
                POVM{36,5} + POVM{37,5} + POVM{40,5} + ...%
                    Quadruple−click event
                POVM{36,6};                               %
                    Quintuple−click event

% Double−click in 2 and 7
CPOVM{6,3} = POVM{12,3} + ...                             %
```

```
    Double-click event
                POVM{29,4} + POVM{32,4} + POVM{34,4} + ...% Triple-click event
                POVM{47,5} + POVM{49,5} + POVM{52,5} + ...% Quadruple-click event
                POVM{46,6};                               % Quintuple-click event

    % Double-click in 2 and 8
    CPOVM{7,3} = POVM{13,3} + ...                          % Double-click event
                POVM{30,4} + POVM{33,4} + POVM{35,4} + ...% Triple-click event
                POVM{48,5} + POVM{50,5} + POVM{53,5} + ...% Quadruple-click event
                POVM{47,6};                               % Quintuple-click event

    % Double-click in 3 and 7
    CPOVM{8,3} = POVM{17,3} + ...                          % Double-click event
                POVM{39,4} + POVM{42,4} + POVM{44,4} + ...% Triple-click event
                POVM{57,5} + POVM{59,5} + POVM{62,5} + ...% Quadruple-click event
                POVM{51,6};                               % Quintuple-click event

    % Double-click in 3 and 8
    CPOVM{9,3} = POVM{18,3} + ...                          % Double-click event
                POVM{40,4} + POVM{43,4} + POVM{45,4} + ...% Triple-click event
                POVM{58,5} + POVM{60,5} + POVM{63,5} + ...% Quadruple-click event
                POVM{52,6};                               % Quintuple-click event

```

```
% Double-click in 7 and 8
CPOVM{10,3} = POVM{28,3} + ...                                % Double-click event
              POVM{52,4} + POVM{55,4} + POVM{56,4} + ...% Triple-click event
              POVM{68,5} + POVM{69,5} + POVM{70,5} + ...% Quadruple-click event
              POVM{56,6};                               % Quintuple-click event

%% Coarse-grained triple-click events

% Triple-click in 1,2 and 3
CPOVM{1,4} = POVM{1,4} + ...                                  % Triple-click event
             POVM{1,5} + POVM{2,5} + POVM{3,5} + ...    % Quadruple-click event
             POVM{1,6} + POVM{2,6} + POVM{5,6} + ...    % Quintuple-click event
             POVM{1,7};                                 % Sextuple-click event

% Triple-click in 1,2 and 7
CPOVM{2,4} = POVM{5,4} + ...                                  % Triple-click event
             POVM{8,5} + POVM{11,5} + POVM{13,5} + ... % Quadruple-click event
             POVM{12,6} + POVM{14,6} + POVM{17,6} + ...% Quintuple-click event
             POVM{11,7};                                % Sextuple-click event

% Triple-click in 1,2 and 8
CPOVM{3,4} = POVM{6,4} + ...                                  % Triple-click event
             POVM{9,5} + POVM{12,5} + POVM{14,5} + ... % Quadruple-click event
             POVM{13,6} + POVM{15,6} + POVM{18,6} + ...%
```

```
            Quintuple-click event
          POVM{12,7};                                %
            Sextuple-click event

% Triple-click in 1,3 and 7
CPOVM{4,4} = POVM{10,4} + ...                        %
  Triple-click event
          POVM{18,5} + POVM{21,5} + POVM{23,5} + ...%
            Quadruple-click event
          POVM{22,6} + POVM{24,6} + POVM{27,6} + ...%
            Quintuple-click event
          POVM{16,7};                                %
            Sextuple-click event

% Triple-click in 1,3 and 8
CPOVM{5,4} = POVM{11,4} + ...                        %
  Triple-click event
          POVM{19,5} + POVM{22,5} + POVM{24,5} + ...%
            Quadruple-click event
          POVM{23,6} + POVM{25,6} + POVM{28,6} + ...%
            Quintuple-click event
          POVM{17,7};                                %
            Sextuple-click event

% Triple-click in 1,7 and 8
CPOVM{6,4} = POVM{21,4} + ...                        %
  Triple-click event
          POVM{31,5} + POVM{34,5} + POVM{35,5} + ...%
            Quadruple-click event
          POVM{33,6} + POVM{34,6} + POVM{35,6} + ...%
            Quintuple-click event
          POVM{21,7};                                %
            Sextuple-click event

% Triple-click in 2,3 and 7
CPOVM{7,4} = POVM{25,4} + ...                        %
  Triple-click event
          POVM{38,5} + POVM{41,5} + POVM{43,5} + ...%
```

```
                 Quadruple-click event
             POVM{37,6} + POVM{39,6} + POVM{41,6} + ...% Quintuple-click event
             POVM{22,7};                                % Sextuple-click event

% Triple-click in 2,3 and 8
CPOVM{8,4} = POVM{26,4} + ...                           % Triple-click event
             POVM{39,5} + POVM{42,5} + POVM{44,5} + ...% Quadruple-click event
             POVM{38,6} + POVM{40,6} + POVM{42,6} + ...% Quintuple-click event
             POVM{23,7};                                % Sextuple-click event

% Triple-click in 2,7 and 8
CPOVM{9,4} = POVM{36,4} + ...                           % Triple-click event
             POVM{51,5} + POVM{54,5} + POVM{55,5} + ...% Quadruple-click event
             POVM{48,6} + POVM{49,6} + POVM{50,6} + ...% Quintuple-click event
             POVM{27,7};                                % Sextuple-click event

% Triple-click in 3,7 and 8
CPOVM{10,4} = POVM{46,4} + ...                          % Triple-click event
              POVM{61,5} + POVM{64,5} + POVM{65,5} + ...% Quadruple-click event
              POVM{53,6} + POVM{54,6} + POVM{56,6} + ...% Quintuple-click event
              POVM{28,7};                               % Sextuple-click event

%% Coarse-grained quadruple-click events

```

```
% Quadraple-click in 1,2,3 and 7
CPOVM{1,5} = POVM{4,5} + ...                               % Quadruple-click event
             POVM{3,6} + POVM{6,6} + POVM{8,6} + ...       % Quintuple-click event
             POVM{2,7} + POVM{4,7} + POVM{7,6} + ...       % Sextuple-click event
             POVM{1,8};                                    % Septuple-click event

% Quadraple-click in 1,2,3 and 8
CPOVM{2,5} = POVM{5,5} + ...                               % Quadruple-click event
             POVM{4,6} + POVM{7,6} + POVM{9,6} + ...       % Quintuple-click event
             POVM{3,7} + POVM{5,7} + POVM{8,6} + ...       % Sextuple-click event
             POVM{2,8};                                    % Septuple-click event

% Quadraple-click in 1,2,7 and 8
CPOVM{3,5} = POVM{15,5} + ...                              % Quadruple-click event
             POVM{16,6} + POVM{19,6} + POVM{20,6} + ...% Quintuple-click event
             POVM{13,7} + POVM{14,7} + POVM{15,6} + ...% Sextuple-click event
             POVM{6,8};                                    % Septuple-click event

% Quadraple-click in 1,3,7 and 8
CPOVM{4,5} = POVM{25,5} + ...                              % Quadruple-click event
             POVM{26,6} + POVM{29,6} + POVM{30,6} + ...% Quintuple-click event
             POVM{18,7} + POVM{19,7} + POVM{20,6} + ...% Sextuple-click event
             POVM{7,8};                                    %
```

```
                    Septuple-click event

    % Quadraple-click in 2,3,7 and 8
    CPOVM{5,5} = POVM{45,5} + ...                            % Quadruple-click event
                 POVM{41,6} + POVM{44,6} + POVM{45,6} + ...% Quintuple-click event
                 POVM{24,7} + POVM{25,7} + POVM{26,6} + ...% Sextuple-click event
                 POVM{8,8};                                % Septuple-click event

    %% Coarse-grained all click event

    CPOVM{1,6} = POVM{10,6} + ...                            % Quintuple-click event
                 POVM{6,7} + POVM{9,7} + POVM{10,6} + ...  % Sextuple-click event
                 POVM{3,8} + POVM{4,8} + POVM{5,8} + ...   % Septuple-click event
                 POVM{1,9};                                % All-click event
end
```

We further coarse grain the POVMs to save computational power to group all multi-clicks into one POVM element.

```
%% Bob's coarse-grained POVM operators for the 3-State protocol with multi-click coarse graining
% We coarse grain the POVMs to save space such that all multi-click POVMs are coarse grained into one POVM
%
% Input:
% * POVM   : POVMs as outputted by the CoarseGrainedPOVMs function
%
% Output:
% * CPOVM : Coarse grained POVMs

```

```
function CPOVM = MultiCoarseGrainedPOVMs(POVM)

    CPOVM = cell(1, 7); % The first 6 are the vacuum and single-click events while the last one is the multi-click POVM event

    CPOVM{1,1} = POVM{1,1};          % No-click event
    n = length(POVM{1,1});
    tempsum = POVM{1,1};
    for i = 1:1:5
        CPOVM{1,i+1} = POVM{i,2};    % Single-click events
        tempsum = tempsum + POVM{i,2};
    end
    CPOVM{1,7} = eye(n) - tempsum;       % Multi-click event

end
```

Bob's Krauss operator

```
%% Bob's Krauss operators  for the 3-State protocol
% The announcements correspond to the basis in which Bob got an outcome, or if the detection event should be discarded. The chosen post-processing here is as follows:
% 1) The 0/1 basis corresponds to all clicks in the Z-basis and outer clicks in the "minus" detector.
% 2) The +/- basis corresponds to all clicks in the middle time slot of the "minus" detector which do not map to the key and so are discarded.
% 3) All multi-clicks are discarded.
%
% Input:
%
% * POVM   : POVMs as outputted by the MultiCoarseGrainedPOVM function
%
% Output:
%
% * BKrauss : a cell of the Krauss operators
```

```

function BKrauss = MultiBobKrauss(CPOVM)

    dim = size(CPOVM{1,1},1); % Dimension of the POVMs

    % Coarse grained POVM operators
    PZero = zeros(dim); % Measurement result 0.
    POne = zeros(dim); % Measurement result 1.

    PZero = CPOVM{1,5} + ...% Single-click in just the 0/1 detector. (7)
            CPOVM{1,2};% Single-click in "-" detector in the first time bin (1)
    POne = CPOVM{1,6} + ...% Single-click in just the 0/1 detector. (8)
           CPOVM{1,4};% Single-click in "-" detector in the last time bin (3)

    % Krauss operator for the 0/1 basis
    BKrauss = sqrt(PZero+POne);
end
```

C.2 Channel simulation

We assume a loss-only channel and simulate it to obtain both the fine-grained and the coarse-grained statistics where all the multi-clicks are taken to be one event.

Fine-grained statistics

```
%% Bob's observed detection statistics for the 3-state protocol with a loss-only channel
%
% We assume that we have a loss-only channel and so whenever Alice sends coherent states, Bob receives coherent states with a reduced amplitude.
% Alice sends the states $|\sqrt{\mu},0\rangle$, $|0,\sqrt{\mu}\rangle$ and $|\sqrt{\frac{\mu}{2}},\sqrt{\frac{\mu}{2}}\rangle$ with probability 1/3 each. In the faithful implementation after
```

```
channel loss, the states going into Bob's apparatus will be essentially the same as what Alice sends except with $\sqrt{\mu}$ replaced with $\alpha = \sqrt{\eta\mu}$, the amplitude of the coherent state after loss.
%
% Input:
%
% * alpha    : Amplitude of the coherent states entering Bob's apparatus.
%
% * t        : Fraction of photons going into the X-basis.
%
% Output:
%
% * pzeroalpha : Cell containing the probabilities corresponding to the different POVM elements conditioned on Alice having sent the zeroalpha state.
%
% * palphazero : Cell containing the probabilities corresponding to the different POVM elements conditioned on Alice having sent the zeroalpha state.
%
% * palphaalpha : Cell containing the probabilities corresponding to the different POVM elements conditioned on Alice having sent the alphaalpha state.


function [pzeroalpha, palphazero, palphaalpha] = ChannelSimulationStatistics(alpha, t)

    alpha = abs(alpha);

    czeroalpha = cell(5,1);
    czeroalpha{1} = exp(t*alpha^2/4)-1;
    czeroalpha{2} = exp(t*alpha^2/4)-1;
    czeroalpha{3} = 0;
    czeroalpha{4} = exp((1-t)*alpha^2)-1;
    czeroalpha{5} = 0;

    calphazero = cell(5,1);
    calphazero{1} = 0;
    calphazero{2} = exp(t*alpha^2/4)-1;
    calphazero{3} = exp(t*alpha^2/4)-1;
    calphazero{4} = 0;
    calphazero{5} = exp((1-t)*alpha^2)-1;

    calphaalpha = cell(5,1);
    calphaalpha{1} = exp(t*alpha^2/8)-1;
    calphaalpha{2} = 0;
    calphaalpha{3} = exp(t*alpha^2/8)-1;
    calphaalpha{4} = exp((1-t)*alpha^2/2)-1;
    calphaalpha{5} = exp((1-t)*alpha^2/2)-1;

    a = exp(-alpha^2); % |<0|alpha>|^2
    pzeroalpha{1,1} = a*exp(t*alpha^2/2);
    palphazero{1,1} = a*exp(t*alpha^2/2);
    palphaalpha{1,1} = a*exp(3*t*alpha^2/4);

    for i = 1:1:5
        comboalphazero = nchoosek(calphazero, i);
        combozeroalpha = nchoosek(czeroalpha, i);
        comboalphaalpha = nchoosek(calphaalpha, i);
        l = size(comboalphazero, 1);
        for j = 1:1:l % Loop over all combinations
            tempalphazero = comboalphazero(j,:);
            tempzeroalpha = combozeroalpha(j,:);
            tempalphaalpha = comboalphaalpha(j,:);
            pzeroalpha{j, i+1} = pzeroalpha{1,1};
            palphazero{j, i+1} = palphazero{1,1};
            palphaalpha{j, i+1} = palphaalpha{1,1};
            for k = 1:1:i
                pzeroalpha{j, i+1} = pzeroalpha{j, i+1}*tempzeroalpha{k};
                palphazero{j, i+1} = palphazero{j, i+1}*tempalphazero{k};
                palphaalpha{j, i+1} = palphaalpha{j, i+1}*tempalphaalpha{k};
            end
        end
    end
end
```

Coarse-grained statistics

```
%% Coarse-graining of the fine-grained statistics to get the statistics with all the multi-click events together.
%
% Input:
%
% * pzeroalpha : Cell containing the probabilities corresponding to the different POVM elements conditioned on Alice having sent the zeroalpha state as outputted by ChannelSimulationStatistics.
%
% * palphazero : Cell containing the probabilities corresponding to the different POVM elements conditioned on Alice having sent the zeroalpha state as outputted by ChannelSimulationStatistics.
%
% * palphaalpha : Cell containing the probabilities corresponding to the different POVM elements conditioned on Alice having sent the alphaalpha state as outputted by ChannelSimulationStatistics.
%
% Output:
%
% * coursestats: An array of the coarse-grained statistics with coursestats(j,:) being the statistics for the $j^{th}$ state.

function coursestats = MultiChannelSimulationStatistics(pzeroalpha, palphazero, palphaalpha)


```

```
    coursestats = zeros(3, 7);
    finestats = cell(3,1);
    finestats{1} = pzeroalpha;
    finestats{2} = palphazero;
    finestats{3} = palphaalpha;

    for j = 1:1:3
        coursestats(j,1) = finestats{j}{1,1};               % No-click event
        probsum = finestats{j}{1,1};
        for i = 1:1:5
            coursestats(j,i+1) = finestats{j}{i,2};        % Single-click events
            probsum = probsum + coursestats(j,i+1);
        end
        coursestats(j,7) = 1 - probsum;             % Multi-click event
    end
end
```

C.3 Bound on the weight inside projected subspace

The code to find the weight inside the projected subspace for the three-state protocol as described in Appendix B.2 is given below.

```
%% Calculate the minimum weight of the <=N-photon subspace of the signal state of three-state protocol
%
% Input:
%
% * N   : number of photons
%
% * t   : fraction of photons going into the X-basis
%
% * pbx: Bob's outcome probability cell conditioned on Alice choice of state where x = 2 indicates her sending the ”+” state
%
```

```
% Output:
%
% * PleqN: probability of signal state in <=N-photon subspace

function PleqN = ThreeStateleqN(N, t, pbx)

    % Refer to CoarseGrainedPOVMS.m for the numbers for pbx

    data = 0; % prob of click in Z-basis or in middle time bin of minus detector
    minus = 0; % minus detector-only click prob

    data = pbx{4,2} + pbx{5,2} + pbx{2,2} + ...  % single clicks
           pbx{6,3} + pbx{7,3} + pbx{10,3} + ... % double clicks
           pbx{9,4};                             % triple clicks

    minus = pbx{1,2} + pbx{2,2} + pbx{3,2} + ... % single clicks
            pbx{1,3} + pbx{2,3} + pbx{5,3} + ... % double clicks
            pbx{1,4};                            % triple clicks

    % Cross-click prob
    cc = 1 - data - minus - pbx{1,1} + ...
         pbx{2,2};                          % to avoid double-counting
    if cc < 0    % cc maybe slightly negative e.g. -1e-16 due to numerical precision
        cc = 0;
    end

    cc_min_0 = 0;
    cc_min_Nplus1 = 1 - t^(N+1)-(1-t/4)^(N+1) + (3*t/4)^(N+1);

    PleqN = 1 - (cc - cc_min_0)/(cc_min_Nplus1 - cc_min_0);

    if PleqN > 1
        PleqN = 1;
    elseif PleqN < 0
        PleqN = 0;
    end
end
```

C.4 Approximate diagonalisation

The code that gives the approximate eigenvalue and eigenvector along with the maximum deviation of the eigenvector as discussed in Section 3.5.

```

%% This function computes the maximum distance between the eigenvectors of two matrices that are close in 1-norm.
%
% Input :
%
% * rho - The known finite dimensional state.
%
% * ep - The maximum distance between the known and unknown state in 1-norm.
%
% * n - Ordering the eigenvalues from largest to smallest, the eigenvector corresponding to the nth eigenvalue is the one whose distance we wish to find. For eg.- n=1 means to find the max dist between the first eigenvector of rho to the corresponding eigenvector in the matrix at most ep away from rho (in 1-norm).
%
% Output :
%
% * vec - The eigenvector corresponding to the nth eigenvalue of rho.
%
% * eigvalue - The nth eigenvalue of rho.
%
% * dist - The maxumum 1-norm of the difference of the two eigenvectors.
%

```

```
function [vec,eigvalue,dist] = closestEigenvectors(rho, ep, n)

    [V,D] = eig(rho);
    [~,ind] = sort(diag(D),'descend');
    D = D(ind,ind);
    V = V(:,ind);
    e = diag(D);
    vec = V(:,n);
    eigvalue = e(n);
    if n == 1
        delta = e(1)-e(2)-ep;
    elseif n == length(rho)
        delta = e(n-1)-e(n)-ep;
    else
        delta = min(e(n-1)-e(n)-ep, e(n)-e(n+1)-ep);
    end

    if delta<0 || ep>delta
        dist = 2;
    else
        Fsquare = 1-ep^2/delta^2;
        dist = 2*sqrt(1-sqrt(Fsquare));
    end


end
```

C.5 Generalised decoy-state method

The code that implements the generalised decoy-state method is given here.

```
%% FUNCTION NAME: decoyBoundsSDP
% This file contains the function to compute decoy bounds without assuming that all the states are diagonal by solving an SDP.
%
% Input:
%
```

```
% * ConstraintPOVM - Cell containing the constraining POVM
%
% * ObjPOVM - Cell containing the objective POVM
%
% * rho - Cell which contains the projected states. So rho{i} = $\Pi_M\rho_i\Pi_M$ the $i^{th}$ state.
%
% * sigma - Cell which contains the objective states. Note that these are assumed to lie within the projected space.
%
% * decoy_expectations - Matrix with all measurement outcomes for different intensities. decoy_expectations(i,j) is the detection statistics of the $i^{th}$ measurement for $\rho_j$.
%
% * Wb - The weight outside the projected subspace where the projection is made on the state Bob receives (dimension reduction on the POVMs)
%
% * C - Signifies whether or not the states commute with the projection. C = 1 indicates that they do, otherwise they do not.
%
% Output:
%
% * YL - Lower bounds on statistics
%
% * YU - Upper bounds on statistics

function [YL, YU] = decoyBoundsSDP(ConstraintPOVM, ObjPOVM, rho, sigma, decoy_expectations, Wb, C)

    linearconstrainttolerance = 1e-10;

    dimM = length(ConstraintPOVM{1});
    dimS = length(rho{1});
    dim = dimM*dimS;
    for objStateIndex = 1:numel(sigma)
        for objMeasurementIndex =  1:1:numel(ObjPOVM)
```

```
            cvx_begin sdp quiet
                cvx_precision default
                cvx_solver Mosek
                variable J(dim,dim) hermitian semidefinite
                minimize real(trace(kron(ObjPOVM{i},sigma{k})*J))
                for consStateIndex = 1:numel(rho)
                   W = 1-trace(rho{consStateIndex});
                    if W<= eps
                       W = 0;
                    end
                    epsilon = 2*sqrt(W);
                    if C == 1 % Commuting case
                        epsilon = W;
                    end
                    for consMeasurementIndex = 1:numel(
                       ConstraintPOVM)
                        if C ==1
                            real(trace((kron(ConstraintPOVM{
                               consMeasurementIndex},rho{
                               consStateIndex})*J)))...
                                <= min(decoy_expectations(
                                   consMeasurementIndex,
                                   consStateIndex)+
                                   linearconstrainttolerance,1);
                        else
                            real(trace((kron(ConstraintPOVM{
                               consMeasurementIndex},rho{
                               consStateIndex})*J)))...
                                <= min(decoy_expectations(
                                   consMeasurementIndex,
                                   consStateIndex)+epsilon+
                                   linearconstrainttolerance,1);
                        end
                        real(trace((kron(ConstraintPOVM{
                           consMeasurementIndex},rho{
                           consStateIndex})*J)))...
                            >= max(decoy_expectations(
                               consMeasurementIndex,
                            consStateIndex)-2*epsilon-Wb(consStateIndex)-linearconstrainttolerance,0);
                end
            end
            1-W-Wb(consStateIndex)-epsilon <= real(trace(kron(eye(dimM),rho{consStateIndex})*J)) <= 1;
            if Wb == 0
                norm(PartialTrace(J,1,[dimM,dimS])-eye(dimS)) <= 0;
            else
                PartialTrace(J,1,[dimM,dimS]) <= eye(dimS);
            end
        cvx_end
        YL(objMeasurementIndex,objStateIndex) = real(trace(kron(ObjPOVM{objMeasurementIndex},sigma{objStateIndex})*J));

        cvx_begin sdp quiet
            cvx_precision default
            cvx_solver Mosek
            variable J(dim,dim) hermitian semidefinite
            maximize real(trace(kron((ObjPOVM{i}),sigma{k})*J))
            for consStateIndex = 1:numel(rho)
                W = 1-trace(rho{consStateIndex});
                if W<= eps
                    W = 0;
                end
%                   epsilon = 2*sqrt(2*W-W^2);
                epsilon = 2*sqrt(W);
                if C == 1
                    epsilon = W;
                end
                for consMeasurementIndex = 1:numel(ConstraintPOVM)
                    if C ==1
```

```
                            real(trace((kron(ConstraintPOVM{
                               consMeasurementIndex},rho{
                               consStateIndex})*J))) <= min(
                               decoy_expectations(
                               consMeasurementIndex,
                               consStateIndex)+
                               linearconstrainttolerance,1);
                        else
                            real(trace((kron(ConstraintPOVM{
                               consMeasurementIndex},rho{
                               consStateIndex})*J))) <= min(
                               decoy_expectations(
                               consMeasurementIndex,
                               consStateIndex)+epsilon+
                               linearconstrainttolerance,1);
                        end
                        real(trace((kron(ConstraintPOVM{
                           consMeasurementIndex},rho{
                           consStateIndex})*J))) >= max(
                           decoy_expectations(
                           consMeasurementIndex,consStateIndex)
                           -2*epsilon-Wb(consStateIndex)-
                           linearconstrainttolerance,0);
                    end
                end
                1-W-Wb(consStateIndex)-epsilon <= real(trace(kron
                   (eye(dimM),rho{consStateIndex})*J)) <= 1;
                if Wb == 0
                    norm(PartialTrace(J,1,[dimM,dimS])-eye(dimS))
                        <= 0;
                else
                    PartialTrace(J,1,[dimM,dimS]) <= eye(dimS)
                end
            cvx_end
            YU(objMeasurementIndex,objStateIndex) = real(trace(
               kron((ObjPOVM{objMeasurementIndex}),sigma{
               objStateIndex})*J));
        end
```

```
    end

end
```