Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Física

Distribución cuántica de claves con ruido realista

Autora: María Fadrique Gutiérrez
Tutores: Mateus Araújo, Luis Miguel Nieto Calzada
Año 2024

—¿Has encontrado la solución a la adivinanza? —preguntó el Sombrerero, dirigiéndose de nuevo a Alicia.
—No. Me doy por vencida. ¿Cuál es la solución?
—No tengo la menor idea —dijo el Sombrerero.
—Ni yo —dijo la Liebre de Marzo.

LEWIS CARROLL, *Alicia en el País de las Maravillas*

A Javi, por ayudarme a mantenerme lúcida durante estos años.

A Bosona, Fey y Brujita, porque todos los trabajos necesitan, al menos, tres e-ratas.

Índice general

Resumen

Con el auge de la computación cuántica, surge la necesidad de desarrollar métodos más seguros para la distribución de claves, más allá de los enfoques tradicionales, mediante la distribución cuántica de claves (QKD). Este trabajo aborda el cálculo de la tasa de clave en protocolos de QKD, un desafío con una larga trayectoria. Los métodos analíticos están limitados a protocolos con bases de medición simétricas, mientras que los métodos numéricos utilizan la min-entropía, que ofrece un límite inferior débil para la entropía de von Neumann.

Se ha implementado en el lenguaje de programación Julia un algoritmo basado en programación semidefinida (SDP), que permite un cálculo numérico más eficiente de la tasa de clave. Además, se analiza el impacto del ruido, como las interferencias ambientales y las imperfecciones de los detectores, sobre el rendimiento del protocolo, utilizando dos modelos de ruido comunes en las fuentes de entrelazamiento: los intervalos de tiempo y los modos espaciales. Este enfoque permite una evaluación más precisa y práctica del rendimiento de los protocolos de QKD en condiciones reales.

Abstract

With the advent of quantum computing, there is an increasing need to develop more secure methods for key distribution, moving beyond traditional approaches through quantum key distribution (QKD). This study focuses on the calculation of key rates in QKD protocols, a longstanding challenge. Analytical methods are restricted to protocols using symmetric measurement bases, while numerical methods rely on min-entropy, which offers a loose lower bound to the von Neumann entropy.

A more efficient numerical calculation of the key rate has been achieved through the implementation of an algorithm in Julia, based on semidefinite programming (SDP). Furthermore, the impact of noise on protocol performance, such as environmental interference and detector imperfections, is analyzed using two common noise models for entanglement sources: time bins and spatial modes. This approach enables a more accurate and practical assessment of QKD protocol performance in real-world conditions.

Introducción

¿Cómo protegeremos nuestros datos más sensibles en un futuro donde la computación cuántica redefine los límites de la seguridad?

A lo largo de la historia, la criptografía ha sido la guardiana de los secretos más preciados de las civilizaciones, desde las técnicas de cifrado del Antiguo Egipto hasta los sofisticados métodos digitales de hoy. Uno de los hitos más significativos en este viaje fue la invención del *One-Time-Pad* [1] por Vernam en 1917, un método que ofrece una seguridad perfecta si se utiliza adecuadamente. En este sistema, cada bit del mensaje se combina con un bit de una clave aleatoria mediante una operación XOR (suma en módulo dos), produciendo un texto cifrado imposible de descifrar sin la clave. Sin embargo, el verdadero desafío ha sido siempre cómo distribuir estas claves de forma segura, que deben ser tan largas como el mensaje y solo pueden ser utilizadas una vez.

En la criptografía convencional, dos partes distantes, tradicionalmente llamadas Alice y Bob, buscan comunicarse de manera privada mientras Eva, una espía, trata de interceptar su conversación. A lo largo de los siglos, los criptógrafos han desarrollado innumerables métodos para frustrar a los criptoanalistas, pero cada avance en seguridad ha sido seguido por un nuevo método de ataque. La seguridad computacional proporcionada por la criptografía clásica siempre está en riesgo, ya que en la física clásica nada impide a Eva copiar la clave durante su distribución.

Con la llegada de la computación cuántica, la situación se vuelve aún más apremiante. Si se construyera una computadora cuántica a gran escala, gran parte de la criptografía convencional[1] podría volverse obsoleta, ya que estos dispositivos tienen la capacidad teórica de romper muchos de los algoritmos actuales en un tiempo significativamente reducido.

Ante este desafío, la criptografía cuántica [2] entra en escena. Utilizando principios fundamentales de la naturaleza, la distribución cuántica de claves, o QKD por sus siglas en inglés (*Quantum Key Distribution*), permite compartir una clave protegida contra cualquier intento de interceptación por parte de Eva. La idea, propuesta por primera vez por Wiesner en la década de 1970 [3] y desarrollada en los años 80 por Bennett y Brassard, ha evolucionado significativamente y, aunque su implementación ideal presenta desafíos, su potencial es enorme.

La distribución cuántica de claves se basa en principios fundamentales de la física cuántica. En primer lugar, *toda medición perturba el sistema*. Es decir, cualquier intento por parte de Eva de obtener información introduce perturbaciones que delatan su presencia. Además, no es posible obtener una copia perfecta del estado compartido, pues el *teorema de no clonación* [4] lo prohíbe. Finalmente, las correlaciones cuánticas obtenidas de mediciones separadas en pares entrelazados *violan las*

[1] RSA, uno de los sistemas de cifrado más importantes del planeta, se basa en la supuesta dificultad de factorizar números enteros grandes en sus factores primos. El algoritmo de Shor [5] podría resolver esto rápidamente si se ejecutara en una computadora cuántica lo suficientemente avanzada.

desigualdades de Bell, lo que significa que los resultados no pueden haber sido preestablecidos, haciendo imposible que un espía conozca los resultados de antemano [6].

Una parte esencial en el análisis de la seguridad de los protocolos de QKD es la **tasa de clave**. Esta tasa representa la cantidad de bits de clave generados de manera segura por unidad de tiempo, teniendo en cuenta imperfecciones prácticas como el ruido y la corrección de errores.

Tradicionalmente, para protocolos con bases de medida altamente simétricas, como BB84 1.1, las tasas de clave secreta se han calculado de manera analítica. Sin embargo, cuando se consideran protocolos con bases de medida arbitrarias, donde las simetrías son menos evidentes o inexistentes, estos cálculos se vuelven más complejos. En estos casos, el enfoque más común ha sido utilizar la min-entropía [7] para establecer un límite inferior en la entropía de von Neumann, lo cual proporciona un límite inferior en la tasa de clave secreta. Aunque este método es efectivo, tiende a proporcionar tasas de clave más bajas de lo que realmente se podría alcanzar.

Recientemente, se han desarrollado métodos numéricos más avanzados para calcular tasas de clave más precisas. Un enfoque notable es el presentado por Brown et al. [8], que utiliza una jerarquía de *programación semidefinida (SDP)* para obtener límites inferiores convergentes en la entropía de von Neumann condicional. Esta jerarquía se basa, a su vez, en la jerarquía NPA [9], la cual se utiliza para caracterizar correlaciones cuánticas. Sin embargo, esta técnica conlleva una alta complejidad computacional, lo que limita su aplicación a sistemas con pocas configuraciones de medición.

En este trabajo, empleamos una adaptación del método de Brown et al., desarrollada por Mateus Araújo et al. [10]. Mientras que el método de Brown et al. se aplica a QKD independiente de dispositivos, donde no se asume ninguna seguridad específica sobre los dispositivos utilizados, nuestra adaptación se enfoca en QKD con dispositivos caracterizados. En este contexto, se tiene un conocimiento preciso del comportamiento de los dispositivos. Por lo tanto, no es necesario utilizar la jerarquía NPA, lo que reduce significativamente la complejidad computacional. En nuestra adaptación, dicha complejidad es prácticamente independiente del número de configuraciones de medición y depende principalmente de la dimensión del estado cuántico.

El objetivo de este trabajo es implementar este algoritmo en el lenguaje de programación *Julia* e integrar modelos de ruido realistas para evaluar la tasa de generación de claves en QKD bajo condiciones prácticas.

Este documento está organizado de la siguiente manera:
En esta primera sección, se introducen las herramientas matemáticas necesarias para la formulación y desarrollo del tema principal. El capítulo 1 profundiza en QKD, comenzando con una explicación del protocolo BB84, el más reconocido, y presentando el protocolo utilizado en este trabajo, junto con el cálculo teórico de la tasa de clave. En el capítulo 2, se describen los dos modelos de ruido realistas considerados en el estudio. El capítulo 3 se dedica a las herramientas y métodos empleados en el desarrollo del algoritmo, como SDP, Julia y JuMP, detallando además el algoritmo en su totalidad. Finalmente, en el capítulo 4, se presentan y analizan los resultados obtenidos, comparándolos con los del método de min-entropía, el cual se describe brevemente en un anexo.

0.1. Definiciones previas

En esta sección se presentan de manera breve herramientas teóricas fundamentales de la mecánica cuántica y la teoría de la información cuántica [11, 12], que son esenciales para el desarrollo de los

temas abordados en el capítulo 1.

0.1.1. Operador densidad

El operador densidad, también conocido como matriz densidad, es una herramienta fundamental para describir sistemas cuánticos, particularmente cuando se encuentran en estados mixtos. Estos estados reflejan la incertidumbre sobre el estado exacto del sistema cuando no se tiene información completa o cuando el sistema interactúa con su entorno. Formalmente, el operador densidad para un estado mixto se define como:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|, \tag{1}$$

donde p_i son las probabilidades asociadas a los estados puros $|\psi_i\rangle$, que forman la mezcla estadística. Este operador cumple varias propiedades importantes: es un operador hermítico ($\rho = \rho^\dagger$), tiene traza unitaria ($\mathrm{Tr}(\rho) = 1$), y es semidefinido positivo (todos sus autovalores son no negativos).

0.1.2. Entrelazamiento cuántico

El entrelazamiento cuántico es uno de los fenómenos más fascinantes de la mecánica cuántica y desempeña un papel esencial en la teoría de la información cuántica.

Si dos sistemas cuánticos, inicialmente aislados, están en los estados $|\psi_A\rangle$ y $|\psi_B\rangle$, el estado conjunto de ambos puede describirse mediante el producto tensorial $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$. Un estado producto indica que los sistemas no están correlacionados cuánticamente, es decir, la medición de uno de los sistemas no afecta al otro.

Sin embargo, si los sistemas interactúan, el estado conjunto puede evolucionar a un estado no factorizable. En tal caso, el sistema completo se encuentra en un estado *entrelazado*, que no se puede expresar como un producto tensorial de los subsistemas. En cambio, este tipo de estados se describen como una combinación lineal de productos tensoriales, reflejando las correlaciones cuánticas entre los subsistemas.

Un ejemplo representativo de entrelazamiento es el estado de Bell $|\phi^+\rangle$, que para dos partículas se expresa como:

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{2}$$

En este estado, si se realiza una medición sobre una de las partículas y el resultado es $|0\rangle$, la otra partícula colapsará instantáneamente al mismo estado $|0\rangle$. De igual forma, si se mide $|1\rangle$ en una de las partículas, la otra también colapsará al estado $|1\rangle$, independientemente de la distancia entre ellas. Este fenómeno, conocido como no-localidad, desafía nuestra comprensión clásica de la información y de la causalidad.

En la práctica, el entrelazamiento se ve afectado por el ruido (1.3), lo cual será crucial en nuestro protocolo y se explorará más a fondo en el capítulo 2.

0.1.3. Entropía de Shannon

En la teoría de la información, la entropía de Shannon de una variable aleatoria A se define como:

$$H(A) = -\sum_{a \in A} p(a) \log_2 p(a), \tag{3}$$

donde A es el conjunto de todos los posibles valores que puede tomar a, y $p(a)$ es la probabilidad de que A tome el valor a. La entropía de Shannon mide la cantidad promedio de información que se necesita para especificar el valor de A. En otras palabras, representa la incertidumbre asociada con la variable aleatoria. Este concepto se basa en el teorema de codificación de Shannon, que afirma que una secuencia de n muestras independientes de A puede ser comprimida a $n \cdot H(A)$ bits sin pérdida de información, con alta probabilidad a medida que n crece.

Entropía binaria

Un caso particular de la entropía de Shannon es la entropía binaria. Esta se aplica a una variable aleatoria que solo puede tomar dos valores, con probabilidades p y $1 - p$, y se define como:

$$h(p) = -p \log_2 p - (1 - p) \log_2 (1 - p). \tag{4}$$

Entropía condicional

Un concepto fundamental es la entropía condicional de una variable aleatoria A dado que conocemos el valor de otra variable aleatoria B. Se define como:

$$H(A|B) = -\sum_{a,b \in A,B} p(a,b) \log_2 \frac{p(a,b)}{p(b)}, \tag{5}$$

donde $p(a,b)$ es la probabilidad conjunta de que A tome el valor a y B tome el valor b, y $p(b)$ es la probabilidad marginal de que B tome el valor b. Intuitivamente, la entropía condicional $H(A|B)$ mide la cantidad de información que contiene la variable A dado que ya conocemos B. Esta entropía es cero si A y B están perfectamente correlacionadas, y es igual a $H(A)$ si A y B son independientes. Cabe destacar que la entropía condicional no es simétrica, es decir, en general $H(A|B)$ no es igual a $H(B|A)$.

0.1.4. Entropía de von Neumann

En el marco de la teoría de la información cuántica, la entropía de von Neumann generaliza la entropía de Shannon al ámbito cuántico. Para un sistema cuántico descrito por una matriz densidad ρ en un espacio de Hilbert $\mathcal{H}$, la entropía de von Neumann se define como:

$$S(\rho) = -\mathrm{Tr}(\rho \log_2 \rho), \tag{6}$$

donde Tr denota la traza del operador.

Entropía condicional cuántica

La entropía condicional cuántica se define como:

$$S(A|B) = S(\rho_{AB}) - S(\rho_B), \tag{7}$$

donde ρ_{AB} es la matriz densidad del sistema conjunto A y B, y ρ_B es la matriz densidad reducida del subsistema B.

Esta entropía puede expresarse en términos de la **entropía relativa cuántica** (o divergencia cuántica), que es una medida de la distinguibilidad entre dos operadores cuánticos. La entropía relativa cuántica entre dos operadores semidefinidos positivos ρ y σ se define como:

$$D(\rho\|\sigma) = \mathrm{Tr}[\rho(\log_2 \rho - \log_2 \sigma)], \tag{8}$$

siempre que el soporte de ρ esté contenido en el soporte de σ ($\mathrm{supp}(\rho) \subseteq \mathrm{supp}(\sigma)$). Esto significa que ρ debe tener componentes solo en los subespacios donde σ también tiene componentes. Si ρ tiene componentes fuera del soporte de σ, la divergencia cuántica se considera infinita, ya que no se puede comparar directamente ρ con σ en esas regiones.

Para un estado bipartito ρ_{AB} en un espacio de Hilbert $\mathcal{H}_A \otimes \mathcal{H}_B$, la entropía condicional de von Neumann se puede expresar en términos de la divergencia cuántica como:

$$S(A|B) = -D(\rho_{AB}\|\mathbb{I}_A \otimes \rho_B), \tag{9}$$

donde $\mathbb{I}_A$ es el operador identidad en el espacio de Hilbert $\mathcal{H}_A$.

Capítulo 1

Protocolos de QKD y fundamentos teóricos para el cálculo de la tasa de clave

La distribución cuántica de claves es la aplicación más destacada de la criptografía cuántica. Desde su invención, se han desarrollado diversos protocolos para mejorar su seguridad y eficiencia. En este capítulo, tomamos el pionero protocolo BB84 como punto de partida, para presentar a continuación el que hemos utilizado en nuestro trabajo. Además, mostramos las definiciones teóricas en las que se basa el cálculo de la tasa de clave.

1.1. Protocolo BB84

El protocolo BB84, propuesto por Bennett y Brassard en 1984 [13], es el primero y uno de los más relevantes debido a su simplicidad. A continuación, se describe su procedimiento [14] (también mostrado en la tabla 1.1).

- Alice y Bob acuerdan un número fijo $0 < p < \frac{1}{2}$, típicamente cercano a 0. Alice prepara una secuencia de fotones individuales, seleccionando entre uno de los cuatro estados de polarización: horizontal $|0\rangle$, vertical $|1\rangle$, 45 grados $|+\rangle$ y -45 grados $|-\rangle$. Estos estados corresponden a dos bases: la base rectilínea o base $\mathbb{Z}$ ($|0\rangle$ y $|1\rangle$), y la base diagonal o base $\mathbb{X}$ ($|+\rangle$ y $|-\rangle$). Alice elige la base $\mathbb{X}$ con una probabilidad p y la base $\mathbb{Z}$ con una probabilidad $1-p$. Vemos que cada uno de los bits 0 y 1 se codifica en dos formas distintas, específicamente en estados no ortogonales, debido a que

$$|\pm\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle \pm |1\rangle \right). \tag{1.1}$$

  Esto implica que hay indeterminación en la medición si se mide en una base diferente de la base en la que el estado fue preparado.

- Alice envía los fotones a Bob a través de un canal cuántico inseguro que podría ser interceptado por Eva. Bob mide la polarización de cada fotón, eligiendo la base $\mathbb{X}$ con una probabilidad p y la base $\mathbb{Z}$ con una probabilidad $1 - p$. Bob registra tanto la base utilizada como el resultado de cada medición.

- Alice y Bob anuncian a través de un canal clásico las bases que usaron para cada medición, sin

revelar los resultados. Este canal no tiene que ser confidencial pero sí estar autenticado. Eva puede escuchar toda la comunicación pero no puede modificarla ni suplantar la identidad de Alice y Bob.

- Organizan sus datos de polarización en cuatro grupos según las bases utilizadas: $\mathbb{ZZ}$ (ambos miden en $\mathbb{Z}$), $\mathbb{XX}$ (ambos en $\mathbb{X}$), $\mathbb{ZX}$ (Alice en $\mathbb{Z}$ y Bob en $\mathbb{X}$) y $\mathbb{XZ}$ (Alice en $\mathbb{X}$ y Bob en $\mathbb{Z}$). A continuación, descartan los dos grupos donde usaron bases diferentes. Los bits restantes forman la *clave filtrada*.

- Toman los datos del grupo $\mathbb{XX}$ como *rondas de prueba*. Estas rondas consisten en revelar públicamente las polarizaciones de todos los m fotones pertenecientes a este grupo y compararlas. A partir del número de discrepancias r que encuentran, calculan la tasa de error como $e = \frac{r}{m}$. Si esta tasa es menor que un valor umbral preestablecido, pueden continuar con el siguiente paso del protocolo. De lo contrario, deben descartar los datos y reiniciar el protocolo para garantizar la seguridad de la clave.

- La clave se genera exclusivamente en la base $\mathbb{Z}$, ya que Alice y Bob eligen esta base con mayor probabilidad. Por lo tanto, los datos pertenecientes al grupo $\mathbb{ZZ}$ (*rondas de clave*) constituyen la *clave bruta*. Esta clave se somete a un procesamiento clásico que incluye dos etapas:
  Reconciliación de información. Alice y Bob comparan una parte de sus claves brutas a través del canal clásico para detectar y corregir discrepancias, asegurándose así de que ambas claves sean idénticas.
  Amplificación de privacidad. Sacrifican parte de la clave restante para eliminar cualquier información que Eva pueda haber obtenido durante la transmisión o en el proceso de reconciliación de la información.

Los bits restantes forman la *clave secreta*, que está lista para ser utilizada en la comunicación de manera segura.

Secuencia de bits de Alice	0	0	1	1	1	0	1	0	0	1
Bases de Alice	Z	X	Z	Z	X	Z	X	Z	Z	X
Polarización de los fotones de Alice	→	↗	↑	↑	↖	→	↖	→	→	↖
Bases de Bob	Z	X	X	Z	X	Z	Z	Z	Z	X
Resultados de Bob	→	↗	↖	↑	↖	→	→	↑	→	↗
Clave filtrada de Alice	0	0		1	1	0		0	0	1
Clave filtrada de Bob	0	0		1	1	0		1	0	0
Rondas de prueba		✓			✓					✗
Clave bruta de Alice	0			1		0		0	0	
Clave bruta de Bob	0			1		0		1	0	

Tabla 1.1: Procedimiento simplificado del protocolo BB84. Las columnas coloreadas en lila indican los casos en los que Alice y Bob midieron en la misma base y obtuvieron resultados coincidentes. Las columnas coloreadas en rojo muestran los casos en los que Bob obtuvo un resultado diferente al de Alice a pesar de haber medido en la misma base, lo que indica la presencia de un espía o algún tipo de ruido.

1.2. Protocolo MUB

Como acabamos de ver, el protocolo BB84 se limita a sistemas de dos niveles (qubits) y utiliza solo dos bases de medición. Una extensión de este concepto es el protocolo de seis estados [15], que incorpora una tercera base, la base $\mathbb{Y}$. Sin embargo, es posible ir más allá y explorar las posibilidades de la QKD en dimensiones superiores utilizando sistemas cuánticos de dimensión d, denominados *qudits* [16]. En estos sistemas, podemos emplear conjuntos completos de bases mutuamente no sesgadas, conocidas como MUB (*mutually unbiased bases*), lo que permite transmitir más bits de información por señal ($\log_2 d > 1$ bit) y mejora la resistencia al ruido.

1.2.1. MUB

Un par de bases ortonormales $\{|\psi_1\rangle, \ldots, |\psi_d\rangle\}$ y $\{|\phi_1\rangle, \ldots, |\phi_d\rangle\}$ en el espacio de Hilbert $\mathbb{C}^d$ se dice que son mutuamente no sesgadas si, y solo si, el cuadrado de la magnitud del producto interno entre cualquier par de estados base $|\psi_i\rangle$ y $|\phi_j\rangle$ es igual al inverso de la dimensión d:

$$|\langle\psi_i|\phi_j\rangle|^2 = \frac{1}{d}, \quad \forall i, j \in \{1, \ldots, d\}. \tag{1.2}$$

Estas bases se consideran no sesgadas en el siguiente sentido: si un sistema se prepara en un estado perteneciente a una de las bases, entonces se predice que todos los resultados de la medición con respecto a la otra base ocurrirán con igual probabilidad. En otras palabras, al medir en la otra base, obtienes la menor cantidad de información posible sobre el estado original.

Cuando d es una potencia de un número primo, existen exactamente $d + 1$ MUB. Para $d = 2$, las bases $\mathbb{Z}$, $\mathbb{X}$ y $\mathbb{Y}$ proporcionan el ejemplo más sencillo de MUB.

1.2.2. Especificaciones teóricas

En lugar de trabajar con fotones individuales, empleamos pares de fotones entrelazados, lo cual simplifica el análisis matemático. Esto implica que existe una fuente externa que emite estos pares de fotones, donde Alice recibe uno y Bob el otro. Aunque inicialmente pueda parecer diferente al escenario donde Alice prepara y envía los fotones a Bob, en realidad es equivalente. Si redefinimos a Alice como la combinación de Alice y la fuente, obtenemos el mismo esquema de *preparar y medir* que en el protocolo BB84 descrito anteriormente.

Definimos el estado que comparten Alice y Bob como un *estado isótropo*, el cual representa un estado entrelazado afectado por ruido. Esto permite simular las tasas de error que se medirían experimentalmente en un escenario real. El estado isótropo se describe mediante el parámetro de *visibilidad* v, que mide la proporción de entrelazamiento frente al ruido, y se expresa como:

$$\rho_{\text{iso}}(v) = v|\phi^+\rangle\langle\phi^+| + (1 - v)\frac{\mathbb{I}}{d^2}, \tag{1.3}$$

donde $|\phi^+\rangle = \frac{1}{\sqrt{d}} \sum_{i=0}^{d-1} |ii\rangle$ es un estado de Bell generalizado y $v \in [0, 1]$. Cuando $v = 1$, tenemos un estado máximamente entrelazado y puro, mientras que para $v = 0$, el estado es máximamente mezclado, representando el caso de máximo ruido. Profundizaremos en este parámetro en el capítulo 2.

Nos basamos en la referencia [16], donde Alice y Bob realizan sus mediciones en un conjunto completo de MUB en un espacio de dimensión d, siendo d un número primo, y donde las bases de Bob son las transpuestas de las de Alice.

En nuestro caso, permitimos que la dimensión d sea cualquier número entero. Cuando d es una potencia de un número primo, se usa la construcción de Wootters-Fields [17] para generar las MUB. Sin embargo, para otras dimensiones, se cree ampliamente que no existe un conjunto completo de $d+1$ MUB [18]. En su lugar, generamos numéricamente bases que son solo aproximadamente no sesgadas mediante un método de descenso de gradiente[1]. Obtenemos un conjunto de n MUB aproximadas minimizando la siguiente expresión [19]:

$$\min_{\{U^{(r)}\}_r} \frac{1}{(d-1)\binom{n}{2}} \sum_{k<l} \sum_{i,j=0}^{d-1} \left(\left| (U^{(k)\dagger} U^{(l)})_{i,j} \right|^2 - \frac{1}{d} \right)^2, \tag{1.4}$$

donde $\{U^{(r)}\}_{r=1}^{n}$ es una colección de matrices unitarias. Esta expresión mide la desviación de las matrices unitarias de formar un conjunto de MUB, por lo que el valor de esta función es cero para un conjunto de MUB exactas.

1.3. Tasa de Devetak-Winter

La eficacia de los protocolos de QKD se mide por la tasa de clave, que es la cantidad de bits de clave secreta que se pueden obtener por cada ronda de clave. Cuando el número de estados preparados tiende a infinito, esta tasa se conoce como tasa de clave asintótica. Esta cantidad está acotada inferiormente por la *tasa de Devetak-Winter*, que se calcula asumiendo que la reconciliación de información y la amplificación de privacidad se realizan de la mejor manera posible.

La tasa de Devetak-Winter viene dada por la expresión [20]:

$$K \geq S(A|E) - H(A|B), \tag{1.5}$$

donde K es la tasa de clave, $S(A|E)$ es la entropía condicional cuántica del sistema de Alice (A) dado que se tiene información sobre el sistema de Eva (E), y $H(A|B)$ es la entropía condicional clásica del sistema de Alice dado que se tiene información sobre el sistema de Bob (B).

Es útil distinguir entre la tasa por ronda de clave, que acabamos de definir, y la tasa por ronda total, que considera todos los tipos de ronda ($\mathbb{ZZ}$, $\mathbb{XX}$, $\mathbb{ZX}$ y $\mathbb{XZ}$). Para calcular esta última, multiplicamos la tasa de Devetak-Winter por la probabilidad de que se produzca una ronda de clave, es decir, que tanto Alice como Bob midan en la base $\mathbb{Z}$. Esta probabilidad está dada por $(1-p)^2$, conocida como *factor de sifting* (factor de filtrado). En nuestro caso, este factor puede aproximarse a 1, ya que $p \approx 0$. De modo que ambas tasas son equivalentes.

1.3.1. $H(A|B)$

En la expresión (1.5), $H(A|B)$ depende únicamente de las estadísticas medidas por Alice y Bob en la base en la que se genera clave, que es la base computacional (base $\mathbb{Z}$).

[1] El método de descenso de gradiente es un proceso iterativo usado para encontrar el mínimo de una función. A partir de un punto inicial, se calcula la dirección en la que la función aumenta más rápidamente, es decir, el gradiente, y se avanza en la dirección opuesta. Este proceso se repite hasta que los cambios sean lo suficientemente pequeños, indicando que se ha alcanzado una proximidad al mínimo de la función.

Para calcular $H(A|B)$, utilizamos la definición clásica de la entropía condicional (5), definiendo las probabilidades de la siguiente manera:

$$p(a,b) = \mathrm{Tr}(\rho_{\text{iso}} \cdot \Pi_a \otimes \Pi_b), \tag{1.6}$$

donde Π_a y Π_b son los operadores de proyección correspondientes a los resultados a y b de las mediciones realizadas por Alice y Bob, respectivamente, y ρ_{iso} es el estado isótropo (1.3).

- Cuando $a = b$

$$p(a,b) = \frac{v}{d} + \frac{1-v}{d^2}. \tag{1.7}$$

- Cuando $a \neq b$

$$p(a,b) = \frac{1-v}{d^2}. \tag{1.8}$$

La probabilidad marginal de que Bob obtenga el resultado b es uniforme:

$$p(b) = \frac{1}{d}. \tag{1.9}$$

Con estas probabilidades, la entropía condicional $H(A|B)$ resulta en:

$$H(A|B) = -d\left(\frac{v}{d} + \frac{1-v}{d^2}\right)\log_2\left(\frac{\frac{v}{d} + \frac{(1-v)}{d^2}}{\frac{1}{d}}\right) - d(d-1)\frac{(1-v)}{d^2}\log_2\left(\frac{\frac{(1-v)}{d^2}}{\frac{1}{d}}\right). \tag{1.10}$$

Esta expresión puede simplificarse a:

$$H(A|B) = h\left(v + \frac{1-v}{d}\right) + \left(1 - v - \frac{1-v}{d}\right)\log_2(d-1), \tag{1.11}$$

donde $h(\cdot)$ es la entropía binaria (4).

1.3.2. $S(A|E)$

El problema realmente interesante es $S(A|E)$, ya que debe ser minimizado sobre todos los estados ρ_{ABE} compartidos por Alice, Bob y Eva, que sean compatibles con las estadísticas medidas por Alice y Bob. Esto significa que debemos considerar el peor de los casos, es decir, la situación en la que Eva tiene la mayor cantidad de información posible sobre la clave compartida, sin ser detectada. Para hacer esto, tomamos la definición de la entropía condicional en términos de la entropía relativa (9):

$$S(A|E) = -D(\rho_{\tilde{A}E} \| \mathbb{I}_A \otimes \rho_E) = -\mathrm{Tr}\left[\rho_{\tilde{A}E}\left(\log_2 \rho_{\tilde{A}E} - \log_2(\mathbb{I}_A \otimes \rho_E)\right)\right], \tag{1.12}$$

donde $\rho_{\tilde{A}E}$ es el estado clásico-cuántico dado por:

$$\rho_{\tilde{A}E} = \sum_a |a\rangle\langle a| \otimes \rho_E(a), \tag{1.13}$$

donde $\rho_E(a) = \mathrm{Tr}_{AB}[(A_0^a \otimes \mathbb{I}_{BE})\rho_{ABE}]$ y $\{A_0^a\}_{a=0}^{d-1}$ es la base utilizada por Alice para generar la clave secreta, es decir, la base $\mathbb{Z}$, siendo $A_0^a = |a\rangle\langle a|$.

Cuadratura de Gauss-Radau

Para obtener un límite inferior de la entropía relativa cuántica, empleamos la cuadratura de Gauss-Radau, que proporciona una aproximación eficiente para representaciones integrales de funciones complicadas como el logaritmo.

El logaritmo en base 2 de una función x, $\log_2(x)$, se puede expresar en función del logaritmo natural $\ln(x)$ como:

$$\log_2(x) = \frac{\ln(x)}{\ln(2)}. \tag{1.14}$$

Una forma común de representar $\ln(x)$ es mediante la integral:

$$\ln(x) = \int_1^x \frac{1}{u}\, du. \tag{1.15}$$

Aplicando el cambio de variable $u = t(x-1)+1$, esta integral se puede reescribir como:

$$\ln(x) = \int_0^1 \frac{x-1}{t(x-1)+1}\, dt. \tag{1.16}$$

Para calcular esta integral de manera numérica, utilizamos la cuadratura de Gauss-Radau, que aproxima la integral de una función $g(t)$ en el intervalo $[0,1]$ mediante una suma ponderada. Para un entero positivo m, la cuadratura de Gauss-Radau se define con un vector de m nodos $t = (t_1, \ldots, t_m)$ y un vector de m pesos $w = (w_1, \ldots, w_m)$ tal que para cualquier polinomio g de grado $2m-2$ se cumple que:

$$\int_0^1 g(t)\, dt \approx \sum_{i=1}^{m} w_i g(t_i), \tag{1.17}$$

siendo $w_i > 0$, $\sum_{i=1}^m w_i = 1$, $w_m = 1/m^2$, $t_i \in (0,1]$, y $t_m = 1$. Al fijar uno de los nodos en el borde del intervalo ($t_m = 1$), la aproximación resultante es un límite inferior global para la función logaritmo.

De modo que la expresión (1.14) resulta en:

$$\log_2(x) \approx \frac{1}{\ln(2)} \sum_{i=1}^{m} w_i \left[\frac{x-1}{t_i(x-1)+1}\right] =: r_m(x). \tag{1.18}$$

Estas funciones $r_m(x)$ forman una familia de aproximaciones racionales del logaritmo que convergen al valor exacto cuando m tiende a infinito. Cada $r_m(x)$ proporciona una cota inferior para el logaritmo, y su error de aproximación disminuye rápidamente a medida que x se acerca a 1.

Siguiendo el enfoque de Brown et al. [8], al aplicar esta secuencia de funciones a la entropía relativa cuántica, se obtiene una cota superior variacional para la entropía relativa cuántica, válida cuando $\text{supp}(\rho) \subseteq \text{supp}(\sigma)$:

$$D(\rho\|\sigma) \leq -\sum_{i=1}^{m} \frac{w_i}{t_i \ln 2} \inf_{Z_i} \left(1 + \text{Tr}\left[\rho(Z_i + Z_i^\dagger + (1-t_i)Z_i^\dagger Z_i)\right] + t_i \,\text{Tr}\left[\sigma Z_i Z_i^\dagger\right]\right), \tag{1.19}$$

donde Z_i son matrices complejas arbitrarias. Esta cota se traduce en un límite inferior variacional para la entropía condicional cuántica de un estado fijo ρ_{ABE}:

$$S(A|E)_{\rho_{ABE}} \geq$$

$$c_m + \sum_{i=1}^{m} \frac{w_i}{t_i \ln 2} \inf_{\{Z_i^a\}_a} \sum_{a=0}^{d-1} \mathrm{Tr}\left[\rho_{ABE}\left(A_0^a \otimes \mathbb{I}_B \otimes \left(Z_i^a + Z_i^{a^\dagger} + (1-t_i) Z_i^{a^\dagger} Z_i^a\right) + t_i \mathbb{I}_{AB} \otimes Z_i^a Z_i^{a^\dagger}\right)\right], \tag{1.20}$$

donde

$$c_m = \sum_{i=1}^{m} \frac{w_i}{t_i \ln 2}. \tag{1.21}$$

En el enfoque de Brown et al. de QKD independiente de dispositivos, la minimización de $S(A|E)$ se realiza sobre todos los posibles estados y bases de medición de Alice y Bob compatibles con las estadísticas medidas. Sin embargo, en nuestro caso, trabajamos con dispositivos caracterizados, donde se ha preestablecido qué tipo de bases utilizarán Alice y Bob (un conjunto de bases MUB). Por lo que únicamente realizamos la minimización sobre los estados ρ_{ABE} compatibles con las estadísticas medidas. Específicamente, queremos resolver el siguiente problema:

$$\begin{aligned} \inf_{\rho_{ABE}, \{Z_i^a\}_{a,i}} c_m + \sum_{i=1}^{m} \sum_{a=0}^{d-1} \frac{w_i}{t_i \ln 2} \mathrm{Tr}\left[\rho_{ABE}\left(A_0^a \otimes \mathbb{I}_B \otimes \left(Z_i^a + Z_i^{a^\dagger} + (1-t_i) Z_i^{a^\dagger} Z_i^a\right) + t_i \mathbb{I}_{AB} \otimes Z_i^a Z_i^{a^\dagger}\right)\right] \\ \text{sujeto a} \quad \rho_{ABE} \geq 0, \quad \mathrm{Tr}(\rho_{ABE}) = 1, \\ \forall k \quad \mathrm{Tr}[\rho_{ABE}(E_k \otimes \mathbb{I}_E)] = f_k, \end{aligned} \tag{1.22}$$

donde E_k son elementos de un POVM (*Positive Operator-Valued Measure*), es decir, un conjunto de operadores que describen las posibles medidas en el sistema cuántico y cumplen con las propiedades de positividad y normalización, y f_k son las probabilidades asociadas.

En el capítulo 3 mostraremos cómo resolver este problema utilizando programación semidefinida (SDP).

Capítulo 2

Modelos de ruido realistas

El término *ruido* se refiere a cualquier tipo de perturbación que provoca errores en la señal cuántica empleada para transmitir información. Este ruido puede provenir de diversas fuentes, como las interferencias ambientales y las imperfecciones de los detectores, y tiene un impacto significativo en la tasa de clave. Por ello, es fundamental considerar modelos de ruido realistas que simulen estas condiciones. En este capítulo, presentamos los dos modelos de ruido realistas de la referencia [21] que hemos incorporado en nuestro protocolo. En cada uno de ellos, el objetivo es obtener dos parámetros fundamentales para ajustar la tasa de clave: la tasa de producción de rondas válidas y la visibilidad.

2.1. Implementación con grados de libertad temporales

Consideramos una fuente láser[1], ubicada entre los laboratorios de Alice y Bob, que produce un estado hiperentrelazado, es decir, entrelazado en más de un grado de libertad. Este estado describe un par de fotones entrelazados, un fotón para Alice y otro para Bob, y es de la forma:

$$|\Psi\rangle = \left|\phi^-\right\rangle_{AB} \otimes \int dt\, f(t)\, |t\rangle_A \otimes |t\rangle_B \,, \tag{2.1}$$

donde $|\phi^-\rangle_{AB}$ representa el entrelazamiento en la polarización de los fotones, y la integral representa el entrelazamiento entre los tiempos de llegada t_A y t_B.

Se define el tiempo de llegada t como el intervalo de tiempo que transcurre desde que los fotones son emitidos por la fuente hasta que son registrados por los respectivos detectores de Alice y Bob. Es una variable continua que puede discretizarse en intervalos de tiempo t_b, siendo t_b el tiempo mínimo que debe transcurrir entre dos eventos de detección para que los detectores los distingan como eventos separados.

Definimos un marco temporal F, fuera del cual los fotones se consideran perdidos. Este marco se define como un múltiplo de t_b, lo que nos permite obtener un sistema discreto de dimensión $d = \frac{F}{t_b}$. Los marcos F en los que tanto el detector de Alice como el de Bob emiten un clic son seleccionados y utilizados para el cálculo de la tasa de clave.

[1] La fuente láser utiliza un proceso llamado conversión paramétrica descendente espontánea (SPDC) para crear fotones entrelazados. En este proceso, un fotón de alta energía se desintegra en dos fotones de menor energía.

La probabilidad de que se produzcan exactamente n pares de fotones entrelazados en el marco $[0, F]$ sigue la distribución de Poisson:

$$P_F(n) = \frac{(\lambda F)^n e^{-\lambda F}}{n!} \tag{2.2}$$

donde λ es el número de pares de fotones producidos por segundo. Entonces, el número medio de pares producidos en un marco temporal es λF, y asumimos que es lo suficientemente pequeño para que la probabilidad de producir más de un par de fotones entrelazados en el mismo marco temporal sea despreciable.

Consideramos dos tipos de ruido: el ruido debido a la interacción de los fotones con el entorno antes de entrar en los laboratorios de Alice y Bob, y el ruido introducido por los detectores.

- **Ruido ambiental**
  - Debido a la interacción con el entorno, un fotón tiene una probabilidad P_L de perderse. Considerando n fotones, la probabilidad de que n_L de ellos se pierdan mientras que el resto lleguen al laboratorio es:

$$\binom{n}{n_L} P_L^{n_L} (1 - P_L)^{n-n_L}. \tag{2.3}$$

  - La probabilidad $P_E(n)$ de que n fotones del entorno entren en el marco temporal F sigue una distribución de Poisson con parámetro νF.
- **Ruido de los detectores**
  - Cada detector tiene una probabilidad P_C de hacer clic cuando llega un fotón.
  - Los detectores también pueden hacer clic en ausencia de fotones, un fenómeno conocido como *dark counts*. La probabilidad $P_D(n)$ de que ocurran n dark counts en F sigue una distribución de Poisson con parámetro μF.

La probabilidad de que tanto Alice como Bob reciban en sus laboratorios i y j fotones, respectivamente, en un intervalo de tiempo F es:

$$P(i,j) = \sum_{n=0}^{\infty} \sum_{n_1=\text{máx}(n-i,0)}^{n} \sum_{n_2=\text{máx}(n-j,0)}^{n} P_F(n) \binom{n}{n_1} P_L^{n_1} (1-P_L)^{n-n_1} \binom{n}{n_2} P_L^{n_2} (1-P_L)^{n-n_2} P_E(i+n_1-n) P_E(j+n_2-n), \tag{2.4}$$

siendo n el número de pares de fotones generados por la fuente, n_1 el número de fotones que se pierden antes de llegar al laboratorio de Alice, n_2 el número de fotones que se pierden antes de llegar al laboratorio de Bob, $i + n_1 - n$ el número de fotones del entorno que llegan al laboratorio de Alice, y $j + n_2 - n$ el número de fotones del entorno que llegan al laboratorio de Bob.

Dados los i fotones que ha recibido Alice, la probabilidad de que su detector emita exactamente un clic en un intervalo F es:

$$P(\text{clic}|i) = (1-P_C)^{i-1} P_C P_D(0) i + (1-P_C)^i P_D(1) = e^{-\mu F} (1-P_C)^i \left(\frac{i P_C}{1-P_C} + \mu F \right). \tag{2.5}$$

El primer sumando representa la probabilidad de que los primeros $i - 1$ fotones no produzcan un clic, pero que el último sí lo haga, combinado con la probabilidad de que no ocurran dark counts y

considerando que cualquiera de los i fotones puede producir el clic. El segundo sumando representa la probabilidad de que ninguno de los i fotones haga clic, pero que aún así ocurra un clic debido a un dark count.

De modo que la probabilidad de que tanto Alice como Bob obtengan un clic en un marco F es:

$$P(11) = \sum_{i,j=0}^{\infty} P(\text{clic}|i)P(\text{clic}|j)P(i,j). \tag{2.6}$$

Antes de mostrar la expresión general correspondiente al desarrollo de la anterior, es interesante considerar una versión aproximada para poder comprender más intuitivamente sus términos. Aplicando la aproximación $P_F(n \geq 2) \approx 0$, podemos simplificar $P(11)$ como:

$$P(11) \approx e^{-F[2(\mu+\nu P_C)+\lambda]} F\beta, \tag{2.7}$$

siendo $\beta = \lambda\alpha^2 + F(\mu+\nu P_C)^2$ y $\alpha = P_C(1-P_L) + F(\mu+\nu P_C)P_L + F(\mu+\nu P_C)(1-P_C)(1-P_L)$.

Para tener una visión general de la expresión (2.7): el término exponencial considera todos los factores que pueden impedir que se registren clics (ausencia de dark counts, ausencia de fotones ambientales detectados o ausencia de pares producidos), mientras que $F\beta$ representa la probabilidad de registrar esos clics. En α observamos que, habiéndose emitido únicamente un par de fotones, la probabilidad de que cada fotón del par llegue y se detecte es $P_C(1-P_L)$, la probabilidad de que llegue pero no se detecte es $(1-P_C)(1-P_L)$, y la probabilidad de que se pierda es P_L. La probabilidad de que un fotón ambiental o un dark count sea el responsable del clic es $F(\mu+\nu P_C)$.

Si consideramos que un detector está más cerca de la fuente que el otro, podemos modificar (2.7) para incluir parámetros diferentes para Alice y Bob:

$$P(11) \approx e^{-\left(\mu_A+\mu_B+\nu_A P_C^A+\nu_B P_C^B+\lambda\right)F} F\left[\lambda\alpha^A\alpha^B + F(\mu^A+\nu^A P_C^A)(\mu^B+\nu^B P_C^B)\right]. \tag{2.8}$$

Teniendo en cuenta la generación de múltiples pares de fotones, escribimos $P(11)$ (2.6) como:

$$P(11) = e^{-(\mu^A+\mu^B+\nu^A P_C^A+\nu^B P_C^B)F} \sum_{n=0}^{\infty} P_F(n)\alpha^A(n)\alpha^B(n), \tag{2.9}$$

siendo $\alpha(n) = (1-P_C+P_CP_L)^{n-1}\left[nP_C(1-P_L) + F(\mu+\nu P_C)(1-P_C+P_CP_L)\right]$. Aquí el primer sumando representa la probabilidad de que los primeros $n-1$ fotones se hayan perdido o no se hayan detectado, pero que el último (que puede ser cualquiera de los n) no se pierda y sí se detecte. El segundo sumando representa la probabilidad de que todos los fotones se hayan perdido o no se hayan detectado, pero que aún así ocurra un clic debido a un dark count o a un fotón ambiental.

2.1.1. Simplificación de $P(11)$

Para trabajar de manera más cómoda con α, realizamos los cambios de variable $S = 1-P_C+P_CP_L$ y $Q = F(\mu+\nu P_C)$. De modo que:

$$\alpha^A(n) = (S^A)^{n-1}\left[nP_C^A(1-P_L^A) + FQ^AS^A\right]. \tag{2.10}$$

$$\alpha^B(n) = (S^B)^{n-1}\left[nP_C^B(1-P_L^B) + FQ^BS^B\right]. \tag{2.11}$$

Expresamos el producto $\alpha^A(n)\alpha^B(n)$ en tres sumandos: el primero incluye un factor n^2, el segundo un factor n, y el tercero es una constante. Esto nos permite descomponer el sumatorio de (2.9) en tres sumatorios, cada uno de las cuales corresponde a una serie de Taylor de la función exponencial:

- En el primer sumatorio identificamos la serie $\sum_{n=0}^{\infty} \frac{x^n}{n!} n^2 = (x + x^2)e^x$.

$$\frac{P_C^A(1-P_L^A)P_C^B(1-P_L^B)}{S^A S^B} e^{-\lambda F} \sum_{n=0}^{\infty} \frac{(\lambda F S^A S^B)^n}{n!} n^2. \tag{2.12}$$

- En el segundo sumatorio identificamos la serie $\sum_{n=0}^{\infty} \frac{x^n}{n!} n = xe^x$.

$$\frac{F}{S^A S^B} \left[P_C^A(1-P_L^A)Q^B S^B + P_C^B(1-P_L^B)Q^A S^A\right] e^{-\lambda F} \sum_{n=0}^{\infty} \frac{(\lambda F S^A S^B)^n}{n!} n. \tag{2.13}$$

- Y en el tercer sumatorio identificamos la serie más simple: $\sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x$.

$$F^2 Q^A Q^B e^{-\lambda F} \sum_{n=0}^{\infty} \frac{(\lambda F S^A S^B)^n}{n!}. \tag{2.14}$$

Combinando los términos y realizando los siguientes cambios de variable:

$$T_{A/B} = Q^{A/B} + \lambda S^{B/A}(1 - S^{A/B}), \tag{2.15}$$

$$\gamma = \lambda P_C^A(1-P_L^A)P_C^B(1-P_L^B) = \lambda(1-S^A)(1-S^B), \tag{2.16}$$

obtenemos finalmente:

$$P(11) = e^{-F(T_A+T_B+\gamma)} \left(F^2 T_A T_B + F\gamma\right). \tag{2.17}$$

$T_{A/B}$ representa el número medio de clics no correlacionados por segundo, que proceden de dark counts, fotones ambientales o fotones del láser cuando una de las partes pierde su fotón o no lo detecta; y γ representa el número medio de fotones entrelazados detectados por segundo, es decir, los fotones provenientes de la fuente láser que no se perdieron y que produjeron un clic.

En nuestro programa tomaremos $T_A = T_B = T$. Esto nos permite definir la relación entre ruido y señal como:

$$\text{Ruido/Señal} = \frac{T}{\gamma + T}. \tag{2.18}$$

2.1.2. Tasa de producción de rondas válidas y visibilidad

El número de rondas válidas por segundo se calcula como:

$$R(d) = \frac{P(11)}{F} = e^{-dt_b(T_A+T_B+\gamma)}(dt_b T_A T_B + \gamma), \tag{2.19}$$

donde se ha sustituido $F = dt_b$. **Esta tasa se multiplicará por la tasa de Devetak-Winter (1.5) para obtener la tasa efectiva de generación de clave, expresada en bits por segundo.**

En la sección 1.2.2 introdujimos el parámetro v representando la visibilidad. Definimos la visibilidad como la probabilidad de que, dado que ambas partes detectaron exactamente un clic, los fotones que hicieron clic fueron los entrelazados provenientes de la fuente láser y no del entorno o dark counts. Por lo tanto, se calcula como:

$$v(d) = \frac{P_S}{P(11)}, \tag{2.20}$$

siendo P_S la probabilidad de que los pares de fotones sobrevivan y se detecten:

$$P_S = e^{-F(Q^A+Q^B)} \sum_{n=0}^{\infty} P_F(n)(S^A S^B)^{n-1}(1-S^A)(1-S^B)n = e^{-F(T^A+T^B+\gamma)}\gamma F. \tag{2.21}$$

Por lo que la visibilidad resulta en:

$$v(d) = \frac{1}{1+dt_b T_A T_B \gamma^{-1}}. \tag{2.22}$$

2.2. Implementación con grados de libertad espaciales

En este modelo, el estado producido por la fuente es de la forma:

$$|\Psi\rangle = \sum_{l=-\infty}^{\infty} c_l \, |l\rangle_A \otimes |-l\rangle_B \,, \tag{2.23}$$

donde l representa los diferentes modos de momento angular, y c_l son coeficientes que dependen de las especificaciones de la fuente.

Debido a la resolución finita de los detectores, discretizamos $|\Psi\rangle$ proyectándolo en un subconjunto finito de modos, denotado por l, con un número total de d modos posibles. La probabilidad de que esta proyección ocurra, $P_P(d)$, la consideramos constante. Esto favorece trabajar en dimensiones más bajas, ya que se reduce la complejidad, mientras se mantiene la misma probabilidad de éxito.

Definimos una ventana de coincidencia temporal Δt, dentro de la cual los clics en los detectores de Alice y Bob son considerados simultáneos. Si un detector registra múltiples clics dentro de la misma ventana temporal, estos clics se toman como un solo evento.

Al igual que en el modelo anterior, un clic puede ser causado por un fotón procedente del láser, por un fotón del ambiente o por un dark count. En la figura 2.1 mostramos un esquema del modelo. En esta implementación, cada parte necesita un detector para cada modo, lo que convierte a los dark counts en la mayor fuente de ruido, ya que generan más coincidencias accidentales dentro de la misma ventana temporal.

Considerando que en total el láser emite n fotones, la probabilidad de que j fotones estén en los modos comprendidos entre $-\frac{d}{2}$ y $\frac{d}{2}$ es:

$$P(j|n) = \frac{(\lambda\Delta t)^n e^{-\lambda\Delta t}}{n!} \binom{n}{j} P_P^j(d) \left[1 - P_P(d)\right]^{n-j}. \tag{2.24}$$

Dados los j fotones producidos por el láser y considerando que r sobreviven, la probabilidad de que no generen ningún clic en uno de los detectores es:

$$P(0|j) = \sum_{r=0}^{j} \binom{j}{r} P_L^{j-r}(1-P_L)^r \binom{r}{0}(1-P_C)^r = [1-P_C(1-P_L)]^j = (1-T)^j, \tag{2.25}$$

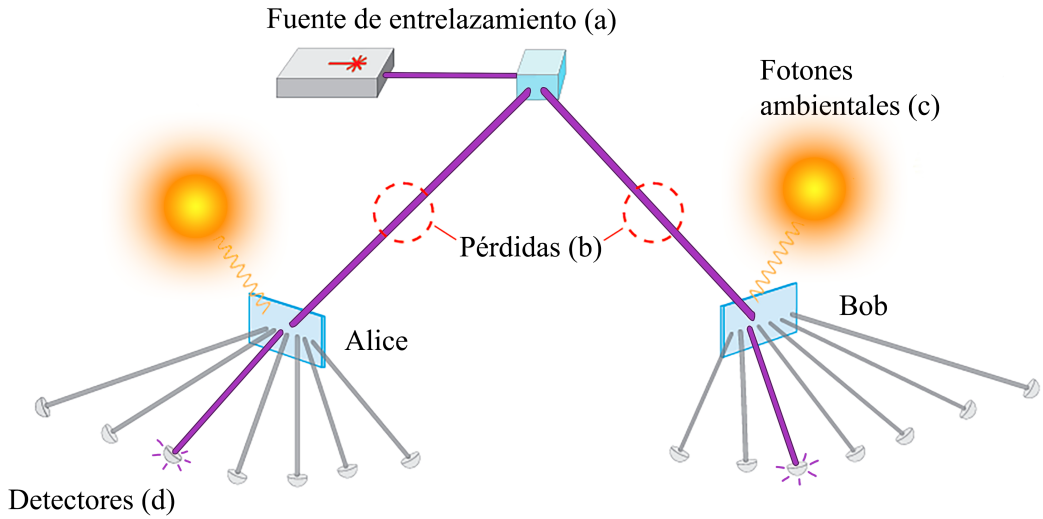

Figura 2.1: Representación esquemática del modelo de ruido: una fuente láser (a) genera λ pares de fotones entrelazados por segundo, siguiendo una distribución de Poisson. Los pares se distribuyen entre Alice y Bob, y a lo largo del camino, cada parte sufre pérdidas (b) con una probabilidad P_L. Además de los fotones entrelazados, cada parte recibe en promedio ν fotones ambientales (c) por segundo, también siguiendo una distribución de Poisson. Cada uno de los modos de momento que se están midiendo tiene asociado un detector (d), el cual emite en promedio μ dark counts por segundo y detecta un fotón con una probabilidad P_C. En la figura, Alice y Bob están registrando un clic en detectores no equivalentes, es decir, en detectores que no miden el mismo modo en sus respectivos sistemas.
Figura extraída de [21] con permiso de los autores, modificada por la autora de este trabajo.

donde $T = P_C(1 - P_L)$. Asumimos que todos los modos sufren las mismas pérdidas, lo que nos permite combinar la probabilidad de detección P_C y las pérdidas del canal P_L en un único parámetro T, facilitando así el cálculo. La expresión (2.25) se ha simplificado aplicando el teorema del binomio, que establece que $(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k$.

Por otro lado, la probabilidad de que se generen uno o más clics en un único detector se calcula de la siguiente manera:

$$\begin{aligned} P(1|j) &= \sum_{r_1=1}^{j} \binom{j}{r_1} P_L^{j-r_1} (1-P_L)^{r_1} \sum_{r_2=1}^{r_1} \binom{d}{1} \binom{r_1}{r_2} \left(1-\frac{1}{d}\right)^{r_1-r_2} (1-P_C)^{r_1-r_2} \left(\frac{1}{d}\right)^{r_2} [1-(1-P_C)^{r_2}] \\ &= d\left[\left(1-T+\frac{T}{d}\right)^{j} - (1-T)^{j}\right], \end{aligned} \tag{2.26}$$

donde r_1 es el número de fotones que sobreviven, r_2 es el número de fotones que están en el modo detectable, y $\frac{1}{d}$ es la probabilidad de que un fotón esté en un modo específico. Por lo que $1-(1-P_C)^{r_2}$ representa la probabilidad de que al menos uno de los r_2 fotones que están en el modo detectable produzca un clic.

Dado que se produjeron j fotones, la probabilidad de que tanto Alice como Bob obtengan un clic en

detectores no equivalentes debido a un fotón del láser es:

$$P(\neq|j) = d(d-1) \sum_{\substack{r_1,r_2,r_3=0 \\ r_0+r_1+r_2+r_3=j}}^{j} \sum_{r_0=0}^{j-1} \frac{j!}{r_0!r_1!r_2!r_3!} \times (P_L^A P_L^B)^{r_0} \left[P_L^B(1-P_L^A)\right]^{r_1} \times$$
$$\times \left[P_L^A(1-P_L^B)\right]^{r_2} \left[(1-P_L^A)(1-P_L^B)\right]^{r_3} \times \sum_{l_1=0}^{r_1} \binom{r_1}{l_1} \left(\frac{d-1}{d}\right)^{r_1-l_1} (1-P_C^A)^{r_1-l_1} \left(\frac{1}{d}\right)^{l_1} \times$$
$$\times \sum_{l_2=0}^{r_2} \binom{r_2}{l_2} \left(\frac{d-1}{d}\right)^{r_2-l_2} (1-P_C^B)^{r_2-l_2} \left(\frac{1}{d}\right)^{l_2} \times$$
$$\times \sum_{\substack{s_3,p_3,q_3=0 \\ s_3+p_3+q_3=r_3}}^{r_3} \left(\frac{1}{d}\right)^{s_3+p_3} \left(\frac{d-2}{d}\right)^{q_3} \frac{r_3!}{s_3!p_3!q_3!} (1-P_C^A)^{p_3+q_3} (1-P_C^B)^{s_3+q_3} \times$$
$$\times \left[1-(1-P_C^A)^{l_1+s_3}\right] \left[1-(1-P_C^B)^{l_2+p_3}\right] = d(d-1)\Bigg\{ \left[(1-T^A)\left(1-T^B\right)\right]^j +$$
$$+ \left[(1-T^A)(1-T^B) + \frac{1}{d}\left[T^A(1-T^B)+T^B(1-T^A)\right]\right]^j - \left[\left(1-T^A+\frac{T^A}{d}\right)(1-T^B)\right]^j -$$
$$- \left[(1-T^A)\left(1-T^B+\frac{T^B}{d}\right)\right]^j \Bigg\}. \tag{2.27}$$

Para facilitar la comprensión de esta expresión, presentamos en la figura 2.2 un diagrama de Venn que muestra cómo se distribuyen los fotones entre los sistemas de Alice y Bob.

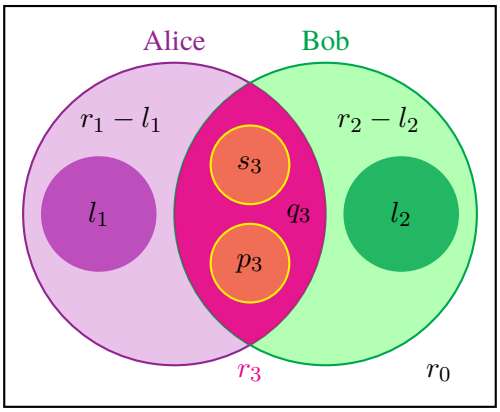

Figura 2.2: Diagrama de Venn que muestra la distribución de fotones en la expresión (2.27). r_0 es el número de fotones que se pierden en ambos sistemas, r_1 es el número de fotones que sobreviven en el sistema de Alice pero se pierden en el sistema de Bob, r_2 es el número de fotones que sobreviven en el sistema de Bob pero se pierden en el sistema de Alice, y r_3 es el número de fotones que sobreviven en ambos sistemas. Dentro de estos grupos, l_1 representa los fotones pertenecientes a r_1 que están en el modo detectable de Alice, y l_2 aquellos de r_2 que están en el modo detectable de Bob. Finalmente, r_3 se descompone en tres grupos: s_3, el número de fotones en el modo detectable de Alice; p_3, el número de fotones en el modo detectable de Bob; y q_3, el número de fotones no detectables por ninguno de los dos sistemas.

De manera similar, la probabilidad de que, dado que se produjeron j fotones, tanto Alice como Bob obtengan un clic en detectores equivalentes debido a un fotón del láser es:

$$
\begin{aligned}
P(=|j) = d \sum_{\substack{r_1,r_2,r_3=0 \\ r_0+r_1+r_2+r_3=j}}^{j} \sum_{r_0=0}^{j-1} \frac{j!}{r_0!r_1!r_2!r_3!} \times (P_L^A P_L^B)^{r_0} \left[P_L^B(1-P_L^A)\right]^{r_1} \times \\
\times \left[P_L^A(1-P_L^B)\right]^{r_2} \left[(1-P_L^A)(1-P_L^B)\right]^{r_3} \times \sum_{l_1=0}^{r_1} \binom{r_1}{l_1} \left(\frac{d-1}{d}\right)^{r_1-l_1} (1-P_C^A)^{r_1-l_1} \left(\frac{1}{d}\right)^{l_1} \times \\
\times \sum_{l_2=0}^{r_2} \binom{r_2}{l_2} \left(\frac{d-1}{d}\right)^{r_2-l_2} (1-P_C^B)^{r_2-l_2} \left(\frac{1}{d}\right)^{l_2} \times \\
\times \sum_{l_3=0}^{r_3} \binom{r_3}{l_3} \left(\frac{d-1}{d}\right)^{r_3-l_3} \left(\frac{1}{d}\right)^{l_3} (1-P_C^A)^{r_3-l_3} (1-P_C^B)^{r_3-l_3} \times \\
\times \left[1-(1-P_C^A)^{l_1+l_3}\right] \left[1-(1-P_C^B)^{l_2+l_3}\right] = d\Bigg\{ \left[(1-T^A)\left(1-T^B\right)\right]^j + \\
+ \left[(1-T^A)(1-T^B) + \frac{1}{d}\left[T^A(1-T^B) + T^B(1-T^A) + T^AT^B\right]\right]^j - \\
- \left[\left(1-T^A+\frac{T^A}{d}\right)(1-T^B)\right]^j - \left[(1-T^A)\left(1-T^B+\frac{T^B}{d}\right)\right]^j \Bigg\}.
\end{aligned}
\tag{2.28}
$$

Observamos que, en el diagrama de Venn correspondiente a este caso, dentro del área r_3 habría un único subconjunto, l_3, que representa el número de fotones que se encuentran en el modo detectable, el cual ahora es común para Alice y Bob.

Procedemos ahora a estudiar los clics debidos a dark counts. De nuevo, siguen una distribución de Poisson, con múltiples clics en el mismo detector en Δt contados como uno solo. Por lo tanto, la probabilidad de que no ocurran dark counts en un detector en el intervalo Δt es $e^{-\mu\Delta t}$, mientras que la probabilidad de que ocurra al menos un dark count en un detector es $1-e^{-\mu\Delta t}$. En total, la probabilidad de que ocurran n dark counts en todos los d detectores es:

$$
P_D(n,d) = \binom{d}{n} \left(e^{-\mu\Delta t}\right)^{d-n} \left(1-e^{-\mu\Delta t}\right)^n, \tag{2.29}
$$

lo que nos proporciona otra cantidad que necesitamos: la probabilidad de que, dado que un detector ya hizo clic debido a un fotón del láser, los otros detectores no hagan clic debido a dark counts. Esta probabilidad se denota como $P_D(0,d-1)$, que indica que no se detectan dark counts en los $d-1$ detectores restantes, y se calcula como:

$$
P_D(0,d-1) = P_D(1,d)\frac{1}{d} + P_D(0,d) = e^{-(d-1)\mu\Delta t}. \tag{2.30}
$$

Finalmente, consideramos el último tipo de clics que registran los detectores: los provenientes de fotones ambientales. Considerando que hay r fotones en el mismo modo, la probabilidad de que al menos uno de ellos produzca un clic es $1-(1-P_C)^r$. Por lo tanto, para un total de q fotones, la probabilidad de que uno de los r fotones produzca un clic mientras que el resto no lo haga es:

$$
\sum_{r=1}^{q} \binom{q}{r} \left(\frac{d-1}{d}\right)^{q-r} (1-P_C)^{q-r} \left(\frac{1}{d}\right)^r \left[1-(1-P_C)^r\right] = \left[1-\frac{P_C(d-1)}{d}\right]^q - (1-P_C)^q. \tag{2.31}
$$

Multiplicando (2.31) por la distribución de Poisson que siguen los fotones ambientales y por el número de modos, obtenemos la probabilidad de que se registre un único clic en todos los detectores debido a un fotón ambiental:

$$\begin{aligned} P_E(1,d) &= d\sum_{q=0}^{\infty}(\nu\Delta t)^q\left[1-\frac{P_C(d-1)}{d}\right]^q\frac{e^{-\nu\Delta t}}{q!} - d\sum_{q=0}^{\infty}(\nu\Delta t)^q(1-P_C)^q\frac{e^{-\nu\Delta t}}{q!} = \\ &= dP_E(0^*,d)\left(1-e^{-P_C\nu\Delta t/d}\right), \end{aligned} \tag{2.32}$$

siendo $P_E(0^*,d) = e^{-P_C\nu\Delta t(d-1)/d}$ la probabilidad de que, dado que un detector ya ha registrado un clic debido a un fotón del láser o a un dark count, r de los q fotones ambientales terminen en este detector, mientras que los $q-r$ restantes se distribuyan en otros detectores y ninguno de ellos produzca un clic. El otro factor, $1-e^{-P_C\nu\Delta t/d}$, representa la probabilidad de que al menos uno de los fotones ambientales produzca un clic en el detector específico.

Con las probabilidades previamente definidas, ahora podemos calcular la probabilidad de que, dado que se produjeron j fotones, un solo detector registre un clic:

$$\begin{aligned} P(1) &= P(1|j)P_D(0,d-1)P_E(0^*,d) + P(0|j)P_D(1,d)P_E(0^*,d) + P(0|j)P_D(0,d)P_E(1,d) = \\ &= dP_D(0,d-1)P_E(0^*,d)\left[\left(1-T+\frac{T}{d}\right)^j - (1-T)^j e^{-\Delta t(\mu+\frac{P_C\nu}{d})}\right]. \end{aligned} \tag{2.33}$$

Entonces, la probabilidad de que tanto Alice como Bob obtengan un clic es:

$$\begin{aligned} P(11) &= \sum_{n=0}^{\infty}\sum_{j=0}^{n}\frac{(\lambda\Delta t)^n e^{-\lambda\Delta t}}{n!}\binom{n}{j}P_P^j(d)[1-P_P(d)]^{n-j}\times \\ &\times\Big\{\Big[P(1|j)P_D(0,d-1)P_E(0^*,d) + P(0|n)P_D(1,d)P_E(0^*,d) + P(0|n)P_D(0,d)P_E(1,d)\Big]^A\times \\ &\times\Big[P(1|j)P_D(0,d-1)P_E(0^*,d) + P(0|n)P_D(1,d)P_E(0^*,d) + P(0|n)P_D(0,d)P_E(1,d)\Big]^B + \\ &+\Big[P_D(0,d-1)P_E(0^*,d)\Big]^A\Big[P_D(0,d-1)P_E(0^*,d)\Big]^B\Big[P(\neq|j) + P(=|j) - P^A(1|j)P^B(1|j)\Big]\Big\}. \end{aligned} \tag{2.34}$$

Simplificando la expresión, obtenemos:

$$P(11) = de^{-\Delta t(d-1)(\mu^A+\frac{\xi^A}{d}+\mu^B+\frac{\xi^B}{d})}e^{-\gamma\Delta t}\left\{d\left(1-e^{-\Delta t(\mu^A+\frac{\xi^A}{d})}\right)\left(1-e^{-\Delta t(\mu^B+\frac{\xi^B}{d})}\right)+e^{\frac{\gamma\Delta t}{d}}-1\right\}, \tag{2.35}$$

donde

$$\xi^{A/B} = P_C^{A/B}\nu^{A/B} + \lambda P_P(d)P_C^{B/A}\left(1-P_L^{B/A}\right)\left(1-P_C^{A/B}+P_C^{A/B}P_L^{A/B}\right), \tag{2.36}$$

y γ es igual que en el anterior modelo:

$$\gamma = P_P(d)\lambda P_C^A(1-P_L^A)P_C^B(1-P_L^B). \tag{2.37}$$

Todos los términos que aparecen en estas variables son constantes experimentales que no dependen de d. γ representa el número medio de fotones entrelazados detectados, mientras que $\xi^{A/B}$ representa el número medio de clics no correlacionados debido al ambiente, pérdidas e ineficiencias del detector.

En nuestro programa tomaremos $\xi^A = \xi^B = \xi$, y $\mu^A = \mu^B = \mu$. Esto nos permite definir la relación entre ruido y señal como:

$$\text{Ruido/Señal} = \frac{\xi + \mu}{\gamma + \xi + \mu} \tag{2.38}$$

2.2.1. Tasa de producción de rondas válidas y visibilidad

Calculamos el número de rondas válidas por segundo como:

$$R(d) = \frac{P(11)}{\Delta t} = \\ = \frac{d}{\Delta t} e^{-\Delta t\left((d-1)(\mu^A + \frac{\xi^A}{d} + \mu^B + \frac{\xi^B}{d}) + \gamma\right)} \left\{ d\left(1 - e^{-\Delta t(\mu^A + \frac{\xi^A}{d})}\right)\left(1 - e^{-\Delta t(\mu^B + \frac{\xi^B}{d})}\right) + e^{\frac{\gamma \Delta t}{d}} - 1 \right\}. \tag{2.39}$$

Finalmente, para obtener la expresión de la visibilidad, necesitamos la probabilidad P_S de que un par entrelazado produzca un clic en ambos laboratorios, mientras que los demás detectores no emitan clic. Para esto, consideramos la probabilidad $P(= |j)$ de que los detectores equivalentes emitan un clic en ambos laboratorios, lo cual es atribuible a los fotones correlacionados y al ruido. Asimismo, tomamos en cuenta la probabilidad $P(\neq |j)$ de que los detectores no equivalentes emitan un clic, lo cual es atribuible solo al ruido. Restamos la contribución de $P(\neq |j)$ de la contribución de $P(= |j)$ en la expresión $P(11)$, obteniendo:

$$P_S = d e^{-\Delta t(d-1)(\mu^A + \frac{\xi^A}{d} + \mu^B + \frac{\xi^B}{d})} e^{-\gamma \Delta t} \left(e^{\frac{\Delta t \gamma}{d}} - 1\right). \tag{2.40}$$

Por lo tanto, la visibilidad es:

$$v(d) = \frac{P_S}{P(11)} = \frac{e^{\frac{\gamma \Delta t}{d}} - 1}{d\left(1 - e^{-\Delta t(\mu^A + \frac{\xi^A}{d})}\right)\left(1 - e^{-\Delta t(\mu^B + \frac{\xi^B}{d})}\right) + e^{\frac{\gamma \Delta t}{d}} - 1}. \tag{2.41}$$

Capítulo 3

Metodología

Recapitulamos el problema derivado en la sección 1.3.2. Introducimos primero la programación semidefinida (SDP, por sus siglas en inglés), la herramienta clave en su solución. A continuación, explicamos cómo será implementada, destacando el uso del lenguaje de programación *Julia*, el paquete de optimización *JuMP* y el solucionador *MOSEK*. Por último, dedicamos la última sección de este capítulo a la presentación y explicación del código desarrollado.

3.1. Programación semidefinida (SDP)

La programación semidefinida es una técnica de optimización convexa donde el objetivo es minimizar (o maximizar) una función lineal bajo ciertas restricciones, entre ellas, la restricción de que las matrices deben ser semidefinidas positivas. De manera estándar, un problema de SDP se puede formular como [22]:

$$\begin{aligned} &\min_{\rho} \operatorname{Tr}(W\rho) \\ &\text{sujeto a} \quad \rho \geq 0 \\ &\Phi(\rho) = A \end{aligned} \tag{3.1}$$

Esta estructura resulta particularmente útil en el ámbito teórico. Sin embargo, en aplicaciones prácticas, es más conveniente emplear una formulación más flexible, como la siguiente:

$$\begin{aligned} &\min_{\{\rho_i\}} \sum_i \operatorname{Tr}(W_i\rho_i) \\ &\text{sujeto a} \quad \Lambda_j(\rho_i) \geq 0 \quad \forall i,j \\ &\Phi_j(\rho_i) = A_{ij} \quad \forall i,j \end{aligned} \tag{3.2}$$

A continuación, se explican los términos de esta formulación:

- **Función objetivo**: $\sum_i \operatorname{Tr}(W_i\rho_i)$ es la función que se desea minimizar (o maximizar). W_i son matrices hermíticas que actúan como pesos, mientras que ρ_i son las matrices variables del problema. La función traza asegura que la función objetivo sea lineal en ρ_i.

- **Restricciones de positividad semidefinida**: $\Lambda_j(\rho_i) \geq 0 \quad \forall i,j$. Aseguran que, para cada i y j, las matrices resultantes de aplicar la transformación lineal Λ_j sobre las matrices ρ_i sean semidefinidas positivas.

- **Restricciones lineales**: $\Phi_j(\rho_i) = A_{ij} \quad \forall i, j$. Imponen condiciones lineales adicionales sobre las matrices ρ_i, donde Φ_j son funciones lineales y A_{ij} son elementos de una matriz rectangular A. Cada A_{ij} establece un valor que el resultado de $\Phi_j(\rho_i)$ debe satisfacer.

Una de las características interesantes de la SDP es que cualquier función convexa que pueda ser minimizada utilizando SDP, puede ser reescrita como una función lineal, trasladando la no linealidad a las restricciones.

3.2. Transformación de (1.22) a un problema de SDP

Por comodidad del lector, reescribimos de nuevo el problema formulado en la sección 1.3.2:

$$\begin{aligned}\inf_{\rho_{ABE},\{Z_i^a\}_{a,i}} c_m + \sum_{i=1}^{m}\sum_{a=0}^{d-1}\frac{w_i}{t_i \ln 2}\mathrm{Tr}\left[\rho_{ABE}\left(A_0^a \otimes \mathbb{I}_B \otimes \left(Z_i^a + Z_i^{a^\dagger} + (1-t_i)Z_i^{a^\dagger}Z_i^a\right) + t_i \mathbb{I}_{AB} \otimes Z_i^a Z_i^{a^\dagger}\right)\right] \\ \text{sujeto a} \quad \rho_{ABE} \geq 0, \quad \mathrm{Tr}(\rho_{ABE}) = 1, \\ \forall k \quad \mathrm{Tr}[\rho_{ABE}(E_k \otimes \mathbb{I}_E)] = f_k,\end{aligned} \tag{3.3}$$

Para transformar este problema en un problema de SDP, necesitamos convertir la función objetivo en una forma lineal. Observamos que la no linealidad surge de los productos de las variables $Z_i^{a\dagger}Z_i^a$ y $Z_i^a Z_i^{a\dagger}$.

En primer lugar, absorbemos las variables Z_i^a dentro de ρ_{ABE} mediante la definición de nuevas variables $\zeta_i^a = \mathrm{Tr}_E\left[\rho_{ABE}\left(\mathbb{I}_{AB} \otimes Z_i^{aT}\right)\right]$. Aunque esto reduce la complejidad de la expresión, la función objetivo sigue siendo no lineal debido a los productos generados $\zeta_i^{a\dagger}\zeta_i^a$ y $\zeta_i^a\zeta_i^{a\dagger}$.

Normalmente, en problemas de optimización, un producto no lineal de variables se maneja construyendo una *matriz de momentos*. Esta matriz es una herramienta que organiza los productos de las variables de manera que simplifica su análisis y facilita la formulación de restricciones. Sin embargo, en este caso, enfrentamos una dificultad adicional debido a las dimensiones de las matrices involucradas. Las variables $\zeta_i^{a\dagger}$ y ζ_i^a son de tamaño $d^2 \times d^2$, lo que generaría matrices de momentos de ese mismo tamaño. Por lo tanto, esta construcción resulta poco práctica en nuestro contexto.

La solución a este problema consiste en construir una matriz de momentos por *bloques* [23]. Esta técnica permite dividir la matriz grande en bloques, organizando los elementos en matrices más pequeñas y manejables. Cada bloque representa momentos específicos, como $\eta_i^a = \zeta_i^{a\dagger}\zeta_i^a$ y $\theta_i^a = \zeta_i^a\zeta_i^{a\dagger}$. De este modo, estos bloques capturan los términos no lineales necesarios, evitando así la construcción de una matriz de grandes dimensiones.

Por lo tanto, el problema descrito anteriormente puede ser reformulado como el siguiente problema de SDP:

$$\begin{aligned}\min_{\sigma,\{\zeta_i^a,\eta_i^a,\theta_i^a\}_{a,i}} c_m + \sum_{i=1}^{m}\sum_{a=0}^{d-1}\frac{w_i}{t_i \ln 2}\mathrm{Tr}\left[(A_0^a \otimes \mathbb{I}_B)\left(\zeta_i^a + \zeta_i^{a\dagger} + (1-t_i)\eta_i^a\right) + t_i\theta_i^a\right] \\ \text{sujeto a} \quad \mathrm{Tr}(\sigma) = 1, \quad \forall k \quad \mathrm{Tr}(E_k\sigma) = f_k \\ \forall a, i \quad \Gamma_{a,i}^1 := \begin{pmatrix}\sigma & \zeta_i^a \\ \zeta_i^{a\dagger} & \eta_i^a\end{pmatrix} \geq 0, \quad \Gamma_{a,i}^2 := \begin{pmatrix}\sigma & \zeta_i^{a\dagger} \\ \zeta_i^a & \theta_i^a\end{pmatrix} \geq 0.\end{aligned} \tag{3.4}$$

Los elementos E_k del conjunto POVM se definen como:

$$E_k := \sum_{i,j=0}^{d-1} E_{k,i,j} = \sum_{i,j=0}^{d-1} \Pi_k^i \otimes \Pi_k^{j^T}, \tag{3.5}$$

donde Π_k^i es el proyector sobre el i-ésimo vector de la k-ésima MUB, con k variando de 0 a d. El primer proyector está asociado al sistema de Alice y el segundo al sistema de Bob. Establecemos que $E_{k,i,j}$ es el proyector que representa el evento en el que Alice y Bob obtienen el mismo resultado, lo que nos permite reescribir E_k como:

$$E_k := \sum_{i=0}^{d-1} \Pi_k^i \otimes \Pi_k^{i^T}. \tag{3.6}$$

Las probabilidades f_k se definen como:

$$f_k := \text{Tr}(\rho_{iso} E_k), \tag{3.7}$$

donde cabe recordar que ρ_{iso} (1.3) es el estado que comparten Alice y Bob.

Además:

- σ es la matriz densidad del sistema, que debe ser hermítica y tiene dimensión $d^2 \times d^2$. Su diagonal es real y contiene d^2 variables. El triángulo superior (o inferior) está formado por elementos complejos, lo que aporta $2 \cdot \frac{d^2 \cdot (d^2-1)}{2}$ variables, considerando tanto la parte real como la imaginaria de cada elemento. En total, σ contiene d^4 variables.

- ζ_i^a son matrices complejas arbitrarias de dimensión $d^2 \times d^2$, y cada una de ellas contiene $2 \cdot (d^2)^2$ variables. Dado que hay d índices a y m índices i, estas matrices aportan un total de $2 \cdot d \cdot m \cdot d^4$ variables.

- η_i^a y θ_i^a son matrices hermíticas de dimensión $d^2 \times d^2$, y todas ellas suman $2 \cdot d \cdot m \cdot d^4$ variables.

En total, el problema contiene $d^4 \cdot (4 \cdot d \cdot m + 1)$ variables, lo que resulta en una complejidad computacional polinómica de $O(m \cdot d^5)$. Por lo tanto, para grandes valores de d y m, los tiempos de cálculo aumentan significativamente, haciendo que el proceso se vuelva ineficiente.

Por otro lado, analizamos el número de restricciones que podemos contabilizar:

- $\text{Tr}(\sigma) = 1$ aporta 1 restricción.

- La condición $\text{Tr}(E_k \sigma) = f_k$ se aplica para cada k, resultando en $d + 1$ restricciones.

- Las condiciones de positividad semidefinida suman $2 \cdot d \cdot m$ restricciones.

En total, el problema contiene $2 + d$ restricciones escalares y $2 \cdot m \cdot d$ restricciones cónicas. Cada una de estas últimas implica una matriz de dimensión $2d^2 \times 2d^2$. Aunque el número de matrices involucradas es considerable, el hecho de utilizar múltiples matrices de menor tamaño, en lugar de una única matriz de mayor dimensión, mejora la eficiencia en la resolución del problema.

3.3. Implementación Computacional

Después de haber planteado el problema de minimización en programación semidefinida y de presentar los modelos de ruido en el capítulo 2, disponemos de toda la información necesaria para desarrollar el código requerido. Para ello, utilizamos el lenguaje de programación Julia en el entorno de *Visual Studio Code*.

Julia ha sido elegida por su biblioteca JuMP [24], que admite números complejos y ofrece una sintaxis sencilla para formular variables, restricciones y funciones objetivo. Además, JuMP es compatible con varios solucionadores que admiten matrices semidefinidas positivas, incluyendo SCS, Clarabel y MOSEK. Inicialmente, intentamos trabajar con SCS y Clarabel, pero ambos presentaron limitaciones manejando matrices de grandes dimensiones. Finalmente, MOSEK resultó ser la opción más eficiente.

A continuación, explicamos la implementación por secciones.

3.3.1. Paquetes y función que incorpora las MUB

```
using JuMP
using MosekTools
using LinearAlgebra
using Plots
using Ket
import Dualization
import JLD2
```

Además de los paquetes ya presentados, empleamos `LinearAlgebra` para realizar operaciones con matrices y `Plots` para crear visualizaciones gráficas de los resultados obtenidos.

El paquete `Ket` proporciona las funciones `ket(i, d)`, que genera un ket de dimensión d con un elemento no nulo en la posición i, `ketbra(v)`, que genera un ketbra a partir del vector v, y `cleanup!`, cuyo uso se detalla en 3.3.2.

La implementación del paquete `Dualization` se describe en 3.3.4.

Por último, `JLD2` es utilizado para cargar el archivo `mubs.jld2`, el cual contiene $d+1$ bases MUB para $2 \leq d \leq 13$, construidas según lo explicado en la sección 1.2.2. Para $d = 6, 10, 12$ las bases son solo aproximadamente no sesgadas.

```
function numerical_mub(d::Int)
    mub_dict = load("mubs.jld2")
    return mub_dict["mubs"][d]
end
```

3.3.2. Estado isótropo (1.3), E_k y f_k

```
function isotropo(d, v::Real)
    phi = zeros(d^2)
    for i = 1:d
        phi_i = 1 / sqrt(d) * kron(ket(i, d), ket(i, d))
```

```
        phi += phi_i
    end
    return v * ketbra(phi) + (1 - v) / d^2 * Matrix(I, d^2, d^2)
end
```

```
function E(d)
    E_matrices = [zeros(ComplexF64, d^2, d^2) for _ = 1:d+1]
    for k = 1:d+1
        for i = 1:d
            P = kron(
                ketbra(mub(d)[k][:, i]),
                transpose(ketbra(mub(d)[k][:, i]))
            )
            E_matrices[k] += P
        end
    end
    cleanup!.(E_matrices)
    return E_matrices
end
```

`E(d)[k]` nos proporciona E_k (3.6). Utilizamos la función `cleanup!` para eliminar las partes reales o imaginarias de `E_matrices` que son menores que un valor de tolerancia predefinido. Esto elimina los residuos numéricos pequeños generados por errores de redondeo.

```
function f(d, v)
    prob = zeros(d + 1)
    for k = 1:d+1
        prob[k] = dot(isotropo(d, v), E(d)[k])
    end
    return prob
end
```

`f(d, v)[k]` nos proporciona f_k (3.7). Observamos que, para un valor fijo de d y v, todas las probabilidades f_k son iguales. Esto se debe a la naturaleza del estado isótropo, que garantiza probabilidades iguales en todo el conjunto de MUB.

Es importante notar el uso de la función `dot` en lugar de la función traza. En este contexto, ambas expresiones son equivalentes:

$$dot(A, B) = \langle vec(A), vec(B)\rangle = \langle A, B\rangle = \sum_{ij} conj(A_{ij})(B_{ij}) = \text{Tr}(A'B) = \text{Tr}(AB). \quad (3.8)$$

La ventaja de usar `dot` radica en su menor complejidad computacional: requiere aproximadamente d^2 operaciones, frente a las d^3 operaciones que implicaría calcular la traza mediante una multiplicación matricial.

3.3.3. Cuadratura de Gauss-Radau

```
function gauss_radau(m::Int)
    diagonal = fill(0.5, m)
    diagonal[m] = (3 * m - 1) / (4 * m - 2)
    superdiagonal = [n / (2 * sqrt(4 * n^2 - 1)) for n = 1:m-1]
    J = SymTridiagonal(diagonal, superdiagonal)
    t, vecs = eigen(J)
```

```
    w = vecs[1, :] .^ 2
    return w, t
end
```

Esta función calcula los m nodos t y los m pesos w de la cuadratura de Gauss-Radau, mediante el cálculo de valores propios y vectores propios, según se describe en la referencia [25]. Primero, se construye una matriz tridiagonal simétrica J, donde la diagonal principal y la superdiagonal se definen con valores específicos. (Una matriz tridiagonal tiene elementos distintos de cero solo en la diagonal principal y en las diagonales adyacentes.) A continuación, se obtienen los valores propios de J, que corresponden a los nodos t, y se eleva al cuadrado la primera fila de los vectores propios para calcular los pesos w.

3.3.4. $S(A|E)$ (3.4)

El solucionador necesita que el problema de SDP esté en una forma estándar para resolverlo. JuMP se encarga de esta transformación y ofrece diferentes opciones para hacerlo: puede interpretar el problema inicial como su versión primal o como su versión dual. Probamos a resolverlo de ambas formas (líneas 3 y 4) y descubrimos que utilizar `Dualization` es más eficiente, pues reduce significativamente el tiempo de ejecución.

```
function Sae(m, d, v)
    model = Model()
    # set_optimizer(model, Mosek.Optimizer)
    set_optimizer(model, Dualization.dual_optimizer(Mosek.Optimizer))

    w, t = gauss_radau(m)
    c_m = sum(w ./ (t .* log(2))) # (1.21)

    # Matriz identidad
    I_B = Matrix(I, d, d)

    # Variables
    @variable(model, sigma[1:d^2, 1:d^2], Hermitian)
    ζ = [@variable(model, [1:d^2, 1:d^2] in ComplexPlane())
        for i = 1:m, a = 1:d]
    η = [@variable(model, [1:d^2, 1:d^2], Hermitian)
        for i = 1:m, a = 1:d]
    θ = [@variable(model, [1:d^2, 1:d^2], Hermitian)
        for i = 1:m, a = 1:d]

    # Restricciones
    @constraint(model, tr(sigma) == 1)
    for k = 1:d+1
        @constraint(model, dot(sigma, E(d)[k]) == f(d, v)[k])
    end
    for i = 1:m
        for a = 1:d
            @constraint(model, Hermitian([sigma ζ[i, a];
                ζ[i, a]' η[i, a]]) in HermitianPSDCone())
            @constraint(model, Hermitian([sigma ζ[i, a]';
                ζ[i, a] θ[i, a]]) in HermitianPSDCone())
        end
    end
```

```
34
35      # Definimos la función a minimizar
36      obj = c_m + sum(
37          w[i] / (t[i] * log(2)) * (
38              dot(
39                  kron(ketbra(ket(a, d)), I_B),
40                  Hermitian(
41                      ζ[i, a] + ζ[i, a]'
42                  ) +
43                  (1 - t[i]) * η[i, a]
44              ) +
45              t[i] * tr(θ[i, a])
46          )
47          for i = 1:m, a = 1:d
48      )
49
50      # Marcamos el objetivo
51      @objective(model, Min, real(obj))
52
53      # Resolvemos
54      optimize!(model)
55      return objective_value(model)
56  end
```

3.3.5. Entropía binaria (4) y $H(A/B)$ (1.11)

```
function entropia_binaria(p::Real)
    function log(x)
        if x == 0
            sol = 0.0
        else
            sol = log2(x)
        end
    end
    return -p * log(p) - (1 - p) * log(1 - p)
end
```

```
function Hab(d, v)
    if v == 1 || v == 0
        return 0.0
    else
        return entropia_binaria(v + (1 - v) / d) +
            (1 - v - (1 - v) / d) * log2(d - 1)
    end
end
```

3.3.6. Tasa de clave de la implementación con grados de libertad temporales

```
function R_time(d, t_b, T, gamma)
    if T == Inf
        return 0.0
    else
        return (d * t_b * T^2 + gamma)
```

```
            * exp(-d * t_b * (2 * T + gamma))
    end
end
```

R_{time} es la tasa de rondas válidas de la implementación con grados de libertad temporales (2.19).

```
function K_time(m, d, noise_signal)
    t_b = 1.31 * 10^-9
    lambda = 2 * 10^5
    PLA = 0.01
    PLB = 0.984
    PCA = 0.6
    PCB = 0.6
    gamma = lambda * PCA * PCB * (1 - PLA) * (1 - PLB) # (2.16)

    T = noise_signal * gamma / (1 - noise_signal)      # (2.18)
    v = 1 / (1 + d * t_b * T^2 * gamma^-1)             # (2.22)

    return R_time(d, t_b, T, gamma) * (Sae(m, d, v) - Hab(d, v))
end
```

3.3.7. Tasa de clave de la implementación con grados de libertad espaciales

```
function R_space(d, Δt, μ, ξ, gamma)
    return (d / Δt) *
        exp(
            -Δt * ((d - 1) * (2 * μ + 2 * ξ / d) + gamma)
        ) * (
            d * (1 - exp(-Δt * (μ + ξ / d)))^2 +
            exp(Δt * gamma / d) - 1
        )
end
```

R_{space} es la tasa de rondas válidas de la implementación con grados de libertad espaciales (2.39).

```
function K_space(m, d, noise_signal)
    Δt = 10^-7
    PP = 1
    lambda = 2 * 10^5
    PLA = 0.01
    PLB = 0.984
    PCA = 0.6
    PCB = 0.6
    μ = 500
    gamma = PP * lambda * PCA * PCB * (1 - PLA) * (1 - PLB) # (2.37)

    ξ = (μ * (noise_signal - 1) + noise_signal * gamma) /
        (1 - noise_signal)                                  # (2.38)

    v = (exp(Δt * gamma / d) - 1) / (
        exp(Δt * gamma / d) - 1
        + d * (1 - exp(-Δt * (μ + ξ / d)))^2)               # (2.41)

    return R_space(d, Δt, μ, ξ, gamma) * (Sae(m, d, v) - Hab(d, v))
end
```

Capítulo 4

Resultados

Representamos la tasa de clave frente a la relación ruido/señal de los dos modelos de ruido realistas considerados, para diferentes dimensiones del sistema.

Debido al rápido crecimiento del número de variables a medida que aumenta la dimensión en el problema (3.4), un ordenador convencional como el utilizado por la autora, con un procesador de 8 hilos, solo permite trabajar de manera eficiente hasta $d = 4$. Para dimensiones superiores, la ejecución se vuelve excesivamente lenta o incluso se detiene. Por ello, las gráficas 4.3 y 4.6 (para $d = 6$) fueron generadas en otro ordenador más potente, con un procesador de 32 hilos.

Los parámetros de ruido empleados en estas representaciones se especificaron en las secciones anteriores 3.3.6 y 3.3.7. Estos se basan en experimentos reales realizados con detectores de fotones.

En primer lugar, se hace una representación para cada modelo de ruido en un rango amplio de ruido/señal (4.1, 4.4) y se observa que es en el extremo superior del intervalo $[0, 1]$ donde las tasas de clave para las dimensiones consideradas experimentan un cambio crítico. Esto ocurre porque, a medida que la relación ruido/señal aumenta, el parámetro v, que previamente se mantenía constante en su valor máximo (en el modelo (2.22), debido a que $t_b T^2$ es despreciable para ruido/señal < 0.9; y en el modelo (2.41), debido a que $\xi \Delta t$ es despreciable en el mismo rango), comienza a disminuir rápidamente. Esta disminución afecta directamente a la diferencia $S(A|E) - H(A|B)$, llevando la tasa de clave a cero en este rango.

En el intervalo en el que $v \approx 1$, resulta interesante destacar el valor constante que presenta la tasa de clave. Este valor se determina como el producto del valor máximo que puede alcanzar la tasa de Devetak-Winter, $\log_2 d$ bits/ronda, por la tasa de rondas válidas R (rondas/s). En este rango, R se aproxima al valor de γ para ambos modelos de ruido. Por lo que si tomamos valores mayores de λ, P_C^A, P_C^B y/o valores menores de P_L^A y P_L^B que los considerados, se alcanzarán tasas de clave superiores.

Para establecer una comparación con nuestros resultados, también representamos la tasa de clave obtenida con el método más comúnmente empleado antes del enfoque utilizado en este trabajo: el método de min-entropía. Este consiste en acotar inferiormente la entropía relativa de von Neumann con la min-entropía, cuyo cálculo se detalla en el anexo A.

4.1. Implementación con grados de libertad temporales

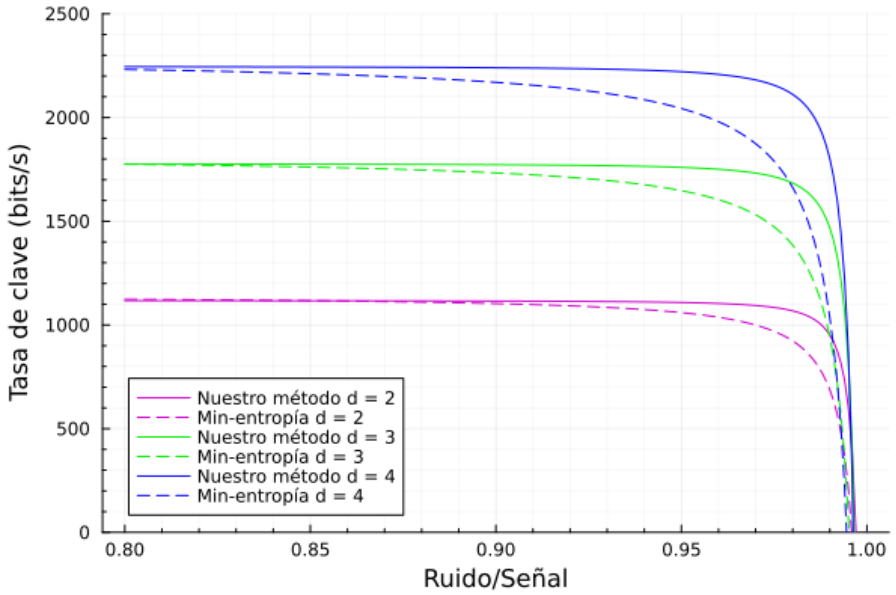

Figura 4.1: Tasa de clave (en bits por segundo) frente a la relación ruido/señal (adimensional) para el modelo de grados de libertad temporales en las dimensiones $d = 2, 3, 4$. Los cálculos se realizaron utilizando $m = 6$ puntos de la cuadratura de Gauss-Radau.

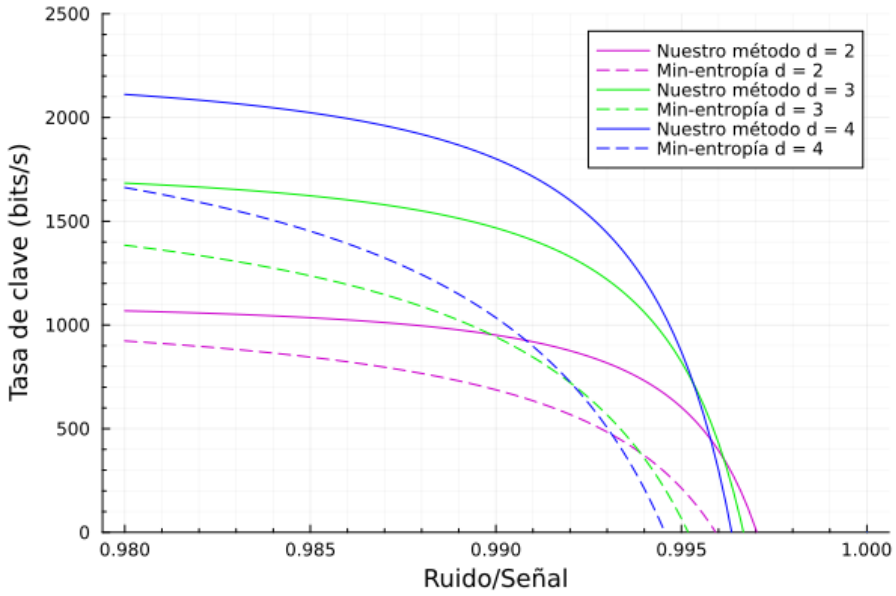

Figura 4.2: Tasa de clave (en bits por segundo) frente a la relación ruido/señal (adimensional) para el modelo de grados de libertad temporales en las dimensiones $d = 2, 3, 4$, centrado en el intervalo donde ocurre el cambio más significativo. Los cálculos se realizaron utilizando $m = 6$ puntos de la cuadratura de Gauss-Radau.

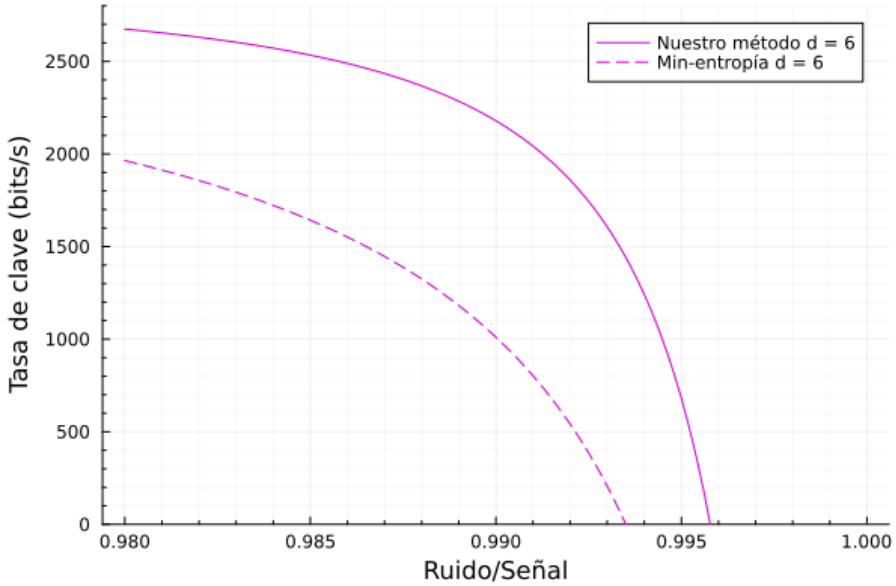

Figura 4.3: Tasa de clave (en bits por segundo) frente a la relación ruido/señal (adimensional) para el modelo de grados de libertad temporales en la dimensión $d = 6$. Los cálculos se realizaron utilizando $m = 5$ puntos de la cuadratura de Gauss-Radau. Dado que d no es una potencia de un número primo, se emplearon bases que son solo aproximadamente no sesgadas.

En primer lugar, al centrarnos en la comparación de ambas técnicas, observamos que nuestro método alcanza tasas de clave más altas que el de min-entropía. Esta diferencia se vuelve más significativa para valores elevados de la relación ruido/señal y para dimensiones superiores.

Por otro lado, para valores de ruido/señal bajos y moderados, trabajar con dimensiones más altas siempre resulta más beneficioso, ya que se obtienen tasas de clave más altas. Sin embargo, sorprendentemente, cuando el nivel de ruido es extremo, este comportamiento cambia. En la figura 4.2 se puede observar que la tasa de clave para $d = 2$ es la que presenta mayor resistencia al ruido. La reflexión sobre este aspecto se abordará en las conclusiones.

4.2. Implementación con grados de libertad espaciales

Se observa la misma diferencia entre los métodos considerados que en el modelo temporal.

Sin embargo, en este modelo, las dimensiones bajas no muestran mayor resistencia al ruido extremo. Esto se debe a que la dependencia de v respecto a d en (2.41) es más compleja: la presencia de $\frac{1}{d}$ en las exponenciales negativas amplifica el impacto del ruido en dimensiones bajas. En contraste, en el modelo temporal, d aparece solo linealmente en el denominador de v (2.22), lo que atenúa el efecto del ruido en bajas dimensiones y permite mayor resistencia en ese rango.

Además, la implementación del modelo espacial requiere d detectores independientes para cada parte (Alice y Bob), por lo que, a medida que la dimensión aumenta, se producen más coincidencias accidentales de dark counts dentro de una ventana temporal. Esto impone un límite, más allá del cual un aumento en d deja de ser ser beneficioso y se vuelve contraproducente. Se observa que para

valores grandes de d y pequeños de Δt, la visibilidad en (2.41) escala como:

$$v(d) \approx \frac{1}{1 + d^2 \Delta t \mu^A \mu^B \gamma^{-1}}. \tag{4.1}$$

Sin embargo, debido a que la complejidad computacional del problema de SDP aumenta rápidamente con d, este resultado teórico no se ha representado gráficamente.

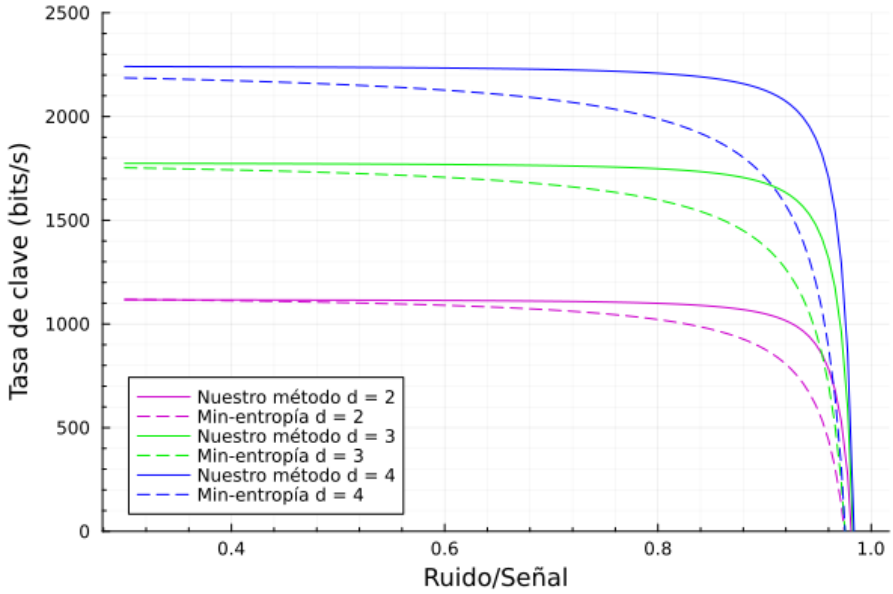

Figura 4.4: Tasa de clave (en bits por segundo) frente a la relación ruido/señal (adimensional) para el modelo de grados de libertad espaciales en las dimensiones $d = 2, 3, 4$. Los cálculos se realizaron utilizando $m = 6$ puntos de la cuadratura de Gauss-Radau.

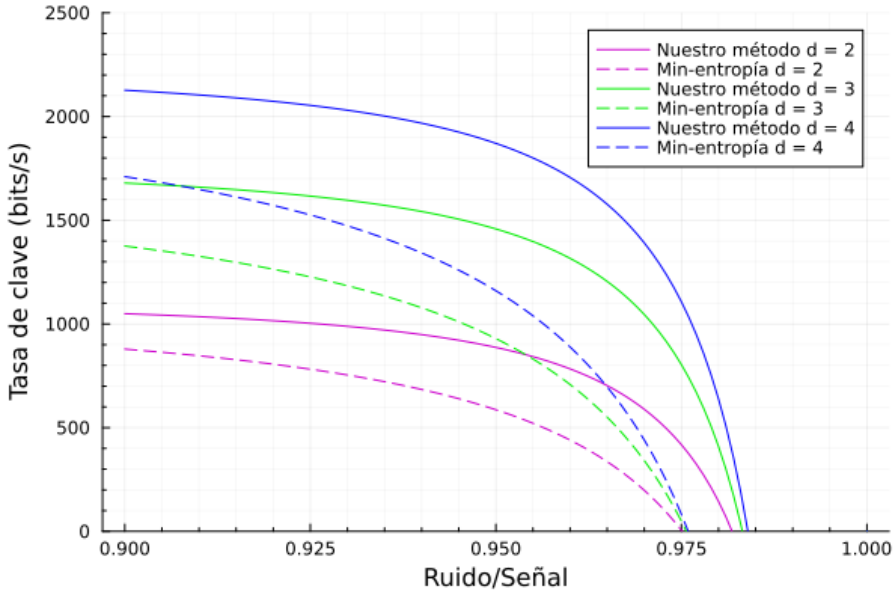

Figura 4.5: Tasa de clave (en bits por segundo) frente a la relación ruido/señal (adimensional) para el modelo de grados de libertad temporales en las dimensiones $d = 2, 3, 4$, centrado en el intervalo donde ocurre el cambio más significativo. Los cálculos se realizaron utilizando $m = 6$ puntos de la cuadratura de Gauss-Radau.

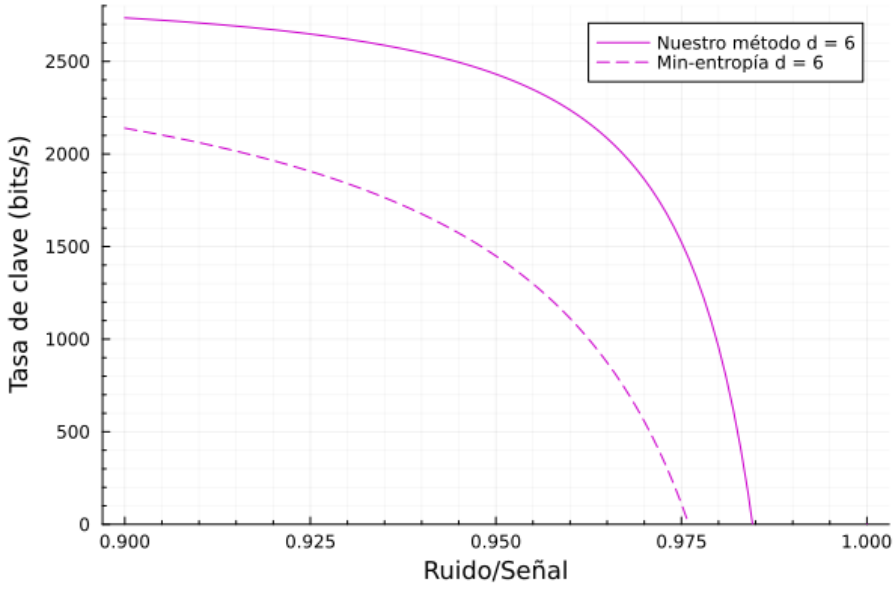

Figura 4.6: Tasa de clave (en bits por segundo) frente a la relación ruido/señal (adimensional) para el modelo de grados de libertad espaciales en la dimensión $d = 6$. Los cálculos se realizaron utilizando $m = 5$ puntos de la cuadratura de Gauss-Radau. Dado que d no es una potencia de un número primo, se emplearon bases que son solo aproximadamente no sesgadas.

Para permitir que cualquier persona pueda reproducir los resultados obtenidos, todo el código presentado en la sección 3.3, así como el utilizado para generar las gráficas (que no se incluye en este documento por considerarse de menor relevancia), está disponible en https://github.com/marfgut/key-rate.

Capítulo 5

Conclusiones

Este trabajo ha implementado satisfactoriamente, en el lenguaje de programación Julia, una jerarquía de programación semidefinida que converge a la entropía condicional de von Neumann para calcular la tasa de clave en distribución cuántica de claves, mostrando una mayor precisión frente al enfoque previo basado en la min-entropía.

Se han estudiado dos modelos de ruido que simulan adecuadamente las condiciones experimentales, considerando tanto el ruido ambiental como el introducido por los detectores, mostrando que los parámetros específicos de implementación influyen significativamente en las tasas de clave alcanzables. Estos modelos podrían ser refinados para incluir otras fuentes de ruido, como los eventos de *multiphotones*, donde la fuente emite más de un par de fotones entrelazados al mismo tiempo. Aunque dichos eventos se han despreciado debido a los pequeños valores de λF y $\lambda \Delta t$ tomados, en otros contextos podrían ser relevantes, reduciendo las tasas de clave alcanzables.

La mayor resistencia a ruidos extremos en dimensiones menores sugiere que podría ser una buena idea dividir el espacio total en subespacios de menor dimensión, lo que permitiría establecer una clave secreta incluso en condiciones experimentales extremadamente ruidosas. De hecho, en [21] ya se exploró esta idea, mostrando que, aunque los subespacios grandes presentan mayor entrelazamiento, los más pequeños requieren menos corrección de errores. Esto da lugar a una interacción compleja entre ruido, tasa y dimensión del subespacio, donde, en condiciones de ruido extremo, los subespacios más pequeños logran alcanzar la mayor, o incluso la única, tasa de clave.

Finalmente, aunque la jerarquía presentada converge teóricamente al valor exacto de la entropía condicional de von Neumann, cada nivel de la jerarquía solo proporciona una aproximación de la tasa de clave. Si se requiere una precisión alta, es necesario aumentar el número de m puntos de la cuadratura de Gauss-Radau, incrementando drásticamente la complejidad computacional. Por ello, se deben explorar herramientas de optimización más avanzadas para calcular tasas de clave en protocolos más complejos, con dimensiones mayores o bases de medición arbitrarias. En [26] se propone una mejora a través de un algoritmo que optimiza directamente sobre el cono de la entropía relativa, resolviendo el problema sin necesidad de descomponerlo en una secuencia de programas semidefinidos.

En conclusión, este trabajo demuestra cómo el desarrollo de técnicas numéricas sigue ampliando las capacidades de la QKD, un campo en constante evolución que abre nuevas oportunidades experimentales y teóricas.

Bibliografía

[1] G. S. Vernam, *Cipher printing telegraph systems: For secret wire and radio telegraphic communications*, Journal of the A.I.E.E. **45**, 109-115 (1926).

[2] N. Gisin, G. Ribordy, W. Tittel and H. Zbinden, *Quantum cryptography*, Reviews of Modern Physics **74**, 145195 (2002), arXiv:quant-ph/0101098.

[3] S. Wiesner, *Conjugate coding*, ACM SIGACT News **15**, 78-88 (1983).

[4] W. K. Wootters and W. H. Zurek, *A single quantum cannot be cloned*, Nature **299**, 802-803 (1982).

[5] P. W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 124-134 (1994).

[6] A. K. Ekert, *Quantum Cryptography Based on Bell's Theorem*, Phys. Rev. Lett **67**, 661-663 (1991).

[7] R. König, R. Renner and C. Schaffner, *The operational meaning of min- and max-entropy*, IEEE Transactions on Information Theory **55**, 4337-4347 (2009), arXiv:0807.1338 [quant-ph].

[8] P. Brown, H. Fawzi and O. Fawzi, *Device-independent lower bounds on the conditional von Neumann entropy*, (2021), arXiv:2106.13692 [quant-ph].

[9] M. Navascués, S. Pironio and A. Acín, *A convergent hierarchy of semidefinite programs characterizing the set of quantum correlations*, New Journal of Physics **10**, 073013 (2008), arXiv:0803.4290 [quant-ph].

[10] M. Araújo, M. Huber, M. Navascués, M. Pivoluska and A. Tavakoli, *Quantum key distribution rates from semidefinite programming*, Quantum **7**, 1019 (2023), arXiv:2211.05725 [quant-ph].

[11] Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, Cambridge: Cambridge University Press (2010).

[12] E. Rieffel and W.H. Polak, *Quantum computing: A gentle introduction*, The MIT Press (2011).

[13] C. H. Bennett and G. Brassard, *Quantum cryptography: public key distribution and coin tossing*, Theoretical Computer Science **560**, (reprint), 7-11 (1984).

[14] H.-K. Lo, H. F. Chau and M. Ardehali, *Efficient Quantum Key Distribution Scheme and a Proof of Its Unconditional Security*, Journal of Cryptology **18**, 133-165 (2005),

arXiv:quant-ph/0011056.

[15] D. Bruß, *Optimal Eavesdropping in Quantum Cryptography with Six States*, Physical Review Letters **81**, 3018-3021 (1998), arXiv:quant-ph/9805019.

[16] L. Sheridan and V. Scarani, *Security proof for quantum key distribution using qudit systems*, Physical Review A **82**, 030301(R) (2010), arXiv:1003.5464 [quant-ph].

[17] W. K. Wootters and B. D. Fields, *Optimal state-determination by mutually unbiased measurements*, Annals of Physics **191**, 363-381 (1989).

[18] I. Bengtsson, W. Bruzda, Å. Ericsson, J.-Å. Larsson, W. Tadej and K. *Życzkowski*, *Mutually unbiased bases and Hadamard matrices of order six*, Journal of Mathematical Physics **48**, 052106 (2007), arXiv:quant-ph/0610161.

[19] I. Bengtsson, *Three ways to look at mutually unbiased bases*, AIP Conference Proceedings **889**, 40-51 (2007), arXiv:quant-ph/0610216.

[20] I. Devetak and A. Winter, *Distillation of secret key and entanglement from quantum states*, Proceedings of the Royal Society of London Series A **461**, 207-235 (2005), arXiv:quantph/0306078.

[21] M. Doda, M. Huber, G. Murta, M. Pivoluska, M. Plesch and C. Vlachou, *Quantum key distribution overcoming extreme noise: simultaneous subspace coding using high-dimensional entanglement*, Physical Review Applied **15**, 034003 (2021), arXiv:2004.12824 [quant-ph].

[22] A. Tavakoli, A. Pozas-Kerstjens, P. Brown and M. Araújo, *Semidefinite programming relaxations for quantum correlations*, arXiv:2307.02551 [quant-ph].

[23] M. Navascués, G. de la Torre and T. Vértesi, *Characterization of Quantum Correlations with Local Dimension Constraints and Its Device-Independent Applications*, Physical Review X **4**, 011011 (2014), arXiv:1308.3410 [quant-ph].

[24] M. Lubin, O. Dowson, J. Garcia, J. Huchette, B. Legat and J. Vielma, *JuMP 1.0: Recent improvements to a modeling language for mathematical optimization*, arXiv:2206.03866 [cs.PL], JuMP Documentation.

[25] G. H. Golub, *Some modified matrix eigenvalue problems*, SIAM Review **15**, 318-334 (1973).

[26] A. G. Lorente, P. V. Parellada, M. Castillo-Celeita and M. Araújo, *Quantum key distribution rates from non-symmetric conic optimization*, arXiv:2407.00152 [quant-ph].

Anexo A

Cálculo de la tasa de clave con el método de min-entropía

Este método consiste en acotar inferiormente la entropía relativa de von Neumann con la min-entropía condicional:

$$H_{min}(A|E) = -\log_2 \left(\frac{\left(\sqrt{vd+1-v} + (d-1)\sqrt{1-v}\right)^2}{d^2} \right). \tag{A.1}$$

Esta expresión ha sido extraída de la referencia [21] (ecuación (3)).

Así, la tasa de clave, antes de multiplicarse por la tasa de producción de rondas válidas de los modelos de ruido, resulta en:

$$K \geq H_{min}(A|E) - H(A|B). \tag{A.2}$$