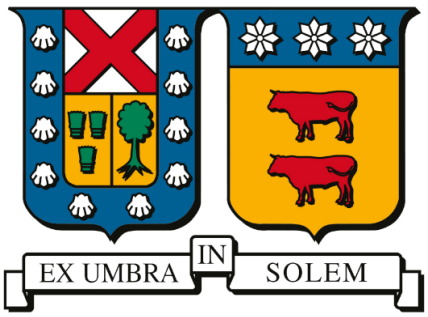
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



CRIPTOANÁLISIS DE PROTOCOLOS DE DISTRIBUCIÓN CUÁNTICA DE LLAVES BASADOS EN LA RECONCILIACIÓN POR FRAMES.

DANIEL LEONARDO ESPINOZA FIGUEROA

TESIS DE GRADO PARA OPTAR POR EL TÍTULO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA ELECTRÓNICA, MENCIÓN TELEMÁTICA, E INGENIERÍA CIVIL TELEMÁTICA

DIRECTOR DE TESIS	:	SR. NICOLÁS JARA
CO-DIRECTOR DE TESIS	:	SR. LUIS LIZAMA

OCTUBRE 2024

A mi familia . . .

Gracias Loreto, por ser la madre perfecta.
Gracias Alonso, por ser el hermano perfecto.

El presente trabajo de Tesis va dedicado a ustedes, quienes me han dado las herramientas para contribuir al mundo.

Agradecimientos

Originalmente, este trabajo se iba a enfocar en el diseño de un nuevo protocolo de distribución cuántica de llaves criptográficas basado en *frames*, utilizando los fenómenos del entrelazamiento cuántico. Sin embargo, el enfoque cambió abruptamente, cuando me di cuenta de que el protocolo que estaba utilizando de base (L23) tenía una vulnerabilidad crítica, donde la llave compartida podía recuperarse de manera trivial. Intenté continuar con ello, probando diversas maneras de corregir la vulnerabilidad, sin éxito.

Luis Lizama, mi Codirector de Tesis, rápidamente había llegado a una posible corrección de la vulnerabilidad, a través de su gran experiencia como Criptógrafo. Así, pensé en probar un camino diferente para corregir la vulnerabilidad, sin éxito.

Fuera de lo académico, desde 2021 que comencé a investigar sobre Criptografía de manera autónoma, donde lo que más me llamaba la atención no era el diseño de protocolos en sí, sino más bien cómo comprometerlos. Gracias a esos 3 años pude encontrar la vulnerabilidad de L23, por lo que decidí dar un giro a mi trabajo de Tesis, enfocándolo al desarrollo de directrices de seguridad para que los prorocolos de distribución cuántica de llaves criptográficas basados en frames sean más seguros. Aprovechando mis conocimientos en Criptoanálisis, pude romper todos los protocolos propuestos hasta la fecha.

Quisiera agradecer enormemente a Luis Lizama, ya que su apoyo constante me permitió mejorar enormemente como Criptoanalista, donde él confío plenamente en mí y mis conocimientos. Gracias a él es que estoy realizando este trabajo de Tesis.

Agradezco profundamente a Nicolás Jara, mi Director de Tesis, por su enorme confianza que ha puesto en mí. Desde que entré al postgrado, en el segundo semestre del 2022, Nicolás ha estado acompañándome en cada propuesta que he tenido, pasando de redes cuánticas a criptografía cuántica y luego criptoanálisis, donde a pesar de que sus conocimientos específicos difieren bastante de estos temas, confió plenamente en mí.

Muchas gracias a todas las personas que me apoyaron durante mi estadía en la Universidad Técnica Federico Santa María, estoy muy contento de haber seguido lo que realmente me apasiona, sin importar lo difícil que fue.

Resumen

La criptografía comprende el estudio del diseño de algoritmos que permitan asegurar la confidencialidad, integridad y autenticidad de la información. El criptoanálisis, en cambio, es el estudio de la seguridad de estos algoritmos, con el objetivo de otorgar una mejora continua de acuerdo a las amenazas actuales. La computación cuántica representa una de estas amenazas, impactando directamente a la seguridad de todo Internet. Así pues, criptógrafos de todo el mundo han propuesto nuevos algoritmos que se basan en problemas matemáticos más robustos, o en la propia naturaleza de la realidad que nos rodea.

La física cuántica puede ser utilizada para intercambiar una llave criptográfica, concepto conocido como Distribución Cuántica de Llaves o *Quantum Key Distribution* (QKD). Desde 1984, diversos protocolos han sido propuestos para ser resilientes frente a las limitaciones que esta tecnología posee, relacionadas con la eficiencia en la corrección de errores y su seguridad, ya que la QKD requiere de dos canales: el cuántico (información cuántica) y el clásico (información clásica).

Los protocolos QKD basados en *frames* son un paradigma relativamente nuevo que pretende superar los principales retos de esta tecnología, como son los conocidos ataques cuánticos *Photon Number Splitting* (PNS) e *Intercept-Resend* (IR), además de la eficiencia en la corrección de errores mediante nuevos métodos integrados a cada protocolo. Este paradigma utiliza frames, es decir, matrices que agrupan pares de estados cuánticos con el objetivo de aumentar, en orden polinomial, el tamaño de la clave compartida entre Alice (transmisor) y Bob (receptor), además de construir métodos de corrección de errores más eficientes.

La seguridad en el canal clásico de los protocolos QKD basados en frames no se ha estudiado adecuadamente, ya que el aumento del tamaño de la llave compartida implica la reutilización de pares de estados cuánticos, un aspecto que aún no se ha tenido en cuenta del punto de vista del criptoanálisis. En el presente trabajo de Tesis, se definen las primeras directrices de seguridad para el desarrollo seguro de protocolos QKD basados en frames mediante el criptoanálisis clásico sobre todos los protocolos publicados en conferencias o revistas que han seguido esta línea de investigación, donde vulnerabilidades críticas fueron encontradas. Así, es posible romper la seguridad de estos esquemas, recuperando llaves privadas utilizadas para cifrar o firmar las comunicaciones, demostrando la viabilidad mediante resultados técnicos y experimentales. El proceso de criptoanálisis clásico concluye con directrices de seguridad para seguir avanzando en la línea de investigación que pretende utilizar frames para intercambiar una clave criptográfica de forma segura, utilizando la comunicación cuántica.

Abstract

Cryptography involves the study of designing algorithms that ensure confidentiality, integrity, and information authenticity. On the other hand, cryptanalysis is the study of assessing the security of these algorithms to enhance their resilience to current threats. The emergence of quantum computing poses a significant threat to the security of the Internet, leading cryptographers worldwide to propose new algorithms based on more robust mathematical problems or the fundamental principles of nature.

The principles of quantum physics can be used to exchange a cryptographic key, a concept known as Quantum Key Distribution (QKD). Since 1984, several protocols have been proposed to be resilient to the limitations of this technology, related to error correction efficiency and security, since QKD requires two channels: quantum (quantum information) and classical (classical information).

Frame-based QKD protocols are a relatively new paradigm that aims to overcome the main challenges of this technology, such as the well-known quantum attacks Photon Number Splitting (PNS) and Intercept-Resend (IR), as well as the efficiency in error correction through new methods integrated into each protocol. This paradigm uses frames, i.e., matrices that group pairs of quantum states to increase, in polynomial order, the size of the shared key between Alice (transmitter) and Bob (receiver), as well as constructing more efficient error correction methods.

The security in the classical channel of frame-based QKD protocols has not been properly studied, since the increase of the shared key size involves reusing pairs of quantum states, an aspect that has not yet been considered from the cryptanalysis perspective. In the present Thesis work, we define the first security guidelines for the secure development of frame-based QKD protocols using classical cryptanalysis on all protocols published in conferences or journals that have followed this path, where critical vulnerabilities were found. Thus, it is possible to break the security of these schemes, recovering private keys used to encrypt or sign communications, demonstrating the feasibility through technical and experimental results. The classical cryptanalysis process concludes with security guidelines to continue advancing the line of research that aims to use frames to exchange a cryptographic key securely, using quantum communication.

Keywords. Quantum Key Distribution, Frame-based Reconciliation, Cryptanalysis, Key Recovery, Security Guidelines

Índice de Contenidos

Índice de Tablas

Índice de Figuras

1 | Introducción

Criptografía se refiere a la ciencia que estudia la escritura de secretos con el objetivo de esconder el significado de un mensaje [1]. Esta disciplina es indispensable para todas las aplicaciones digitales ya que provee:

- Confidencialidad: Sólo las entidades emisoras y receptoras pueden conocer el contenido original del mensaje. Para ello, el mensaje se esconde en tránsito a través de su versión encriptada, la cual se realiza mediante una llave (número o cadena de caracteres) que sólo conocen los emisores y los receptores.
- Integridad: La información no debe ser modificada en tránsito. Funciones no biyectivas son utilizadas para estos efectos.
- Autenticación: La información debe ser auténtica, reconociendo a la entidad emisora que la envió. Para ello, se utilizan llaves privadas que permiten entregar una firma de los mensajes.

La criptografía cubre la confidencialidad a través del cifrado simétrico (llaves equivalentes para cifrar y descifrar) y asimétrico (llaves diferentes para cifrar y descifrar), utilizando esquemas estándar como el *Advanced Encryption Standard* (AES) [2] y *Rivest-Shamir-Adleman* (RSA) [3], respectivamente. Para la integridad se utilizan las funciones hash, funciones no biyectivas con el objetivo de demostrar que la información fue recibida de manera íntegra, utilizando estándares como el *Secure Hash Algorithm* (SHA) [4] y los *Message Authentication Codes* (MAC) [5]. La autenticación utiliza las propiedades de la integridad para adicionalmente demostrar que la entidad emisora es quien dice ser, a través de estándares como los *JSON Web Tokens* (JWTs) [6] o incluso RSA.

Consideremos que dos partes, llamadas Alice y Bob, se quieren comunicar de forma segura. Es bien sabido que el cifrado simétrico es más eficiente de computar que el cifrado asimétrico en términos de memoria y procesamiento [7], por lo que generalmente ambas partes requieren de una llave criptográfica común para cifrar y descifrar mensajes secretos. Para ello, es necesario que ambas partes realicen un intercambio de llaves, utilizando estándares como *Diffie-Hellman* (DH) [8] o su versión en curvas elípticas [9], ilustrada en la Figura 1.1. En un contexto real, el protocolo *Transport Layer Security* (TLS) [10] permite realizar una comunicación HTTP segura, siendo Alice el cliente que consume el contenido de un servidor, Bob, donde previamente no existía una llave criptográfica común para que la comunicación fuera confidencial. El intercambio de llaves permite que Alice y Bob puedan encriptar y desencriptar los paquetes HTTP que se envían a través de la red. Por otro lado, el uso de firmas digitales permite que Alice reconozca a Bob, teniendo conocimiento de que existe una firma digital a nombre de una entidad certificadora confiable que, valga la redundancia, certifica que el servicio a consumir existe y se encuentra a nombre de Bob.

La seguridad de los protocolos criptográficos depende del problema matemático subyacente que los construyó. Por ejemplo, la seguridad de RSA se basa en el problema de la factorización de números enteros (*Integer Factorization Problem*):

El problema de la factorización de números enteros:
Sea $n \in \mathbb{Z}$ un número entero que se compone de la multiplicación de dos números primos p y q, es decir, $n = p \cdot q$. Entonces, encontrar p y q conociendo únicamente n es un problema NP.

RSA es un protocolo que se construye utilizando el problema de la factorización de números enteros como base, por lo que n, p y q son parámetros en una instancia RSA. Por estándar, n debe ser un número natural de aproximadamente 3078 bits mínimo, por lo que p y q deben ser números naturales de aproximadamente 1539 bits [11].

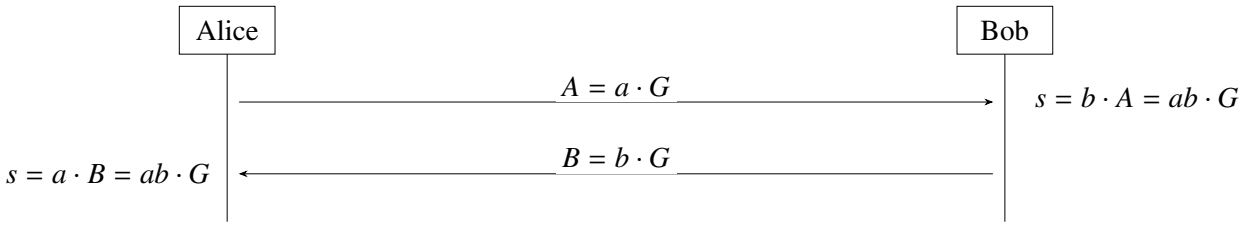


Figura 1.1: Diagrama de *Elliptic Curve Diffie-Hellman* (ECDH), un protocolo de intercambio de llaves basado en curvas elípticas. Las llaves privadas $a, b \in \{2, \ldots, |E|\}$ se generan de manera aleatoria, donde E es una curva elíptica conocida públicamente.

RSA provee autenticación de la siguiente manera: un mensaje $m \in \mathbb{Z}_n$ es firmado a través del cómputo de $c \in \mathbb{Z}_n$, definido como

$$c \equiv m^d \bmod n. \tag{1.1}$$

Se define $d \in \mathbb{Z}_n$ como la llave privada o exponente privado de la instancia RSA. Es importante notar que la llave privada identifica inequívocamente al emisor de la firma, y cualquier entidad puede validar su integridad a través del cómpúto de $m' \in \mathbb{Z}_n$:

$$m' \equiv c^e \bmod n. \tag{1.2}$$

Si $m' = m$, entonces la firma c es válida. En caso contrario, el mensaje m puede ser descartado ya que se concluye que fue modificado en tránsito. Cualquier entidad puede validar la integridad de la firma digital ya que $e \in \mathbb{Z}_n$ corresponde a la llave pública o exponente público, donde n también es un valor que se conoce públicamente, denominado como "módulo público".

La construcción de e y d no es arbitraria, deben tener una relación matemática. En concreto, uno es el inverso multiplicativo del otro:

$$e \equiv d^{-1} \bmod \phi(n), \tag{1.3}$$

donde $\phi(n) = (p-1) \cdot (q-1)$. Un aspecto importante de la generación de las llaves es que sólo la entidad emisora de las firmas digitales conoce la factorización de n, es decir, p y q, por lo que además conoce $\phi(n)$. Lo anterior implica que la única manera de obtener d a través de e, siendo un atacante o una entidad diferente a la emisora, es resolviendo el problema de la factorización de números enteros. Resolver este problema implicaría la posibilidad de usurpar la identidad de un servidor, en el caso de TLS.

Otro problema matemático que nos interesa es el problema del logaritmo discreto en curvas elípticas (*Elliptic Curve Discrete Logarithm Problem*), utilizado para construir ECDH. Las curvas elípticas que se utilizan por estándar deben tener una cantidad considerable de puntos, aproximadamente del orden de 256 bits [11].

El problema del logaritmo discreto en curvas elípticas:
Dada una curva elíptica E sobre un campo finito $\mathbb{F}_p$ y un punto $Q = d \cdot P$, donde P es un generador de E. Entonces, encontrar $d \in \mathbb{F}_p$ conociendo P y Q es un problema NP.

Como podemos apreciar, la criptografía es un pilar que sostiene firmemente la seguridad de millones de aplicaciones que se desarrollan hoy en día. Sin embargo, es importante que esta ciencia se encuentre en constante actualización, más aún cuando nuevas tecnologías se hacen presentes.

1.1. La era de la computación y comunicación cuántica

A inicios de los años 80, un nuevo paradigma tecnológico fue propuesto, tomando como referencia el concepto de una máquina de Turing, pero aplicándolo a la mecánica cuántica [12]. Dos décadas después, esta nueva tecnología se encuentra actualmente revolucionando la computación tal y como la conocemos hoy en día, con la creación de los computadores cuánticos [13].

La computación cuántica difiere respecto a la clásica ya que utiliza bits cuánticos como unidad básica de procesamiento. Cada bit cuántico, cúbit o qubit tiene, en esencia, infinitos estados posibles, ya que la superposición cuántica permite asignar una probabilidad de que el estado, al ser medido, colapse a un bit 0 o 1 clásico [14].

En la computación clásica, un bit corresponde a una diferencia de potencial, definida de manera determinística y compuesta de dos estados posibles, 0 o 1. En cambio, un bit cuántico representa, por ejemplo, el espín de una partícula o un fotón polarizado [15]. A modo general, un bit cuántico $|\phi\rangle$ se modela de la siguiente manera:

$$|\phi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle \ , \ \alpha, \beta \in \mathbb{C}. \tag{1.4}$$

En todo momento se cumple que $\alpha^2 + \beta^2 = 1$, donde α^2 corresponde a la probabilidad de que el estado cuántico o bit cuántico $|\phi\rangle$ colapse al estado $|0\rangle$ al ser medido, y β^2 la probabilidad de colapsar al estado $|1\rangle$. En el caso de que $\alpha, \beta \neq 0$, hablamos de que $|\phi\rangle$ se encuentra en una superposición de estados, lo que demuestra que un qubit posee infinitos estados posibles, dado los posibles valores de α y β. Sin embargo, la medición de un estado cuántico lo hace colapsar a un resultado binario, correspondiente a $|0\rangle$ o $|1\rangle$, o bien, un 0 o 1 clásico, y es que nos referiremos al concepto de medir como a realizar una medición proyectiva en dos posibles bases, cuestión que detallaremos más adelante.

Las condiciones que debe cumplir todo estado cuántico determina que cada uno de ellos corresponde a un vector unitario en $\mathbb{R}^3$, formándose la denominada esfera de Bloch, ilustrada en la Figura 1.2. Por ello, las compuertas cuánticas no son más que diferentes rotaciones en esta esfera, por lo que podemos construir las compuertas NOT, OR, AND y XOR que conocemos de la computación clásica, para desarrollar algoritmos cuánticos.

El propósito de un computador cuántico no es reemplazar al computador clásico, sino utilizarlo en ciertos escenarios donde sea significativamente más eficiente, como en la química cuántica [16], que comprende el estudio de sistemas químicos, a nivel atómico y molecular, para diversas condiciones en sus reacciones químicas. En cuanto a lo que es de nuestro interés, la computación cuántica supone un peligro inminente al problema de la factorización de números enteros y el problema del logaritmo discreto, ya que Peter Shor, en 1994, propuso un algoritmo que permite romper RSA y ECDH en tiempo polinomial, para el momento en que exista un computador cuántico con la cantidad de qubits suficientes para ejecutarlo en llaves que cumplan con los estándares actuales [17].

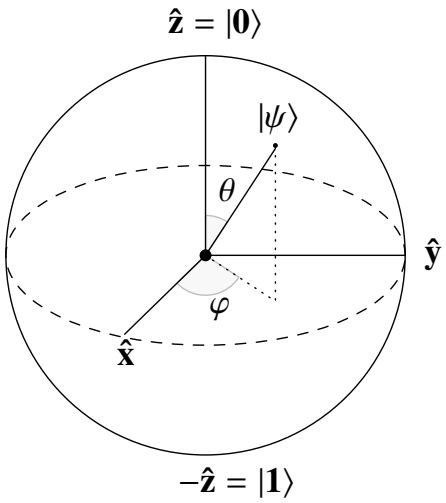


Figura 1.2: Esfera de Bloch

Dada la amenaza inminente a la seguridad de millones de aplicaciones existentes hoy en día, el Instituto Nacional de Estándares y Tecnologías de Estados Unidos realizó un llamado global en 2017 para recibir propuestas de estandarización de nuevos esquemas criptográficos en el marco de *Post-Quantum Cryptography* (PQC), utilizando de base problemas matemáticos que, hasta el día de hoy, siguen siendo NP.

Las aplicaciones industriales de la criptografía siempre se habían centrado en el uso de esquemas donde su seguridad se sustentaba en problemas matemáticos del tipo NP. Sin embargo, no es hasta el año 2007 donde, por primera vez, se utilizó la física cuántica para asegurar las comunicaciones [18]. *ID Quantique*, una empresa de Ginebra, Suiza, desarrolló un esquema criptográfico híbrido (como lo es TLS) para sus elecciones, el cual se sigue utilizando hasta el día de hoy. La empresa suiza, en su solución llamada *Cerberis*, utiliza la física cuántica para intercambiar una llave criptográfica entre dos partes, donde posteriormente se encriptan las comunicaciones de manera *clásica*, con AES.

Utilizar física cuántica para intercambiar una llave criptográfica de forma segura no es un concepto nuevo, al contrario, es una propuesta que se ha estado investigando desde 1984 [19], conocida como *Quantum Key Distribution* (QKD). En ella, un transmisor (Alice) envía estados cuánticos $|\phi\rangle$ que son recibidos y medidos por un receptor (Bob), proceso descrito en el Capítulo 2. Considerando que los estados cuánticos se representan a través de vectores con cierta orientación en la esfera de Bloch, entonces el envío de estos puede realizarse mediante un haz de fotones de baja energía, polarizado en aquella orientación [20]. Es importante notar que los fenómenos de la física cuántica se hacen presentes únicamente cuando la energía se encuentra cuantizada, razón por la cual se requieren fotones de baja energía. Por lo tanto, en QKD, no necesitamos una computadora cuántica, ya que el procesamiento de los estados cuánticos no es requerido.

La seguridad de la QKD no se basa en un problema matemático, sino en la propia naturaleza de la física cuántica, o más general, en las leyes de la física. Sin embargo, la implementación de la QKD introduce una serie de desafíos, ya que al haber una restricción física para que los fenómenos cuánticos se hagan presentes, la atenuación del medio pasa a ser un factor mucho más relevante que en las tecnologías convencionales. Un resumen de las ventajas y desventajas más relevantes de la QKD se describe en la Tabla 1.1.

Tabla 1.1: Ventajas y desventajas de la Distribución Cuántica de Llaves, o bien, *Quantum Key Distribution* (QKD) [21].

Ventajas	Desventajas
El colapso de un estado cuántico a un bit clásico se comporta como un *True Random Number Generator* (TRNG)	Se requieren mayores costos de implementación, agregando repetidores y un canal dedicado a la comunicación cuántica
Es imposible copiar un estado cuántico	La distancia máxima de transmisión se ve reducida, dada la baja potencia del haz de fotones
QKD ofrece una comunicación *future-proof*, es decir, no se puede comprometer la seguridad luego de haber finalizado la comunicación	Errores en el canal deben ser detectados y corregidos. Las fuentes del error son: codificación, depolarización y medición

Para la corrección de errores, se ha propuesto utilizar técnicas que vienen de la computación clásica, como lo son la Comprobación de Paridad de Baja Densidad o *Low Density Parity Check* (LDPC), además de los Turbo Códigos o *Turbo Codes* [22]. Modificaciones a estos métodos se han propuesto [23], pero ninguno ha sido capaz de detectar y corregir todos los errores que pudiesen ocurrir en un canal no confiable y ruidoso, debido al Límite de Shannon.

Para finalizar, es importante recalcar que la QKD sólo permite intercambiar una llave criptográfica de forma segura, por lo que PQC tiene un rol crucial en los nuevos estándares que serán publicados en los años venideros. Así pues, QKD y PQC no deben tratarse como soluciones independientes, sino como herramientas de un mismo criptosistema que permiten asegurar las comunicaciones actuales y las futuras.

1.2. Descripción del problema

1.2.1. La nueva revolución de las comunicaciones cuánticas

Durante el año 2020, un nuevo paradigma de protocolos QKD fue propuesto, con el objetivo de ser resistentes a ciertos ataques que implican una interacción física con el canal dedicado a la comunicación cuántica, llamados *Photon Number Splitting* (PNS) [24] e *Intercept-Resend* (IR) [25]. Además, este nuevo paradigma, el cual lleva 4 protocolos propuestos al día de hoy, pretende resolver el desafío de la corrección de errores, a través de métodos integrados a cada protocolo, sin utilizar información redundante y mejorando su eficiencia, a comparación con los métodos anteriormente mencionados [26].

Las nuevas propuestas utilizan estructuras matemáticas llamadas *frames*, que corresponden a matrices n x m que agrupan pares de estados cuánticos no ortogonales, descritos en el Capítulo 2. Con los frames, es posible aumentar, como mínimo, cuadraticamente los bits de la llave secreta que es intercambiada entre Alice (transmisor) y Bob (receptor).

El uso de frames en la QKD, al ser una propuesta nueva, es importante que madure, a través de la revisión constante de la seguridad de los diferentes protocolos que se postulen en la academia. Actualmente, se ha declarado que sólo una de las cuatro propuestas es insegura [27].

1.2.2. El descuido del criptoanálisis clásico

El criptoanálisis, en comparación con la criptografía, abarca el estudio de esquemas criptográficos con el objetivo de encontrar posibles debilidades que rompan su seguridad, recuperando llaves privadas a través de llaves públicas, descifrando mensajes encriptados sin la necesidad de la llave privada, entre otros métodos o vectores de ataque. En los protocolos QKD basados en frames, se mencionan aspectos de seguridad que pretenden demostrar robustez al criptoanálisis cuántico, de acuerdo a los ataques PNS y IR. Sin embargo, el criptoanálisis clásico es un aspecto que no ha sido tratado con el debido detalle, ya que la reutilización de pares es una decisión de diseño que no se ha considerado del punto de vista de la seguridad de los protocolos.

Como se procederá a describir en el Capítulo 2, la QKD opera en dos canales: el cuántico y el clásico. En el cuántico, información cuántica es transmitida, es decir, es un canal dedicado al envío de haces de fotones de baja energía, para que los fenómenos de la física cuántica se hagan presentes. El canal clásico, en cambio, es requerido para el envío de información clásica que manejamos día a día. En la QKD, la información clásica se utiliza para enviar resultados de medición, confirmaciones, información adicional para el protocolo, entre otros datos. Así pues, el criptoanálisis debe ser un estudio que abarque ambos canales, ya que los vectores de ataque podrían variar significativamente y, dado que la QKD pretende sobrellevar la amenaza de la computación cuántica, es fundamental el criptoanálisis clásico.

1.3. Contribuciones del trabajo de tesis

Las contribuciones del trabajo de tesis se describen a continuación, considerando que los problemas identificados corresponden a la necesidad de madurez en los protocolos QKD basados en frames, respecto al ejercicio del criptoanálisis clásico:

- Demostrar, a través de la formulación técnica y simulación computacional de los protocolos QKD basados en frames, que las cuatro propuestas publicadas hasta el día de hoy son vulnerables al criptoanálisis clásico, permitiendo recuperar la llave criptográfica que Alice (transmisor) y Bob (receptor), intentan intercambiar de manera segura.
- Definir las primeras directrices de seguridad para los protocolos QKD basados en frames, con el objetivo de construir una referencia que permita desarrollar protocolos más seguros.

1.4. Organización del escrito

Los capítulos restantes del escrito se describen a continuación, tomando en cuenta el estado del arte de los protocolos QKD tradicionales y los basados en frames, además de las contribuciones del trabajo de tesis:

- *Capítulo 2: Protocolos de distribución cuántica de llaves criptográficas*: Se define formalmente la solución que permite, tanto a Alice (transmisor) como a Bob (receptor), intercambiar una llave criptográfica de manera segura, a través de la física cuántica. conocida como *Quantum Key Distribution* (QKD). Con un contexto de la estructura básica de la QKD, se introducen los dos primeros protocolos que revolucionaron las comunicaciones tal y como las conocemos hoy en día: BB84 y BBM92. Luego, se introduce el concepto de frame, y cómo este se utilizó para desarrollar un nuevo paradigma de la QKD, describiendo las cuatro propuestas de protocolos basados en frames: LL20, LL21, LLS21 y L23.
- *Capítulo 3: Criptoanálisis de la reconciliación por frames*: Contextualizando el método de estudio para la realización del ejercicio del criptoanálisis, se describen tres vulnerabilidades encontradas que permiten recuperar, en su totalidad, la llave compartida: *Pairs reuse*, *Conjugate-Pairs reuse* y *Avalanche-Effect*. Cada ataque es descrito de manera teórica, mencionando el impacto que existe en cada protocolo, para luego adjuntar los resultados de la simulación computacional que demuestran su factibilidad, si aplica.
- *Capítulo 4: Directrices de Seguridad*: De acuerdo a las vulnerabilidades descritas en el Capítulo 3, directrices de seguridad son descritas con el objetivo de construir protocolos QKD basados en frames que sean más seguros en el futuro. Las directrices abarcan factores de diseño como la derivación de la llave compartida y la reutilización de los pares de estados cuánticos para el cómputo de frames, considerando estándares de seguridad actuales.
- *Capítulo 5: Conclusiones*: El trabajo de tesis es concluido, describiendo las contribuciones y la importancia del criptoanálisis como una manera de construir protocolos más seguros para el presente y el futuro de las comunicaciones. Discusiones finales son presentadas al lector, comentando trabajos futuros y resumiendo los resultados obtenidos.

2 | Protocolos de distribución cuántica de llaves criptográficas

La QKD (*Quantum Key Distribution*) es un sistema que permite intercambiar llaves criptográficas de forma segura a través de la comunicación por dos canales: el clásico y el cuántico. Nos referimos al canal clásico como el medio por el cual se transmite información clásica, utilizando tecnologías convencionales. Por otro lado, el canal cuántico se refiere al medio por el cual enviamos información cuántica, es decir, energía que se encuentre cuantizada para que los fenómenos de la física cuántica se hagan presentes. Para ello, nuevas tecnologías son requeridas, tanto para transmitir como para recibir estados cuánticos. Es importante recordar que no es necesario un computador cuántico para implementar la QKD, ya que no se tiene la necesidad de procesar, sino únicamente generar los estados cuánticos, enviarlos por el canal cuántico y recibirlos.

Todo protocolo QKD entre dos partes: Alice (el transmisor) y Bob (el receptor), se divide en 6 etapas fundamentales:

1. **Preparación**: Alice prepara n estados cuánticos $|\phi\rangle$ a enviar por el canal cuántico. Es importante notar que estos estados no son almacenados, sino únicamente generados y enviados uno a uno, esta etapa permite describir la estructura que cada estado cuántico debe tener. El conjunto de n estados cuánticos $|\phi\rangle$ lo denotamos como $|\phi\rangle^{\oplus n}$.
2. **Envío**: Alice envía $|\phi\rangle^{\oplus n}$ por el canal cuántico.
3. **Medición**: Bob mide los estados cuánticos recibidos por Alice, dentro de un conjunto de bases de medición.
4. ***Sifting***: Bob envía información que le permite a Alice derivar la llave compartida. La derivación de la llave del lado de Bob generalmente se realiza en esta fase, y el envío de dicha información se realiza a través del canal clásico.
5. **Reconciliación**: Alice y Bob intercambian información clásica que les permite detectar y corregir los posibles errores del canal cuántico.
6. **Amplificación de privacidad**: Luego de que Alice y Bob intercambien una llave criptográfica exitosamente, ambos realizan operaciones públicas que permiten reducir su tamaño, considerando que un atacante pudiese saber cierta cantidad de la llave compartida inicial.

La medición de un estado cuántico funciona mediante la elección de una base, comúnmente escogida dentro de dos opciones: Z (estándar) y X (*Hadamard*). En la base estándar, se realiza una medición proyectiva en el eje z de la esfera de Bloch, y el eje x se mide utlizando la base de *Hadamard*. Se sabe que los estados $|0\rangle$ y $|1\rangle$ se encuentran en el eje z, por lo que al utilizar la base Z, ambos estados cuánticos colapsan a sus representaciones binarias con un 100 % de probabilidad, es decir:

$$|0\rangle \xrightarrow{Z} \text{'0'},$$

$$|1\rangle \xrightarrow{Z} \text{'1'}.$$

Si se utiliza la base X, entonces el resultado es indeterminable, es decir, cada bit tiene un 50 % de probabilidad de aparecer:

$$|0\rangle \xrightarrow{X} \{0, 1\},$$

$$|1\rangle \xrightarrow{X} \{0, 1\}.$$

Para los estados $|+\rangle$ y $|-\rangle$ definidos como

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \tag{2.1}$$

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \tag{2.2}$$

el resultado es equivalente, con las bases de medición intercambiadas:

$$|+\rangle \xrightarrow{Z} \{0, 1\},$$

$$|-\rangle \xrightarrow{Z} \{0, 1\},$$

$$|+\rangle \xrightarrow{X} \text{'0'},$$

$$|-\rangle \xrightarrow{X} \text{'1'}.$$

El hecho de que el bit resultante sea indeterminable si la base de medición no coincide con la orientación del estado cuántico corrobora una de las ventajas de la QKD, que es la aleatoriedad del fenómeno cuántico, modelado como un *True Random Number Generator* (TRNG). Nótese que, del punto de vista de Bob, no es posible saber si el resultado colapsó con un 100 % o 50 % de probabilidad, sólo Alice puede saber esa información, ya que ella conoce los estados cuánticos que envía.

2.1. Los que dieron el primer paso: *Bennet & Brassard* (1984)

Charles Bennet y Gilles Brassard, en 1984, desarrollaron el primer protocolo QKD [19], el cual propone utilizar 4 estados cuánticos posibles: $|0\rangle, |1\rangle, |+\rangle$ y $|-\rangle$. Los estados $|0\rangle$ y $|1\rangle$ se ubican en el eje z de la esfera de Bloch, mientras que $|+\rangle$ y $|-\rangle$ se encuentran en el eje x. A continuación, el protocolo BB84 es descrito mediante una serie de pasos, considerando cada etapa descrita previamente. Además, un diagrama del proceso es ilustrado en la Figura 2.1.

Preparación:

1. Alice genera una cadena de bits aleatorios de largo n, denotada como a.
2. Alice codifica cada bit a_i a un estado cuántico $|\phi_i\rangle \in \{ |0\rangle, |1\rangle, |+\rangle, |-\rangle \}$, siguiendo la regla indicada en la Tabla 2.1.

Envío:

3. Alice envía $|\phi_i\rangle^{\oplus n}$ por el canal cuántico.

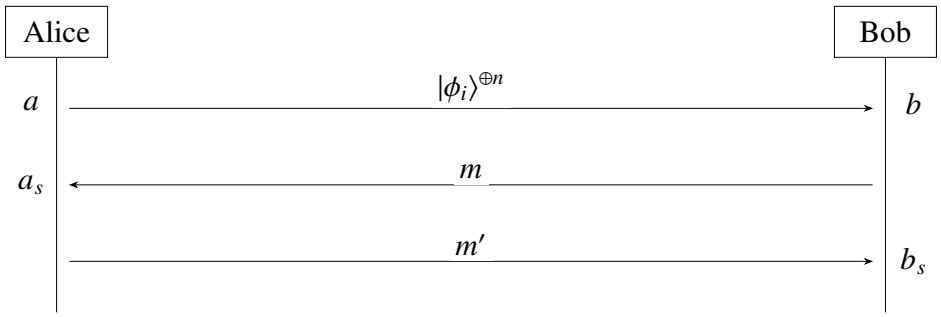


Figura 2.1: Diagrama simple de BB84, el primer protocolo QKD.

Medición:

4. Bob mide cada estado cuántico $|\phi_i\rangle$ utilizando, de manera aleatoria, las bases Z (estándar) o X (*Hadamard*), generando una cadena de bits aleatoria denominada b. Es importante recalcar que las bases definen la dirección de la medición proyectiva, donde una elección correcta permite colapsar el estado cuántico de Alice directamente al bit a_i con un 100 % de probabilidad. En cambio, una elección errónea de la base de medición hace colapsar el estado cuántico de Alice al bit a_i con un 50 % de probabilidad.

Sifting:

5. Bob envía, por el canal clásico, el listado de las bases de medición utilizadas para medir cada estado cuántico $|\phi_i\rangle$, denotado como m.
6. Alice recibe m y, dado que ella conoce los estados que fueron enviados, también sabe cuáles son las bases que fueron escogidas correctamente, es decir, sabe cuáles fueron los estados cuánticos que colapsaron al bit secreto con un 100 % de probabilidad. Así pues, Alice obtiene $a_s \subseteq a$, tomando sólo los bits donde sus correspondientes estados cuánticos fueron medidos por Bob utilizando la base correcta.
7. Alice envía, por el canal clásico, el listado de las bases de medición que fueron escogidas correctamente, denotado como m'. El método de decisión se adjunta en la Tabla 2.2.
8. Bob recibe m' y obtiene $b_s \subseteq b$, correspondiente a los bits que fueron obtenidos al utilizar las bases correctas indicadas en m'. Si no hay errores en el canal cuántico, entonces $a_s = b_s$.
9. En presencia de errores en el canal cuántico, Alice y Bob exponen una porción de sus bits, para estimar el error del canal. Es importante notar que los bits expuestos son descartados de las llaves de Alice y Bob.

Reconciliación:

10. Alice y Bob utilizan métodos correctores de errores basados en LDPC o Turbo Códigos. Así, ambos obtienen a'_s y b'_s respectivamente, donde se cumple que $a'_s = b'_s$.

Amplificación de privacidad [28]:

11. Alice y Bob estiman la cantidad de información (en bits) que un posible atacante activo podría tener, considerando el error del canal.
12. Alice y Bob aplican una función hash H a sus llaves, donde se cumple que $H(a'_s) = H(b'_s)$.

Base	$a_i = 0$	$a_i = 1$
Z	$\|0\rangle$	$\|1\rangle$
X	$\|+\rangle$	$\|-\rangle$

Tabla 2.1: Codificación de un bit clásico a un estado cuántico en BB84. Alice elige, de manera aleatoria, entre las bases Z y X.

m_i	$\|\phi_i\rangle$	Verificación
Z	$\|0\rangle$	✓
	$\|1\rangle$	✓
	$\|+\rangle$	✗
	$\|-\rangle$	✗
X	$\|0\rangle$	✗
	$\|1\rangle$	✗
	$\|+\rangle$	✓
	$\|-\rangle$	✓

Tabla 2.2: Regla de verificación para determinar si la base de medición b_i es correcta. En palabras simples, se verifica que b_i coincida con la base que Alice eligió para codificar la cadena de bits a qubits.

La seguridad de BB84, en el canal clásico, se ha mantenido intacta dado que únicamente se comparten las bases de medición de Bob y las que coinciden con Alice [29]. Así pues, no es posible recuperar la llave compartida a través de las bases de medición. Sin embargo, para el canal cuántico, se han demostrado vectores de ataque viables para poder comprometer la ejecución del protocolo, asumiendo un atacante activo [30]. Nos referimos a un atacante activo, cuando es posible modificar la información cuántica entre Alice y Bob.

Para la recuperación de la llave compartida, únicamente con la información clásica, es necesario realizar un ataque de fuerza bruta. Considerando que, dados los estándares actuales de llaves criptográficas [11], Alice y Bob deben compartir como mínimo 128 bits, el tiempo de ejecución del ataque de fuerza bruta es de aproximadamente 1 billón (10^{12}) de años.

2.2. Utilizando el entrelazamiento cuántico: *Bennet*, *Brassard* & *Mermin* (1992)

Recordemos que todo bit cuántico se modela de la siguiente manera:

$$|\phi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle \ , \ \alpha, \beta \in \mathbb{C}.$$

Se dice que $|\phi\rangle$ es un estado puro o *pure state*, ya que se compone de una combinación lineal de las bases $|0\rangle$ y $|1\rangle$. Por otro lado, existen los estados mixtos o *mixed states*, los cuales combinan estados puros. Así pues, un estado mixto $|\psi\rangle$ que combina dos estados puros se describe a continuación:

$$|\psi\rangle = \frac{|00\rangle + |01\rangle}{\sqrt{2}}. \tag{2.3}$$

Es importante notar que, en este caso $\alpha = \beta = \frac{1}{\sqrt{2}}$, donde se cumple que $\alpha^2 + \beta^2 = 1$, es decir, la definición de un estado cuántico. Además de que $|\psi\rangle$ sea un estado mixto, también es separable, ya que es posible expresar $|\psi\rangle$ como el producto de dos estados puros:

$$|\psi\rangle = |0\rangle \cdot \frac{|0\rangle + |1\rangle}{\sqrt{2}}. \tag{2.4}$$

La separabilidad del estado cuántico $|\psi\rangle$ implica que se puede construir mediante dos partículas independientes. Sin embargo, para Charles Bennet, Gilles Brassard y Nathaniel Mermin, los estados mixtos no separables, como $|\beta\rangle$, tenían el potencial de construir un paradigma diferente de QKD:

$$|\beta\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

Se dice que $|\beta\rangle$ es un estado mixto no separable, o bien, un estado entrelazado. En los estados entrelazados, al no existir una manera de tener dos partículas independientes, ambas partículas tienen que estar *conectadas*, es decir, la medición de un estado cuántico *afecta* al otro.

En 1992, Charles Bennet, Gilles Brassard y Nathaniel Mermin, desarrollaron uno de los primeros protocolos QKD basados en entrelazamiento cuántico, tomando con fuerza la estructura del BB84 [31]. Medir un estado entrelazado $|\beta\rangle$ hace que este colapse en un estado puro, el cual posee el mismo comportamiento que indicamos previamente en la Sección 2.1. El colapso de un estado entrelazado es indeterminable, independiente de la base de medición que se utilice, es decir:

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \xrightarrow{z} \{|0\rangle, |1\rangle\},$$

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \xrightarrow{x} \{|0\rangle, |1\rangle\}.$$

Así pues, el protocolo BBM92 es descrito mediante una serie de pasos, los cuales se encuentran ilustrados en la Figura 2.2 y, al igual que en BB84, la descripción se divide en cada etapa fundamental de un protocolo QKD.

Preparación:

1. Alice genera n estados cuánticos $|\beta\rangle$. Nótese que cada estado cuántico se compone de un sistema de 2 partículas, o bien, 2 qubits.
2. Alice mide sólo una partícula de cada estado cuántico $|\beta_i\rangle$ utilizando, de manera aleatoria, las bases Z o X. Debido al entrelazamiento cuántico, la partícula restante de $|\beta_i\rangle$ colapsa a un qubit $|\phi_i\rangle$, y se genera una cadena de bits aleatoria de largo n, denotada como a.

Envío:

3. Alice envía $|\phi_i\rangle^{\oplus n}$ por el canal cuántico.

Medición:

4. Bob mide cada estado cuántico $|\phi_i\rangle$ utilizando, de manera aleatoria, las bases Z (estándar) o X (*Hadamard*), obteniendo una cadena de bits aleatoria b.

Sifting:

5. Bob envía, por el canal clásico, el listado de las bases de medición utilizadas para medir cada estado cuántico $|\phi_i\rangle$, denotado como m.
6. Alice recibe m y obtiene $a_s \subseteq a$, tomando los bits a_i en los cuales su correspondiente estado cuántico $|\phi_i\rangle$ fue medido por Bob utilizando la base de medición correcta.
7. Alice envía, por el canal clásico, el listado de las bases de medición que fueron escogidas correctamente, denotado como m'. El método de decisión se adjunta en la Tabla 2.2.
8. Bob recibe m' y obtiene $b_s \subseteq b$, correspondiente a los bits que fueron obtenidos al utilizar las bases correctas indicadas en m'. Si no hay errores en el canal cuántico, entonces $a_s = b_s$.
9. En presencia de errores en el canal cuántico, Alice y Bob exponen una porción de sus bits, para estimar el error del canal. Es importante notar que los bits expuestos son descartados de las llaves de Alice y Bob.

Reconciliación:

10. Alice y Bob utilizan métodos correctores de errores basados en LDPC o Turbo Códigos. Así, ambos obtienen a'_s y b'_s respectivamente, donde se cumple que $a'_s = b'_s$.

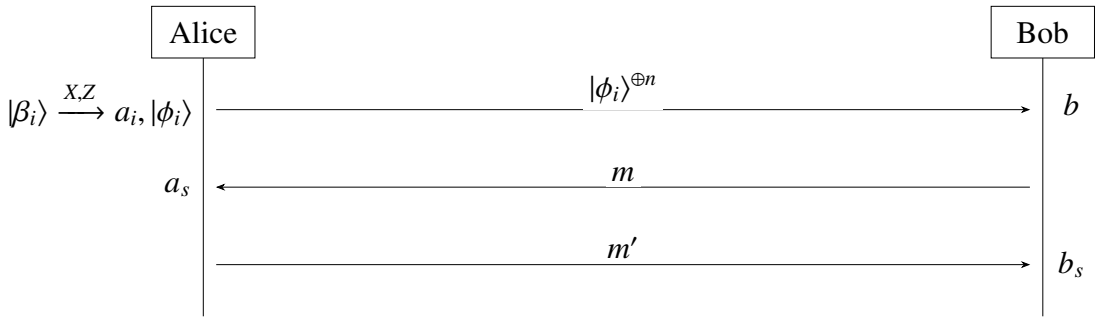


Figura 2.2: Diagrama simple de BBM92, la versión de BB84 que utiliza estados entrelazados.

Amplificación de privacidad:

11. Alice y Bob estiman la cantidad de información (en bits) que un posible atacante activo podría tener, considerando el error del canal.
12. Alice y Bob aplican una función hash H a sus llaves, donde se cumple que $H(a'_s) = H(b'_s)$.

La seguridad de BBM92 es equivalente a la de BB84, en términos del canal clásico. Así pues, no es posible recuperar la llave compartida únicamente con las bases de medición de Bob y el listado m' de forma eficiente. Sin embargo, a pesar de que BBM92 es más robusto que BB84 respecto a la seguridad del canal cuántico, debido al fenómeno del entrelazamiento, una gran parte de los ataques propuestos en la literatura se basan en la construcción de los dispositivos QKD [32].

2.3. Reconciliación basada en *frames*

Los protocolos QKD basados en frames son propuestas relativamente nuevas que cambian completamente la manera de implementar la QKD, con el objetivo de solucionar los principales problemas de BB84 y de QKD en general, como lo son los ataques *Photon Number Splitting* (PNS), *Intercept-Resend* (IR), las limitaciones que suponen los errores del canal cuántico y por ende, el tamaño de las llaves criptográficas resultantes.

A través de los frames es posible mitigar los ataques PNS e IR, además de corregir los errores del canal cuántico mediante métodos que toleran un mayor umbral de error, aumentando considerablemente el tamaño de la llave compartida. A continuación, se realiza un análisis del estado del arte de los protocolos QKD basados en frames, comenzando con la definición de un frame.

2.3.1. Definición de *frame*

Un *frame* se define como una matriz de dimensiones n x m, la cual agrupa estados cuánticos, y son utilizadas para las etapas de *Sifting*, Reconciliación y Amplificación de privacidad en los protocolos QKD basados en frames. La construcción de los frames asume que tanto Alice como Bob se encuentran sincronizados en tiempo, al igual que en BB84. En otras palabras, ambas partes conocen los índices i que identifican al estado cuántico (del lado de Alice), la base de medición (del lado de Bob) y el bit resultante (del lado de Alice y Bob). Consideremos el siguiente frame de dimensiones $n = m = 2$ de ejemplo:

$$f_b = \begin{pmatrix} 8 & 1 \\ 2 & 16 \end{pmatrix}.$$

Cada entrada de la matriz corresponde a ciertos índices i de los estados cuánticos que envía Alice en la etapa de Envío. Así pues, los estados cuánticos en las posiciones 1, 2, 8 y 16 son agrupados por Bob en un frame 2 x 2, para este ejemplo. Dado que Alice conoce los estados cuánticos que envía, ella puede transcribir el frame f_b de la siguiente manera:

$$f_a = \begin{pmatrix} |+\rangle & |1\rangle \\ |-\rangle & |0\rangle \end{pmatrix}.$$

Así pues, el frame f_a es una conversión del frame f_b, donde cada entrada de la matriz corresponde al estado cuántico que Alice envía en su respectivo índice de f_b.

Antes de continuar con el análisis del estado del arte de los protocolos QKD basados en frames, es necesario realizar un cambio en la notación de los estados cuánticos, para ser afín a la nueva notación que se utiliza en los protocolos que serán descritos, y para que el lector tenga una mayor claridad y legibilidad de los mismos. Los cambios en la notación se resumen en la Tabla 2.3.

Notación anterior	Notación actual
$\|0\rangle$	$\|0_Z\rangle$
$\|1\rangle$	$\|1_Z\rangle$
$\|+\rangle$	$\|0_X\rangle$
$\|-\rangle$	$\|1_X\rangle$

Tabla 2.3: Conversión de notación de estados cuánticos.

2.3.2. Los que dieron el primer paso: *Lizama* & *López* (2020)

En 2020, Luis Lizama y José Mauricio López desarrollaron el primer protocolo QKD basado en frames de dimensionalidad $n = m = 2$, el cual propone una mejora sustancial respecto a la seguridad y el tamaño de la llave criptográfica de BB84, ya que el uso de los frames permite mitigar los ataques PNS e IR, además de aumentar, cuadráticamente, el tamaño de la llave compartida [33]. Cada frame agrupa dos pares de estados cuánticos no ortogonales, los cuales se describen en la etapa de Preparación.

Para este protocolo y los venideros, se hablará del concepto del *double matching*. Alice enviará pares de estados los cuales serán medidos por Bob utilizando la misma base (Z o X). Si Bob obtiene el mismo bit en ambos estados cuánticos, entonces se dice que Bob obtuvo un *double matching* o doble coincidencia. Así pues, además de que Bob pueda construir, por ejemplo, el frame f_b, él también puede transcribirlo de la siguiente manera:

$$f'_b = \begin{pmatrix} - & 1_Z \\ 1_X & - \end{pmatrix}. \tag{2.5}$$

El frame f'_b corresponde a una conversión de f_b, el cual nos dice que el primer par de estados cuánticos, denotado como $(8, 1)$, colapsó a los bits '11', es decir, ocurrió un *double matching*. Adicionalmente, esa doble coincidencia se obtuvo utilizando la base Z, por lo que el par se transcribe como $(-, 1_Z)$. En esta nueva notación, los *double matchings* en Z se anotan a la derecha, mientras que los que se realizan en X se anotan a la izquierda. Por último, en el segundo par, es decir, $(2, 16)$, se obtuvo un *double matching* en X, colapsando en '11', por lo que se transcribe como $(1_X, -)$.

En los 4 protocolos QKD basados en frames, también se habla de los *sifting bits*. Los *sifting bits* se calculan a través del operador XOR de cada columna de f'_b, donde el símbolo $-$ se toma como un bit '0'. Así pues, los *sifting bits* de f'_b son '11', ya que $0 \oplus 1 = 1$ (primera columna) y $1 \oplus 0 = 1$ (segunda columna). Estos valores se utilizan para derivar la llave compartida de Alice y Bob, además de poder detectar y corregir parcialmente los errores del canal cuántico.

A continuación, el protocolo LL20 es descrito mediante una serie de pasos, respetando las etapas fundamentales mencionadas en la Sección 2.1. Además, un diagrama de interacción entre Alice y Bob es ilustrado en la Figura 2.3:

Preparación:

1. Alice genera n pares de estados cuánticos $P_i \in \{ (|0_X\rangle, |0_Z\rangle), (|0_X\rangle, |1_Z\rangle), (|1_X\rangle, |0_Z\rangle), (|1_X\rangle, |1_Z\rangle) \}$. Se dice que P_i es un par no ortogonal, ya que se encuentra compuesto de un par de estados cuánticos no ortogonales respecto a la esfera de Bloch.

Envío:

2. Alice envía P por el canal cuántico.

Medición:

3. Bob mide cada par P_i de manera aleatoria, escogiendo entre las bases Z y X. Es importante notar que ambos estados son medidos con la base escogida.

Sifting:

4. Bob envía, por el canal clásico, un listado de los índices i de los pares P_i que colapsaron al mismo bit (*double matching*), denominado m.

5. Alice recibe m y calcula todos los posibles frames usables. Un frame se denomina usable si sus correspondientes *sifting bits* permiten determinar, con total certeza, las bases de medición utilizadas por Bob para medir cada par. Así pues, Alice calcula $\binom{|m|}{2}$ combinaciones, almacenando cada frame usable f_j. En presencia de errores en el canal cuántico, Alice también incluye frames *auxiliares*.

6. Alice envía el listado de frames f por el canal clásico.

7. Bob recibe cada frame f_j y calcula su *Sifting String* (SS), denotado como S_j. El SS se compone de los *sifting bits* y los bits obtenidos, es decir:

$$SS = \text{1er sifting bit} \,\|\, \text{2do sifting bit} \,,\, \text{1er bit obtenido} \,\|\, \text{2do bit obtenido} \,,$$

   donde el símbolo $\|$ corresponde a una concatenación.

8. Bob calcula la llave compartida k_b utilizando la Tabla 2.4. El bit secreto de un frame f_j se obtiene a través de su SS y *Measurement Result* (MR), donde este último corresponde a la orientación de las bases de medición, el cual es expresado mediante dos bits, utilizando la Tabla 2.5.

9. Bob envía el listado de *Sifting Strings*, es decir, S, por el canal clásico.

10. Alice recibe S y procede a derivar los MR de Bob, utilizando la Tabla 2.6.

Reconciliación:

11. En presencia de errores en el canal cuántico, estos son detectables y corregibles por Alice a través de los frames *auxiliares*. Un frame auxiliar no es apto para determinar su correspondiente MR, por lo que se utiliza únicamente para detectar posibles errores de otros frames usables, siempre y cuando un par del frame auxiliar no contenga errores.

12. Alice envía, por el canal clásico, el listado de frames f' que serán descartados de la llave compartida, que corresponden tanto a los frames auxiliares como a los que contienen SS iguales a $00, 00$, $10, 01$, $10, 10$, $01, 01$ o $01, 10$. En caso de no haber errores en el canal cuántico, se mantienen todos los frames.

13. Alice deriva la llave compartida k_a utilizando la Tabla 2.4, donde a cada frame f_j se le asigna un bit secreto, dependiendo de su SS y MR. Al finalizar el protocolo, $k_a = k_b$.

Amplificación de privacidad:

- En LL20, esta etapa pasa a ser llamada **Pre-amplificación de privacidad**, la cual es realizada al calcular $\binom{|m|}{2}$ combinaciones en el paso 5.

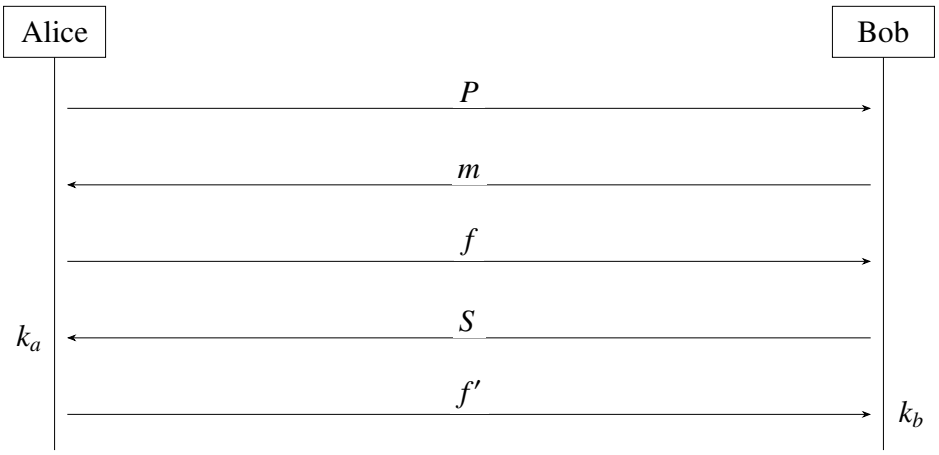


Figura 2.3: Diagrama simple de LL20, el primer protocolo QKD basado en frames. Se asumen errores en el canal cuántico, por lo que la derivación de la llave compartida se realiza luego de descartar los frames que no son posibles de concluir sus MRs. El diagrama es equivalente al protocolo LLS21

Tabla 2.4: Tabla de derivación de los bits secretos de LL20. Por cada S_j, se listan sus dos posibles MR (MR_j^1, MR_j^2) y bits secretos (k_j^1, k_j^2).

S_j	MR_j^1	MR_j^2	k_j^1	k_j^2
00,11	00	01	0	1
11,11	11	10	0	1
01,10	01	11	0	1
01,01	10	01	0	1
10,01	00	11	0	1
10,10	00	10	0	1

Tabla 2.5: Tabla de derivación de los *Measurement Results* (MRs) en LL20 y LL21, dada la orientación de las bases de medición del frame f_j. Del lado de Alice, ella debe determinar la orientación de las bases de Bob, previo a la obtención de los MRs. Por otro lado, Bob calcula los MRs de manera directa. Es importante notar que los MRs no dependen del bit resultante de cada *double matching*, sino únicamente de la orientación de las bases de medición.

Orientación de las bases	MR
$\begin{pmatrix} X & - \\ X & - \end{pmatrix}$	00
$\begin{pmatrix} - & Z \\ - & Z \end{pmatrix}$	01
$\begin{pmatrix} X & - \\ - & Z \end{pmatrix}$	10
$\begin{pmatrix} - & Z \\ X & - \end{pmatrix}$	11

Tabla 2.6: Orientaciones de las bases de medición, dado un *Sifting String* (SS) de LL20. Es importante notar que no todas las combinaciones de SS existen, como en $00, 01$, donde no existe un frame f_j que tenga *sifting bits* '00' y que además el primer y el segundo bit obtenido sean 0' y '1', respectivamente.

Sifting String (SS)	Orientaciones de las bases
$00, 00$	$\begin{pmatrix} 0_X & - \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ 0_X & - \end{pmatrix}$
$00, 01$	-
$00, 10$	-
$00, 11$	$\begin{pmatrix} 1_X & - \\ 1_X & - \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ - & 1_Z \end{pmatrix}$
$01, 00$	-
$01, 01$	$\begin{pmatrix} 0_X & - \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ - & 1_Z \end{pmatrix}$
$01, 10$	$\begin{pmatrix} - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 0_X & - \end{pmatrix}$
$01, 11$	-
$10, 00$	-
$10, 01$	$\begin{pmatrix} - & 0_Z \\ 1_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ 1_X & - \end{pmatrix}$
$10, 10$	$\begin{pmatrix} 1_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 1_X & - \\ - & 0_Z \end{pmatrix}$
$10, 11$	-
$11, 00$	-
$11, 01$	-
$11, 10$	-
$11, 11$	$\begin{pmatrix} 1_X & - \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 1_X & - \end{pmatrix}$

A diferencia de BB84 y BBM92, en LL20 no se exponen las bases medición. En cambio, información asociada a los *double matchings*, la construcción de los frames f_j y sus correspondientes SS es entregada, aumentando la superficie de ataque. Nos referimos a la superficie de ataque como el conjunto de todos los métodos posibles que se pueden utilizar para intentar comprometer, en este caso, el protocolo QKD, del punto de vista del criptoanálisis clásico.

Por otro lado, el tamaño de la llave criptográfica aumenta cuadráticamente, respecto de la cantidad de *double matchings* ($|m|$) y el *Quantum Bit Error Rate* (QBER) e:

$$T_{LL20} = \frac{1}{4} \cdot \binom{|m|}{2} \cdot \left(\frac{1}{2} - \frac{1}{3}e\right).$$

La seguridad de LL20, hasta el momento, se ha considerado intacta, ya que un SS se encuentra asociado a mínimo dos MRs [27]. Sin embargo, en el Capítulo 3, demostramos por primera vez que es posible recuperar la llave compartida de manera parcial y total, considerando una comunicación sin y con errores, respectivamente.

2.3.3. Aumentando la dimensionalidad de los *frames*: *Lizama*, *López* & *Samperio* (2021)

Tomando como referencia el protocolo LL20, los investigadores Luis Lizama, José Mauricio López y Emmanuel Samperio, en el año 2021, desarrollaron una extensión del protocolo, para frames de dimensionalidad 3 x 2 [26]. Así, las mismas características del protocolo LL20 se mantienen, pero el aumento del tamaño de la llave criptográfica pasa a ser de orden cúbico, respecto a $|m|$ y e. A continuación, el protocolo LLS21 es descrito mediante cada una de las etapas fundamentales de un protocolo QKD, donde la interacción entre Alice y Bob es equivalente a la Figura 2.3:

Preparación:

1. Alice genera n pares de estados cuánticos no ortogonales $P_i \in \{ (|0_X\rangle, |0_Z\rangle), (|0_X\rangle, |1_Z\rangle), (|1_X\rangle, |0_Z\rangle), (|1_X\rangle, |1_Z\rangle) \}$.

Envío:

2. Alice envía P por el canal cuántico.

Medición:

3. Bob mide cada par P_i de manera aleatoria, escogiendo entre las bases Z y X. Es importante notar que ambos estados son medidos con la base escogida.

Sifting:

4. Bob envía, por el canal clásico, un listado de los índices i de los pares P_i que colapsaron al mismo bit (*double matching*), denominado m.
5. Alice recibe m y calcula todos los posibles frames usables. De un total de $\binom{|m|}{3}$ combinaciones, los frames usables f_j son almacenados. En presencia de errores en el canal cuántico, frames auxiliares son añadidos.
6. Alice envía el listado de frames f por el canal clásico.
7. Bob recibe cada frame f_j y calcula su *Sifting String* (SS), denotado como S_j. El SS se compone de los *sifting bits* y los bits obtenidos, es decir:

   SS = 1er sifting bit || 2do sifting bit || 3er sifting bit, 1er bit obtenido || 2do bit obtenido || 3er bit obtenido,

   donde el símbolo || corresponde a una concatenación. El tercer sifting bit es equivalente a un bit de paridad, dependiendo de la orientación de las bases de medición. El cálculo de este bit es descrito en la Tabla 2.7.
8. Bob calcula la llave compartida k_b utilizando la Tabla 2.8. El bit secreto de un frame usable f_j se obtiene a través de su SS y *Measurement Result* (MR), donde este último se obtiene a través de la Tabla 2.13.
9. Bob envía el listado de *Sifting Strings*, es decir, S, por el canal clásico.
10. Alice recibe S y procede a derivar los MR de Bob, utilizando las Tablas 2.9, 2.10, 2.11 y 2.12.

Reconciliación:

11. En presencia de errores en el canal cuántico, estos son detectables y corregibles por Alice a través de los frames auxiliares, donde dos pares deben encontrarse sin errores.
12. Alice envía, por el canal clásico, el listado de frames f' que serán descartados de la llave compartida, que corresponden tanto a los frames auxiliares como a los que contienen SS distintos a $011, 111$, $011, 001$, $110, 110$ o $111, 011$. En caso de no haber errores en el canal cuántico, se mantienen todos los frames.
13. Alice deriva la llave compartida k_a utilizando la Tabla 2.8, donde a cada frame f_j se le asigna un bit secreto, dependiendo de su SS y MR. Al finalizar el protocolo, $k_a = k_b$.

Amplificación de privacidad:

- En LLS21, esta etapa pasa a ser llamada **Pre-amplificación de privacidad**, la cual es realizada al calcular $\binom{|m|}{3}$ combinaciones en el paso 5.

Al igual que en LL20, las bases de medición no son expuestas, pero sí los índices de los pares de estados que colapsaron al mismo bit (*double matching*), los frames f_j que serán construidos y eliminados, además de sus correspondientes SS. La mejora de LLS21 recae en el tamaño de la llave, ya que aumenta cúbicamente, respecto a $|m|$ y e:

$$T_{LLS21} = \frac{3}{8} \cdot \binom{|m|}{3} \cdot \left(\frac{1}{3} - \frac{2}{7}e\right).$$

La seguridad de LLS21 no ha sido analizada todavía. En el Capítulo 3, demostramos que es posible recuperar la llave compartida de manera total, independiente del tipo de comunicación (con o sin errores).

Tabla 2.7: Tabla de derivación del tercer *sifting bit* de LLS21

Orientaciones de bases	Bit de paridad
$\begin{pmatrix} X & - \\ X & - \\ X & - \end{pmatrix}, \begin{pmatrix} - & Z \\ - & Z \\ - & Z \end{pmatrix}, \begin{pmatrix} X & - \\ - & Z \\ X & - \end{pmatrix}, \begin{pmatrix} - & Z \\ X & - \\ - & Z \end{pmatrix}$	0
$\begin{pmatrix} X & - \\ X & - \\ - & Z \end{pmatrix}, \begin{pmatrix} - & Z \\ - & Z \\ X & - \end{pmatrix}, \begin{pmatrix} X & - \\ - & Z \\ - & Z \end{pmatrix}, \begin{pmatrix} - & Z \\ X & - \\ X & - \end{pmatrix}$	1

2.3.4. El camino a la reconciliación perfecta: *Lizama* & *López* (2021)

Con el objetivo de detectar y corregir todos los posibles errores del canal cuántico, Luis Lizama y José Mauricio López, en 2021, desarrollaron una modificación de LL20 [34], ya que los *Sifting Strings* (SS) no permiten detectar los errores provenientes de los frames usables f_j donde sus correspondientes SS son iguales a $10, 01$, $10, 10$, $01, 01$ o $01, 10$, por lo que son descartados. Para solucionar esto, se introduce el *Composed Sifting String* (CSS), el cual hace uso de los frames *conjugados*. Vamos a tomar el frame f'_b que presentamos como ejemplo para introducir las diferentes transcripciones que posee un frame en la Sección 2.4.2, donde simplificaremos su notación a f_b:

$$f_b = \begin{pmatrix} - & 1_Z \\ 1_X & - \end{pmatrix}.$$

La versión conjugada de f_b, denotada como f_b^c, corresponde a la negación de todos sus bits, es decir:

$$f_b^c = \begin{pmatrix} - & 0_Z \\ 0_X & - \end{pmatrix}.$$

Así pues, podemos notar que los *sifting bits* del frame conjugado son '00'. A continuación, el protocolo LL21 es descrito mediante una serie de pasos, abarcando todas las etapas fundamentales de un protocolo QKD. La interacción entre Alice y Bob es ilustrada mediante la Figura 2.4.

Preparación:

1. Alice genera n pares de estados cuánticos no ortogonales $P_i \in \{ (|0_X\rangle, |0_Z\rangle), (|0_X\rangle, |1_Z\rangle), (|1_X\rangle, |0_Z\rangle), (|1_X\rangle, |1_Z\rangle) \}$.

Envío:

2. Alice envía P por el canal cuántico.

Medición:

3. Bob mide cada par P_i de manera aleatoria, escogiendo entre las bases Z y X. Es importante notar que ambos estados son medidos con la base escogida.

Sifting:

4. Bob envía, por el canal clásico, un listado de los índices i de los pares P_i que colapsaron al mismo bit (*double matching*), denominado m.
5. Alice recibe m y calcula todos los posibles frames. De un total de $\binom{|m|}{2}$ combinaciones, cada frame usable f_j es almacenado. En presencia de errores en el canal cuántico, frames auxiliares son añadidos.
6. Alice envía el listado de frames f por el canal clásico.

Tabla 2.8: Tabla de derivación de los bits secretos de LLS21. Por cada S_j, se listan sus dos posibles MR (MR_j^1) y MR_j^2) y bits secretos (k_j^1 y k_j^2).

S_j	MR_j^1	MR_j^2	k_j^1	k_j^2
000,110	000	001	0	1
000,011	000	001	0	1
001,011	110	111	0	1
001,110	100	101	0	1
011,010	110	101	0	1
011,111	100	111	0	1
010,001	001	011	0	1
010,100	001	011	0	1
010,010	001	010	0	1
010,111	001	010	0	1
011,001	110	100	0	1
011,100	101	111	0	1
100,001	000	010	0	1
100,100	000	010	0	1
100,010	000	011	0	1
100,111	000	011	0	1
101,001	111	101	0	1
101,100	100	110	0	1
101,010	111	100	0	1
101,111	101	110	0	1
110,011	010	011	0	1
110,110	010	011	0	1
111,011	101	100	0	1
111,110	110	111	0	1

Tabla 2.9: Orientaciones de las bases de medición, dado un *Sifting String* (SS) de LLS21. Es importante notar que no todas las combinaciones de SS existen, como en $S_j = 000, 001$, donde no existe un frame que al ser medido tenga *sifting bits* '000' y que además los tres bits hayan colapsado en '0', '0' y '1', respectivamente.

Sifting String (SS)	Orientaciones de las bases
$000, 000$	$\begin{pmatrix} - & 0_Z \\ 0_X & - \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} 0_X & - \\ - & 0_Z \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ - & 0_Z \\ - & 0_Z \end{pmatrix}$
$000, 001$	-
$000, 010$	-
$000, 011$	$\begin{pmatrix} - & 0_Z \\ - & 1_Z \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} 0_X & - \\ 1_X & - \\ 1_X & - \end{pmatrix}$
$000, 100$	-
$000, 101$	$\begin{pmatrix} 1_X & - \\ - & 0_Z \\ 1_X & - \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 0_X & - \\ - & 1_Z \end{pmatrix}$
$000, 110$	$\begin{pmatrix} - & 1_Z \\ - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} 1_X & - \\ 1_X & - \\ 0_X & - \end{pmatrix}$
$000, 111$	-
$001, 000$	$\begin{pmatrix} 0_X & - \\ 0_X & - \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ 0_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ - & 0_Z \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ - & 0_Z \\ - & 0_Z \end{pmatrix}$
$001, 001$	-
$001, 010$	-
$001, 011$	$\begin{pmatrix} - & 0_Z \\ 1_X & - \\ 1_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ - & 1_Z \\ - & 1_Z \end{pmatrix}$
$001, 100$	-
$001, 101$	-
$001, 110$	$\begin{pmatrix} - & 1_Z \\ - & 1_Z \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 1_X & - \\ 1_X & - \\ - & 0_Z \end{pmatrix}$
$001, 111$	-
$010, 000$	-
$010, 001$	$\begin{pmatrix} - & 0_Z \\ - & 0_Z \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ 0_X & - \\ - & 1_Z \end{pmatrix}$
$010, 010$	$\begin{pmatrix} - & 0_Z \\ - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} 0_X & - \\ - & 1_Z \\ 0_X & - \end{pmatrix}$
$010, 011$	-

Tabla 2.10: *Cont.*

Sifting String (SS)	Orientaciones de las bases
$010, 100$	$\begin{pmatrix} - & 1_Z \\ 0_X & - \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ - & 0_Z \\ - & 0_Z \end{pmatrix}$
$010, 101$	-
$010, 110$	-
$010, 111$	$\begin{pmatrix} 1_X & - \\ - & 1_Z \\ 1_X & - \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ - & 1_Z \\ - & 1_Z \end{pmatrix}$
$011, 000$	-
$011, 001$	$\begin{pmatrix} 0_X & - \\ - & 0_Z \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} 0_X & - \\ 0_X & - \\ - & 1_Z \end{pmatrix}$
$011, 010$	$\begin{pmatrix} 0_X & - \\ - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ - & 1_Z \\ 0_X & - \end{pmatrix}$
$011, 011$	-
$011, 100$	$\begin{pmatrix} - & 1_Z \\ 0_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ - & 0_Z \\ 0_X & - \end{pmatrix}$
$011, 101$	-
$011, 110$	-
$011, 111$	$\begin{pmatrix} 1_X & - \\ 1_X & - \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 1_X & - \\ 1_X & - \end{pmatrix}$
$100, 000$	-
$100, 001$	$\begin{pmatrix} 0_X & - \\ 0_X & - \\ 1_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ - & 0_Z \\ 1_X & - \end{pmatrix}$
$100, 010$	$\begin{pmatrix} 0_X & - \\ 1_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ 1_X & - \\ - & 0_Z \end{pmatrix}$
$100, 011$	-
$100, 100$	$\begin{pmatrix} 1_X & - \\ 0_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 1_X & - \\ - & 0_Z \\ 0_X & - \end{pmatrix}$
$100, 101$	-
$100, 111$	$\begin{pmatrix} 1_X & - \\ 1_X & - \\ 1_X & - \end{pmatrix}, \begin{pmatrix} . & 1_Z \\ 1_X & - \\ - & 1_Z \end{pmatrix}$
$101, 000$	-

Tabla 2.11: *Cont.*

Sifting String (SS)	Orientaciones de las bases
$101,001$	$\begin{pmatrix} - & 0_Z \\ 0_X & - \\ 1_X & - \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ - & 0_Z \\ 1_X & - \end{pmatrix}$
$101,010$	$\begin{pmatrix} - & 0_Z \\ 1_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ 1_X & - \\ - & 0_Z \end{pmatrix}$
$101,011$	-
$101,100$	$\begin{pmatrix} 1_X & - \\ 0_X & - \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} 1_X & - \\ - & 0_Z \\ - & 0_Z \end{pmatrix}$
$101,101$	-
$101,110$	-
$101,111$	$\begin{pmatrix} - & 1_Z \\ - & 1_Z \\ 1_X & - \end{pmatrix}, \begin{pmatrix} 1_X & - \\ - & 1_Z \\ - & 1_Z \end{pmatrix}$
$110,000$	-
$110,001$	-
$110,010$	-
$110,011$	$\begin{pmatrix} 0_X & - \\ - & 1_Z \\ 1_X & - \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ 1_X & - \\ - & 1_Z \end{pmatrix}$
$110,100$	-
$110,101$	-
$110,110$	$\begin{pmatrix} 1_X & - \\ - & 1_Z \\ 0_X & - \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 1_X & - \\ - & 0_Z \end{pmatrix}$
$110,111$	-
$111,000$	-
$111,001$	-
$111,010$	-
$111,011$	$\begin{pmatrix} - & 0_Z \\ - & 1_Z \\ 1_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ 1_X & - \\ - & 1_Z \end{pmatrix}$
$111,100$	-
$111,101$	$\begin{pmatrix} - & 1_Z \\ - & 0_Z \\ 1_X & - \end{pmatrix}, \begin{pmatrix} 1_X & - \\ - & 0_Z \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 0_X & - \\ 1_X & - \end{pmatrix}, \begin{pmatrix} 1_X & - \\ 0_X & - \\ - & 1_Z \end{pmatrix}$
$111,110$	$\begin{pmatrix} 1_X & - \\ - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 1_X & - \\ 0_X & - \end{pmatrix}$

Tabla 2.12: *Cont.*

Sifting String (SS)	Orientaciones de las bases
$111,111$	-

Tabla 2.13: Tabla de derivación de los *Measurement Results* (MRs) en LLS21, dada la orientación de las bases de medición del frame f_j. Del lado de Alice, ella debe determinar la orientación de las bases de Bob, previo a la obtención de los MRs. Por otro lado, Bob calcula los MRs de manera directa. Es importante notar que los MRs no dependen del bit resultante de cada *double matching*, sino únicamente de la orientación de las bases de medición.

Orientación de las bases	MR
$\begin{pmatrix} X & - \\ X & - \\ X & - \end{pmatrix}$	000
$\begin{pmatrix} - & Z \\ - & Z \\ - & Z \end{pmatrix}$	001
$\begin{pmatrix} X & - \\ - & Z \\ X & - \end{pmatrix}$	010
$\begin{pmatrix} - & Z \\ X & - \\ - & Z \end{pmatrix}$	011
$\begin{pmatrix} X & - \\ X & - \\ - & Z \end{pmatrix}$	100
$\begin{pmatrix} - & Z \\ - & Z \\ X & - \end{pmatrix}$	101
$\begin{pmatrix} X & - \\ - & Z \\ - & Z \end{pmatrix}$	110
$\begin{pmatrix} - & Z \\ X & - \\ X & - \end{pmatrix}$	111

7. Bob recibe cada frame f_j y calcula su *Composed Sifting String* (CSS), denotado como C_j. El CSS se compone de los *sifting bits* del frame regular y su versión conjugada, es decir:

$$CSS = \text{1er sifting bit} \,||\, \text{2do sifting bit} \,,\, \text{1er sifting bit conjugado} \,||\, \text{2do sifting bit conjugado}$$

   donde el símbolo || corresponde a una concatenación.

8. Bob calcula la llave compartida k_b utilizando la Tabla 2.15. El bit secreto de un frame usable f_j se obtiene a través de su CSS y *Measurement Result* (MR), donde este último corresponde a la orientación de las bases de medición, el cual es expresado mediante dos bits, utilizando la Tabla 2.5.
9. Bob envía el listado de *Composed Sifting Strings*, es decir, C, por el canal clásico.
10. Alice recibe C y procede a derivar los MR de Bob, utilizando la Tabla 2.14.

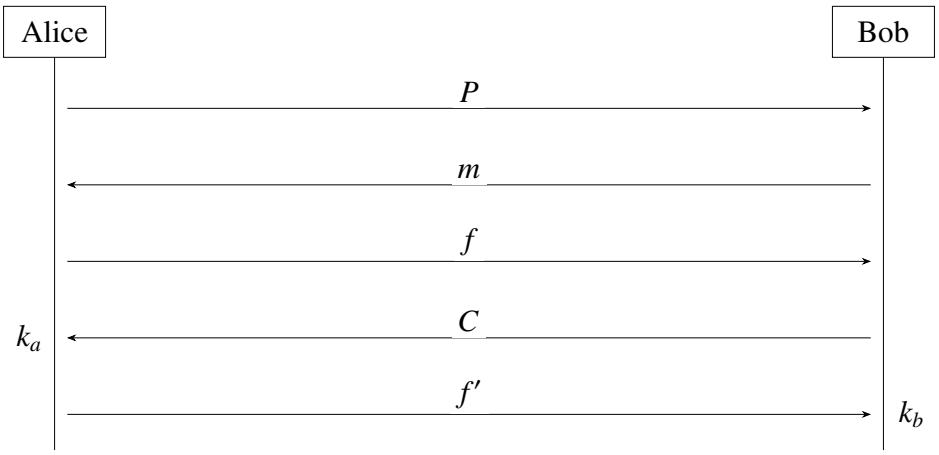


Figura 2.4: Diagrama simple de LL21, una versión de LL20 que no expone los bits medidos, sino los *Sifting bits* del frame conjugado. Se asumen errores en el canal cuántico, por lo que la derivación de la llave compartida se realiza luego de descartar los frames que no son posibles de concluir sus MRs.

Reconciliación:

11. En presencia de errores en el canal cuántico, estos son detectables y corregibles por Alice a través de los frames auxiliares y regulares. Para los frames regulares, es importante que un par se encuentre sin error.
12. Alice envía, por el canal clásico, el listado de frames f' que serán descartados de la llave compartida, que corresponden a los frames auxiliares. En caso de no haber errores en el canal cuántico, se mantienen todos los frames.
13. Alice deriva la llave compartida k_a utilizando la Tabla 2.15, donde a cada frame f_j se le asigna un bit secreto, dependiendo de su SS y MR. Al finalizar el protocolo, $k_a = k_b$.

Amplificación de privacidad:

- En LL21, esta etapa pasa a ser llamada **Pre-amplificación de privacidad**, la cual es realizada al calcular $\binom{|m|}{2}$ combinaciones en el paso 5.

Tabla 2.15: Tabla de derivación de los bits secretos de LL21. Por cada C_j, se listan sus dos posibles MR (MR_j^1) y MR_j^2) y bits secretos (k_j^1 y k_j^2).

S_j	MR_j^1	MR_j^2	k_j^1	k_j^2
01,10	10	11	0	1
10,01	11	10	0	1
00,11	10	11	0	1
00,00	00	01	0	1
11,00	11	10	0	1

La comunicación clásica de LL21 posee características equivalentes respecto a sus antecesores, donde las bases de medición no son expuestas, pero sí las listas m, f, C y f'. Además, el tamaño de la llave compartida aumenta cuadráticamente, respecto de $|m|$:

$$T_{LL21} = \frac{1}{2} \cdot \binom{|m|}{2}.$$

Tabla 2.14: Orientaciones de las bases de medición, dado un *Composed Sifting String* (CSS) de LL21. Es importante notar que no todas las combinaciones de CSS existen, como en $C_j = 00, 01$, donde no existe un frame que al ser medido tenga *sifting bits* '00' y que además su versión conjugada tenga *sifting bits* '01'.

Composed Sifting String (CSS)	Orientaciones de las bases
$00, 00$	$\begin{pmatrix}0_X & -\\ 0_X & -\end{pmatrix}, \begin{pmatrix}- & 0_Z\\ - & 0_Z\end{pmatrix}, \begin{pmatrix}1_X & -\\ 1_X & -\end{pmatrix}, \begin{pmatrix}- & 1_Z\\ - & 1_Z\end{pmatrix}$
$00, 01$	-
$00, 10$	-
$00, 11$	$\begin{pmatrix}0_X & -\\ - & 0_Z\end{pmatrix}, \begin{pmatrix}- & 0_Z\\ 0_X & -\end{pmatrix}$
$01, 00$	-
$01, 01$	$\begin{pmatrix}- & 1_Z\\ - & 0_Z\end{pmatrix}, \begin{pmatrix}- & 0_Z\\ - & 1_Z\end{pmatrix}$
$01, 10$	$\begin{pmatrix}- & 1_Z\\ 0_X & -\end{pmatrix}, \begin{pmatrix}0_X & -\\ - & 1_Z\end{pmatrix}$
$01, 11$	-
$10, 00$	-
$10, 01$	$\begin{pmatrix}1_X & -\\ - & 0_Z\end{pmatrix}, \begin{pmatrix}- & 0_Z\\ 1_X & -\end{pmatrix}$
$10, 10$	$\begin{pmatrix}1_X & -\\ 0_X & -\end{pmatrix}, \begin{pmatrix}0_X & -\\ 1_X & -\end{pmatrix}$
$10, 11$	-
$11, 00$	$\begin{pmatrix}1_X & -\\ - & 1_Z\end{pmatrix}, \begin{pmatrix}- & 1_Z\\ 1_X & -\end{pmatrix}$
$11, 01$	-
$11, 10$	-
$11, 11$	-

El protocolo LL21 se considera inseguro, debido a que existen CSS con sólo un MR posible, permitiendo recuperar, de manera parcial, la llave compartida [27]. En concreto, estos CSS son $10, 10$ y $01, 01$, los cuales poseen MRs iguales a '00' y '01', respectivamente, razón por la cual los autores recomiendan descartarlos. Sin embargo, en el Capítulo 3 se demuestra, por primera vez, que un ataque de recuperación total de la llave compartida es posible y eficiente de realizar.

2.3.5. La reconciliación en reversa: *Lizama* (2023)

Habiendo introducido los protocolos LL20, LLS21 y LL21, podemos notar que, entre ellos, comparten fuertemente una estructura base, correspondiente a la construcción de los frames y sus SS (o CSS en el caso del LL21). Además, en estos protocolos se envían pares de estados cuánticos no ortogonales, tendiendo a dificultar posibles implementaciones reales debido a la necesidad de emitir y recibir pares de fotones polarizados.

Un nuevo protocolo QKD basado en frames fue propuesto por Luis Lizama, en el año 2023, el cual se diferencia de sus antecesores ya que se toma la etapa de Preparación de BB84, la construcción de los frames se realiza del lado de Bob, el cálculo de los MR del lado de Alice se realiza a través del listado de frames enviados por Bob, y todos

los posibles errores del canal cuántico pueden ser detectados y corregidos. Así pues, el protocolo L23 es descrito a continuación mediante una serie de pasos, respetando cada etapa fundamental que debe tener un protocolo QKD, descrita en la Sección 2.1. La interacción entre Alice y Bob es ilustrada en la Figura 2.5.

Preparación:

1. Alice genera, de manera aleatoria, n estados cuánticos $|\phi_i\rangle \in \{ |0\rangle, |1\rangle, |+\rangle, |-\rangle \}$.

Envío:

2. Alice envía $|\phi_i\rangle^{\oplus n}$ por el canal cuántico.

Medición:

3. Bob mide cada estado cuántico $|\phi_i\rangle$ utilizando, de manera aleatoria, las bases Z o X, generando una cadena de bits aleatoria denominada b.

Sifting:

4. Bob construye las listas b_0 y b_1, que contienen los índices de los estados que colapsaron a un bit '0' y '1', respectivamente.
5. Bob calcula todos los posibles pares P_j en b_1 que fueron medidos utilizando la misma base. Dicho de otra manera, Bob agrupa todos los *double matchings* posibles, realizando $|P| = \binom{|b_1^X|}{2} + \binom{|b_1^Z|}{2}$ combinaciones, donde b_1^X y b_1^Z corresponden a las listas de los índices de los estados cuánticos que colapsaron a un bit '1', utilizando las bases X y Z respectivamente.
6. Bob procede a construir dos listas de frames, denotadas como L_1 y L_2. La lista L_1 contiene todos los posibles frames donde las bases de medición entre ambos pares P_j son diferentes. Por otro lado, la lista L_2 contiene todos los posibles frames donde las bases de medición entre ambos pares P_j son iguales. Las posibles orientaciones de las bases de medición que contiene cada lista se describen en la Tabla 2.16.
7. Bob procede a calcular la llave compartida k_b^r, utilizando la lista L_1 y la Tabla 2.17.
8. Bob envía las listas L_1 y L_2, a través del canal clásico.
9. Alice recibe las listas L_1 y L_2. Luego, ella procede a buscar frames f_1 y f_5 en L_1, descritos en la Tabla 2.18. Los conjuntos de frames f_1 y f_5 encontrados los denotamos como F_1 y F_5, respectivamente.

Reconciliación:

10. Alice procede a detectar y corregir posibles errores de los frames f_1 y f_5 contenidos en F_1 y F_5. Para ello, cada frame pasa por un esquema de votación, donde una mayoría considerable demuestra que el frame escogido no contiene errores. El esquema aprovecha el hecho de que un frame es erróneo en un 25 %, ya que se requiere que los dos pares se encuentren con error.
11. Alice escoge un pivote p, definido como un par de índices P_j perteneciente a un frame de F_1 o F_5 (escogido de manera aleatoria). Con él, Alice construye un frame *de prueba* f_T, incluyendo un par de índices de cualquier frame $f_m \in L_1 - F_1 - F_5$. Luego, la Tabla 2.19 es utilizada para derivar el bit secreto de f_m, y el proceso es repetido para todos los frames f_m restantes. Al finalizar, Alice obtiene la llave compartida k_a^r.
12. Bob invierte la cadena de bits b y los pasos desde el 4 al 11 son repetidos. A este proceso le llamaremos *rondas*, donde utilizaremos un superíndice $r \in \{1, 2\}$ para referirnos a la llave compartida obtenida en la r-ésima ronda. Así pues, el protocolo finaliza luego de realizarse ambas rondas, donde $k_a = k_b$, con $k_a = k_a^1 \,||\, k_a^2$ y $k_b = k_b^1 \,||\, k_b^2$.

Amplificación de privacidad:

- En L23, esta etapa pasa a ser llamada **Pre-amplificación de privacidad**, la cual es realizada al calcular $|P| = \binom{|b_1^X|}{2} + \binom{|b_1^Z|}{2}$ combinaciones en el paso 5.

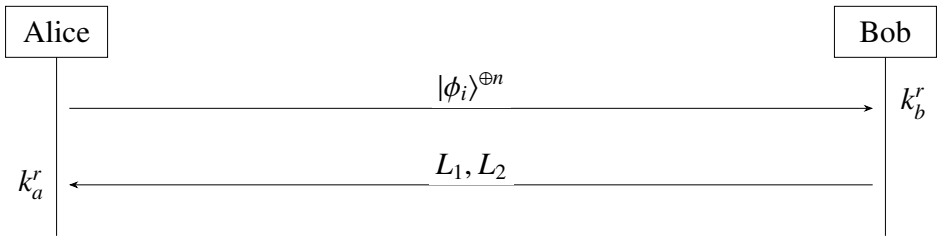


Figura 2.5: Diagrama simple de L23. Se asume que $r = 1$, es decir, una única ronda.

Tabla 2.16: Posibles orientaciones de las bases de medición de Bob, contenidas en las listas L_1 y L_2.

Listas	Orientaciones de las bases
L_1	$\begin{pmatrix} X & - \\ - & Z \end{pmatrix}, \begin{pmatrix} - & Z \\ X & - \end{pmatrix}$
L_2	$\begin{pmatrix} X & - \\ X & - \end{pmatrix}, \begin{pmatrix} - & Z \\ - & Z \end{pmatrix}$

Tabla 2.17: Tabla de derivación de la llave compartida. Cada frame de L_1 tiene asociado un bit secreto, dependiendo de la orientación de las bases de medición.

Orientación de las bases	k_i
$\begin{pmatrix} - & Z \\ X & - \end{pmatrix}$	0
$\begin{pmatrix} X & - \\ - & Z \end{pmatrix}$	1

Tabla 2.18: Definición de los frames f_1 y f_5

Notación	Frame
f_1	$\begin{pmatrix} \|+\rangle & \|1\rangle \\ \|-\rangle & \|0\rangle \end{pmatrix}$
f_5	$\begin{pmatrix} \|-\rangle & \|0\rangle \\ \|+\rangle & \|1\rangle \end{pmatrix}$

Tabla 2.19: Tabla de derivación de la orientación de las bases de un frame f_m. Con los resultados de esta tabla, Alice usa la Tabla 2.17 para derivar cada bit secreto. Se asume que el par extraído de f_m para generar f_T es el primero.

p	Condición	Orientación de las bases
$(X, -)$	$f_T \in L_1$	$\begin{pmatrix} - & Z \\ X & - \end{pmatrix}$
$(X, -)$	$f_T \in L_2$	$\begin{pmatrix} X & - \\ - & Z \end{pmatrix}$
$(-, Z)$	$f_T \in L_1$	$\begin{pmatrix} X & - \\ - & Z \end{pmatrix}$
$(-, Z)$	$f_T \in L_2$	$\begin{pmatrix} - & Z \\ X & - \end{pmatrix}$

En L23, las bases de medición no son expuestas, pero sí las listas L_1 y L_2, para cada una de las rondas. Además, el tamaño de la llave compartida aumenta de cuadráticamente, respecto de $|P|$:

$$T_{L23} = 2 \cdot \binom{|P|}{2} = |P|^2 - |P|.$$

Es importante notar que el tamaño es incluso mayor que en los protocolos LL20 y LL21, ya que no es necesario verificar que los frames construídos por Bob sean usables. En L23, el concepto de frame usable no existe, por lo que el SS tampoco se encuentra definido. Así pues, la seguridad de L23 se basa en la búsqueda de los frames f_1 y f_5.

La seguridad de L23 se ha analizado en el artículo original, declarando que existe una dificultad notable en recuperar la llave compartida a través de la fuerza bruta, ya que cada frame en L_1 posee dos orientaciones diferentes. Sin embargo, en el Capítulo 3, demostramos por primera vez que es posible recuperar la llave compartida en su totalidad, independiente del tipo de comunicación (con o sin errores).

3 | Criptoanálisis de la reconciliación por frames

Criptoanálisis se refiere al estudio de criptosistemas, con el objetivo de encontrar vulnerabilidades que permitan reducir su seguridad o comprometerla, sin el conocimiento de la información privada [1]. Algunos ejemplos de metodologías o vectores de ataque se describen a continuación, para diferentes esquemas criptográficos:

- **Cifrado simétrico**: Recuperar la llave privada que permite encriptar y desencriptar, a través de mensajes encriptados.
- **Cifrado asimétrico**: Recuperar la llave privada que permite desencriptar, a través de la llave pública.
- **Funciones hash**: Encontrar un mensaje $m' \neq m$, donde se cumple que $H(m') = H(m)$, siendo H la función hash. Lo anterior implica que es posible falsificar información.
- **Firmas digitales**: Recuperar la llave privada que permite firmar un mensaje, a través de la llave pública utilizada únicamente para validar su autenticidad. Lo anterior implica que es posible robar la identidad de clientes y servidores.
- **Intercambio de llaves**: Recuperar la llave compartida, utilizada comúnmente para cifrar o firmar mensajes, a través de la llave pública.

Para el caso de los protocolos QKD basados en frames, nos interesa el último punto. El ejercicio del criptoanálisis se va a centrar únicamente en el canal clásico, considerando la siguiente superficie de ataque:

1. El canal clásico se encuentra autenticado.
2. El atacante es pasivo, es decir, sólo es posible copiar la información clásica.

Así pues, los ataques que son descritos en el presente capítulo comparten el mismo escenario, donde el atacante sólo copia la información clásica e intenta recuperar la llave compartida, sin interactuar posteriormente con Alice o Bob. Nótese que la llave pública en los protocolos LL20 y LLS21 se compone del listado de *double matchings* m, la construcción de frames f, los *Sifting Strings* S y los frames f' que serán removidos de la llave compartida. Para el protocolo LL21, la lista S es reemplazada por C, que contiene los *Composed Sifting Strings*. Por último, la llave pública del protocolo L23 se compone de las listas L_1 y L_2.

3.1. Implementaciones

Las demostraciones de los ataques encontrados se realiza de manera teórica y experimental. Para lo segundo, un repositorio de GitHub fue desarrollado para incluir las implementaciones de cada protocolo QKD basado en frames, con sus correspondientes pruebas de concepto que demuestran la factibilidad de los ataques [35]. Resultados experimentales son presentados para ilustrar la probabilidad de éxito de cada ataque, si aplica, donde las especificaciones de la máquina utilizada se describen en la Tabla 3.1.

Dispositivo	Procesador	Núcleos	Frecuencia (MHz)	RAM
Laptop	Intel Core *i*7-11800	8	1900 (min), 4600 (max)	11GB

Tabla 3.1: Especificaciones de la máquina utilizada para ejecutar los ataques

Las implementaciones adjuntas en el repositorio no son completas, es decir, ninguno de los 4 protocolos (LL20, LLS21, LL21 y L23) fue implementado de manera completa, sino hasta la etapa de *Sifting* y considerando que la llave compartida coincide con los *Measurement Results* (MRs), debido a los siguientes puntos:

1. Dado que cada bit secreto depende del SS (CSS) y MR, el hecho de recuperar los MRs implica directamente recuperar los bits secretos, ya que los SS (CSS) son públicos. En L23, los bits secretos coinciden con los MRs.
2. El objetivo de los ataques es recuperar la llave compartida de Bob. Así pues, dado que la etapa de Reconciliación es efectuada por Alice y no existe un intercambio de información adicional aparte de los frames f_j que serán descartados de la llave compartida, podemos asumir que en presencia de errores en el canal cuántico, SS específicos serán descartados, por lo que no es necesario implementar la etapa de Reconciliación explícitamente.

Las etapas de Preparación y Medición son simuladas a través de Qiskit [36], un kit de desarrollo de código abierto que permite trabajar, en nuestro caso, con las compuertas cuánticas necesarias para inicializar los estados cuánticos $|\phi_i\rangle$. Por otro lado, la etapa de Envío no se encuentra implementada explícitamente, dado que Qiskit no es una herramienta que implemente la comunicación cuántica. Para simular la transmisión, el usuario puede configurar la probabilidad de un modelo de error de depolarización, el cual modifica la polarización del estado cuántico $|\phi_i\rangle$ de manera aleatoria en los ejes x, y o z en la esfera de Bloch. Así, dado que en la etapa de Reconciliación Alice descarta SS específicos, los resultados de los ataques en este escenario son independientes del valor del QBER.

Teniendo en cuenta la superficie de ataque y las consideraciones empleadas para implementar cada protocolo QKD basado en frames, las próximas secciones presentan los 3 ataques descubiertos que permiten comprometer la seguridad del canal clásico en su totalidad, es decir, es eficiente recuperar la llave compartida a través de la llave pública. Los ataques explotan una falla de diseño, correspondiente a la reutilización de los pares m_i al construir los frames f_j en los protocolos LL20, LLS21, LL21 y L23, debido al cálculo de las $\binom{|m|}{2}$, $\binom{|m|}{3}$ y $\binom{|P|}{2}$ combinaciones.

Por último, cada ataque se demuestra de manera teórica y experimental. Para lo segundo, se adjuntan gráficos para los ataques a los protocolos LL20, LLS21 y LL21, debido a que la recuperación exitosa de la llave compartida depende de la cantidad de frames f. Sin embargo, para el caso de L23, el ataque se realiza con éxito en el 100 % de los casos, independiente del largo de la lista L_1.

3.2. *Pairs reuse attack*

La reutilización de los pares m_i al construir los frames f_j del lado de Alice es una consecuencia del cálculo de las $\binom{|m|}{2}$ y $\binom{|m|}{3}$ combinaciones. Esta decisión de diseño permite explotar una vulnerabilidad, ya que si bien cada SS tiene dos posibles MR, la reutilización de pares permite determinar, para frames con SS específicos, sus MRs con un 100 % de probabilidad. Los protocolos LL20 y LLS21 son vulnerables al ataque de reutilización de pares.

3.2.1. LL20

Recordemos la construcción de *Sifting Strings* (SS) del lado de Bob:

$$SS = \text{1er sifting bit} \,\|\, \text{2do sifting bit} \,,\, \text{1er bit obtenido} \,\|\, \text{2do bit obtenido} \,.$$

La Tabla 2.6 describe todas las posibles orientaciones de las bases de medición de Bob para un frame f_j, respecto de su SS. Vamos a comenzar con el ataque asumiendo que se reutiliza un par m_i en dos frames f_1 y f_2, donde $S_1 = 11, 11$ y $S_2 = 10, 10$. Podemos notar que la única manera de que ambos frames compartan un par, es que coincida cualquier par de f_1 con el primer par de f_2, es decir:

$$\underset{S_1=11,11}{\begin{pmatrix}1_X & - \\ - & 1_Z\end{pmatrix},\begin{pmatrix}- & 1_Z \\ 1_X & -\end{pmatrix}} - \underset{S_2=10,10}{\begin{pmatrix}1_X & - \\ 0_X & -\end{pmatrix},\begin{pmatrix}1_X & - \\ - & 0_Z\end{pmatrix}}.$$

Si el primer par de f_1 coincide con el primer par de f_2, podemos concluir con un 100 % de probabilidad que la orientación de las bases de f_1 es $[(1_X, -), (-, 1_Z)]$, por lo que $MR_1 = 10$. Por otro lado, si el segundo par de f_1 coincide con el primer par de f_2, la orientación de f_1 es $[(-, 1_Z), (1_X, -)]$, lo que implica que $MR_1 = 11$.

De manera equivalente, podemos recuperar los MR de los frames f_1, si se reutilizan pares entre otros frames f_3, f_4 y f_5, donde $S_3 = 10, 01$, $S_4 = 01, 10$ y $S_5 = 01, 01$. Continuamos la demostración del ataque, incorporando el frame f_3:

$$\underset{S_1=11,11}{\begin{pmatrix}1_X & - \\ - & 1_Z\end{pmatrix},\begin{pmatrix}- & 1_Z \\ 1_X & -\end{pmatrix}} - \underset{S_3=10,01}{\begin{pmatrix}- & 0_Z \\ 1_X & -\end{pmatrix},\begin{pmatrix}0_X & - \\ 1_X & -\end{pmatrix}}.$$

A diferencia del caso de f_2, la reutilización de los pares puede realizarse únicamente en el segundo par de f_3. Así, si el primer par de f_1 coincide con el segundo de f_3, entonces $MR_1 = 10$. Por otro lado, si el segundo par de f_1 coincide con el de f_3, entonces $MR_1 = 11$.

Seguimos con el frame f_4. Si el primer par de f_1 coincide con el de f_4, entonces $MR_1 = 11$, y si el segundo par de f_1 es igual al primer par de f_4, entonces $MR_1 = 10$:

$$\underset{S_1=11,11}{\begin{pmatrix}1_X & - \\ - & 1_Z\end{pmatrix},\begin{pmatrix}- & 1_Z \\ 1_X & -\end{pmatrix}} - \underset{S_4=01,10}{\begin{pmatrix}- & 1_Z \\ - & 0_Z\end{pmatrix},\begin{pmatrix}- & 1_Z \\ 0_X & -\end{pmatrix}}.$$

Por último, incorporamos el frame f_5. Si el primer par de f_1 coincide con el segundo par de f_5, entonces $MR_1 = 11$, y si sus segundos pares son iguales, entonces $MR_1 = 10$:

$$\underset{S_1=11,11}{\begin{pmatrix}1_X & - \\ - & 1_Z\end{pmatrix},\begin{pmatrix}- & 1_Z \\ 1_X & -\end{pmatrix}} - \underset{S_5=01,01}{\begin{pmatrix}0_X & - \\ - & 1_Z\end{pmatrix},\begin{pmatrix}- & 0_Z \\ - & 1_Z\end{pmatrix}}.$$

De esta manera, podemos recuperar prácticamente todos los bits secretos asociados a los frames f_j donde $S_j = 11, 11$, con $MR_j = 10$ o $MR_j = 11$. Luego, aplicamos una metodología equivalente para recuperar frames f_6, donde $S_6 = 00, 11$:

$$\underset{S_6=00,11}{\begin{pmatrix}1_X & - \\ 1_X & -\end{pmatrix},\begin{pmatrix}- & 1_Z \\ - & 1_Z\end{pmatrix}} - \underset{S_2=10,10}{\begin{pmatrix}1_X & - \\ 0_X & -\end{pmatrix},\begin{pmatrix}1_X & - \\ - & 0_Z\end{pmatrix}},$$

$$\underset{S_6=00,11}{\begin{pmatrix}1_X & - \\ 1_X & -\end{pmatrix},\begin{pmatrix}- & 1_Z \\ - & 1_Z\end{pmatrix}} - \underset{S_3=10,01}{\begin{pmatrix}- & 0_Z \\ 1_X & -\end{pmatrix},\begin{pmatrix}0_X & - \\ 1_X & -\end{pmatrix}},$$

$$\underset{S_6=00,11}{\begin{pmatrix}1_X & - \\ 1_X & -\end{pmatrix},\begin{pmatrix}- & 1_Z \\ - & 1_Z\end{pmatrix}} - \underset{S_4=01,10}{\begin{pmatrix}- & 1_Z \\ - & 0_Z\end{pmatrix},\begin{pmatrix}- & 1_Z \\ 0_X & -\end{pmatrix}},$$

$$\underset{S_6=00,11}{\begin{pmatrix}1_X & - \\ 1_X & -\end{pmatrix},\begin{pmatrix}- & 1_Z \\ - & 1_Z\end{pmatrix}} - \underset{S_5=01,01}{\begin{pmatrix}0_X & - \\ - & 1_Z\end{pmatrix},\begin{pmatrix}- & 0_Z \\ - & 1_Z\end{pmatrix}}.$$

Así pues, si un frame f_6 comparte un par con un f_2 o f_3, entonces $MR_6 = 00$. Por otro lado, si se comparte un par con frames f_4 o f_5, entonces $MR_6 = 01$.

Para finalizar el ataque de reutilización de pares en LL20, tomamos el listado de frames recuperados, denotado como f^{rec}, y en el caso de existir un frame f_6 que comparte un par con un frame $f_1 \in f^{rec}$, es decir:

$$\underset{S_1=11,11}{\begin{pmatrix} 1_X & - \\ - & 1_Z \end{pmatrix}} - \underset{S_6=00,11}{\begin{pmatrix} 1_X & - \\ 1_X & - \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ - & 1_Z \end{pmatrix}},$$

o bien,

$$\underset{S_1=11,11}{\begin{pmatrix} - & 1_Z \\ 1_X & - \end{pmatrix}} - \underset{S_6=00,11}{\begin{pmatrix} 1_X & - \\ 1_X & - \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ - & 1_Z \end{pmatrix}},$$

entonces podemos concluir que $MR_6 = 00$ o $MR_6 = 01$, respectivamente. Así pues, el Algoritmo 1 describe el pseudocódigo del ataque de reutilización de pares en LL20, donde nos referiremos a los pares de un frame f_j, en orden, como f_j^1 y f_j^2. Adicionalmente, las Figuras 3.1 y 3.2 ilustran gráficos que describen la fracción de bits recuperados al ejecutar el ataque y el tiempo de ejecución, respecto a los *double matchings*. Nótese que nos referimos a los bits recuperados como los MRs, según lo comentado en la Sección 3.2.

La recuperación total de la llave compartida ocurre cuando consideramos la etapa de Reconciliación. Recordemos que, en ella, Alice tiene que descartar tanto a los frames auxiliares como a los que cumplen con que $S_j \in \{00, 00; 10, 01; 10, 10; 01, 01; 01, 10\}$. Este conjunto corresponde precisamente a los SS que no se pudieron recuperar en el *Pairs reuse attack*, lo que implica que la fracción de bits recuperados aumenta considerablemente, a tal punto de poder obtener, en su totalidad, la llave criptográfica que comparten Alice y Bob. En la Figura 3.3 se demuestra lo descrito previamente, alcanzando el 100 % de bits recuperados.

Para finalizar, es importante demostrar por qué la reutilización de pares entre dos frames denominados como f_1 y f_2 no es factible cuando $S_1, S_2 \neq 11, 11$ o $S_1, S_2 \neq 00, 11$. Consideremos el siguiente ejemplo:

$$\underset{S_2=10,10}{\begin{pmatrix} 1_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 1_X & - \\ - & 0_Z \end{pmatrix}} - \underset{S_5=01,01}{\begin{pmatrix} 0_X & - \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ - & 1_Z \end{pmatrix}}.$$

Podemos notar que, a pesar de que ambos frames pueden compartir pares, que sería el segundo par de f_2 con el primer par de f_5, no es posible distinguir el MR de ninguno de ellos. Dicho de otra manera, ambos MRs son posibles para cada frame.

3.2.2. LLS21

Comenzamos el ejercicio de criptoanálisis del protocolo QKD basado en frames de dimensionalidad 3 x 2, analizando las posibles orientaciones de las bases de Bob, dependiendo del *Sifting String* (SS):

SS = 1er sifting bit || 2do sifting bit || 3er sifting bit, 1er bit obtenido || 2do bit obtenido || 3er bit obtenido.

Las Tablas 2.9, 2.10, 2.11 y 2.12 muestran todas las posibles orientaciones de las bases de un frame f_j, dado S_j. El ataque de reutilización de pares en LLS21 consiste en dos fases: *subframes reuse* y *single pair reuse*. Dado que Alice calcula $\binom{|m|}{3}$ combinaciones, entonces existirán pares compartidos entre diferentes frames f_j, al igual que en LL20. Sin embargo, también se reutilizan *subframes*, es decir, frames 2 x 2 dentro de la matriz 3 x 2.

Siguiendo la misma tónica que en LL20, vamos a considerar dos frames denotados como f_1 y f_2 donde, por ejemplo, $S_1 = 111, 110$ y $S_2 = 101, 111$. Si entre ellos existe un subframe en común, entonces podemos distinguir los MRs de ambos frames, ya que:

$$\underset{S_1=111,110}{\begin{pmatrix} 1_X & - \\ - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 1_X & - \\ 0_X & - \end{pmatrix}} - \underset{S_2=101,111}{\begin{pmatrix} - & 1_Z \\ - & 1_Z \\ 1_X & - \end{pmatrix}, \begin{pmatrix} 1_X & - \\ - & 1_Z \\ - & 1_Z \end{pmatrix}}.$$

Vamos a denotar a los subframes de cada frame f_j, en orden, como $f_j^{s_1}$, $f_j^{s_2}$ y $f_j^{s_3}$. Nótese que el tercer subframe de f_j corresponde al frame 2 x 2 que se construye juntando el primer y tercer par. Así, si $f_1^{s_1} = f_2^{s_1}$ (marcados en rojo), entonces $MR_1 = MR_2 = 110$. Luego, si $f_1^{s_1} = f_2^{s_2}$ (marcados en cian), entonces $MR_1 = 111$ y $MR_2 = 101$. Durante la fase de *subframes reuse*, podemos recuperar los MRs de dos frames siempre y cuando las posiciones de los subframes compartidos sean distinguibles. Es importante notar que la reutilización de subframes es un ataque con mayor severidad que la reutilización de pares en LL20, ya que podemos recuperar ambos MRs.

Algorithm 1 *Pairs reuse attack* en LL20

Input: frames f y *Sifting Strings* S.
Output: MRs de frames f_j tales que $S_j = 11, 11$ y $S_j = 00, 11$.

$f^{rec} \leftarrow []$
$f_{SS} \leftarrow [f_j \in f \mid S_j = SS]$
for $f_j, f_k \in f_{11,11} \times (f_{10,10} \parallel f_{10,01})$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **then**
    f^{rec}.add$(f_j, 10)$
  else if $f_j^2 = f_k^1$ **or** $f_j^2 = f_k^2$ **then**
    f^{rec}.add$(f_j, 11)$
  end if
end for
for $f_j, f_k \in f_{11,11} \times (f_{01,10} \parallel f_{01,01})$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **then**
    f^{rec}.add$(f_j, 11)$
  else if $f_j^2 = f_k^1$ **or** $f_j^2 = f_k^2$ **then**
    f^{rec}.add$(f_j, 10)$
  end if
end for
for $f_j, f_k \in f_{00,11} \times (f_{10,10} \parallel f_{10,01} \parallel f_{01,10} \parallel f_{01,01})$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **then**
    if $S_j = 10, 10$ **or** $S_j = 10, 01$ **then**
      f^{rec}.add$(f_j, 00)$
    else if $S_j = 01, 10$ **or** $S_j = 01, 01$ **then**
      f^{rec}.add$(f_j, 01)$
    end if
  end if
end for
for $f_j, f_k \in f^{rec} \times f_{00,11}$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **then**
    if $MR_j = 10$ **then**
      f^{rec}.add$(f_k, 00)$
    else if $MR_j = 11$ **then**
      f^{rec}.add$(f_j, 01)$
    end if
  end if
end for
return f^{rec}

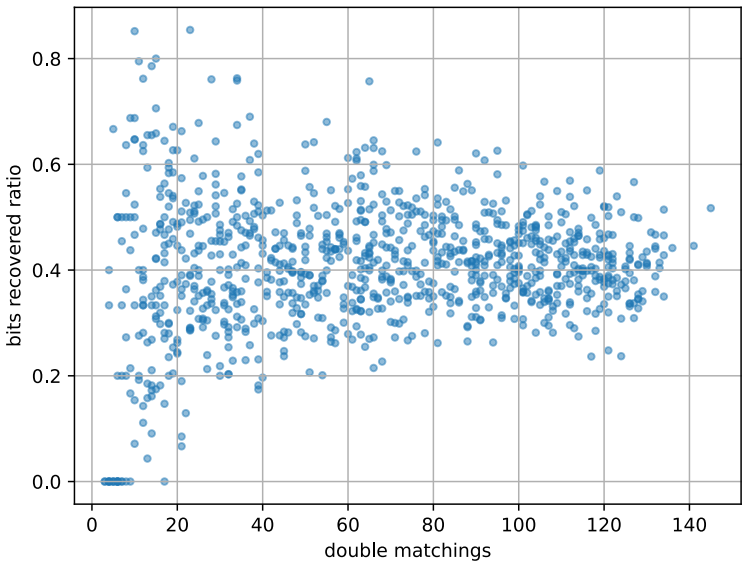


Figura 3.1: Razón entre la cantidad de bits recuperados y los totales, luego de ejecutar el *Pairs reuse attack* en LL20 respecto a los *double matchings*, utilizando 1000 muestras aleatorias. En este caso, se asume que la transmisión cuántica se realiza sin errores, por lo que todos los SS son considerados.

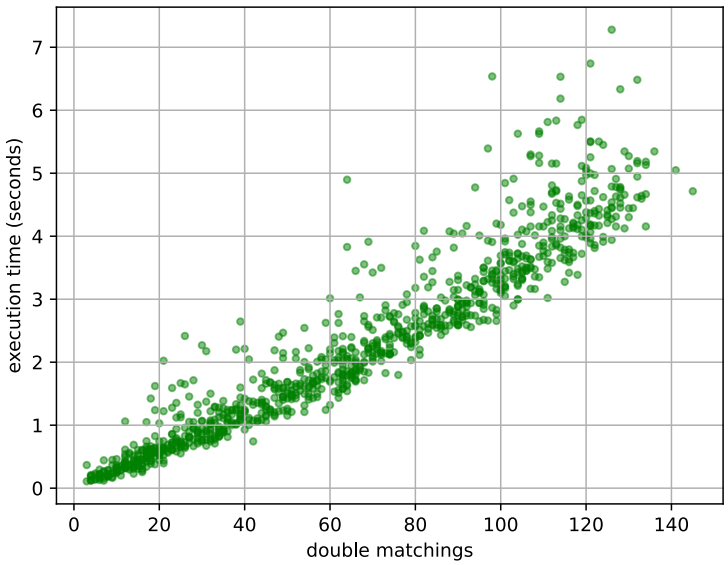


Figura 3.2: Tiempo de ejecución (en segundos) del *Pairs reuse attack* en LL20 respecto a los *double matchings*, utilizando 1000 muestras aleatorias. En este caso, se asume que la transmisión cuántica se realiza sin errores, por lo que todos los SS son considerados.

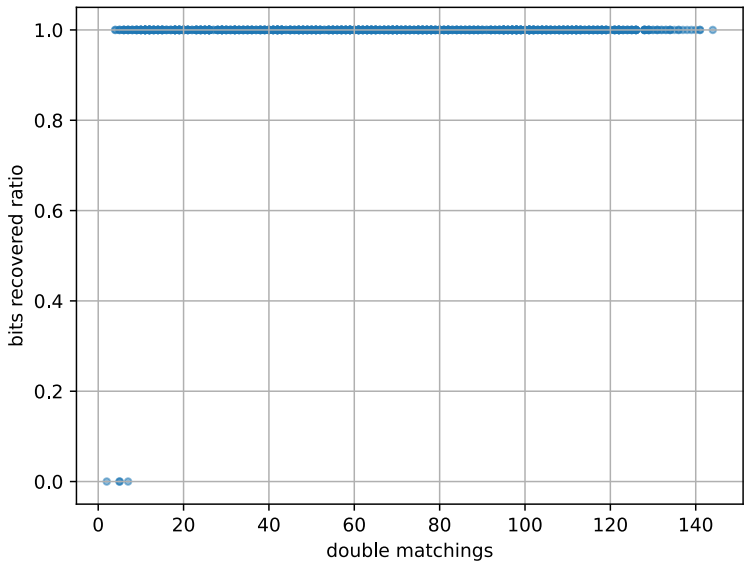


Figura 3.3: Razón entre la cantidad de bits recuperados y los totales, luego de ejecutar el *Pairs reuse attack* en LL20 respecto a los *double matchings*, utilizando 1000 muestras aleatorias. Los resultados consideran que la transmisión cuántica se realiza con errores, por lo que sólo los SS que cumplen con $S_j = 00, 11$ o $S_j = 11, 11$ son considerados.

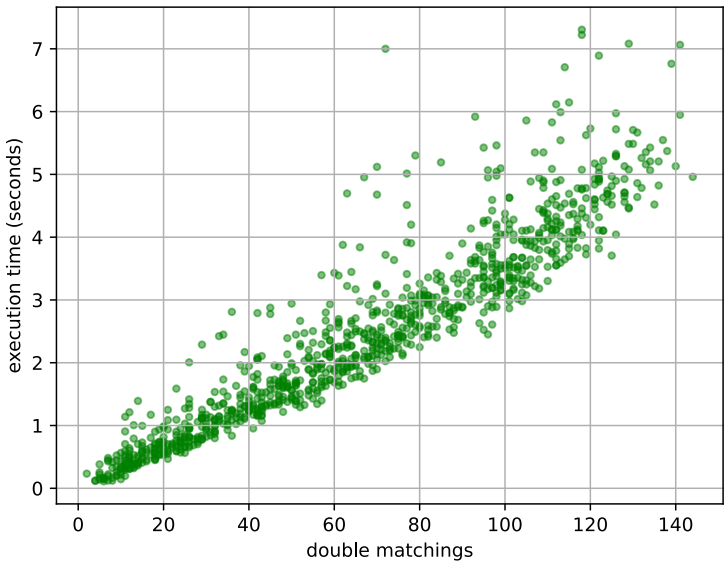


Figura 3.4: Tiempo de ejecución (en segundos) del *Pairs reuse attack* en LL20 respecto a los *double matchings*, utilizando 1000 muestras aleatorias. Los resultados consideran que la transmisión cuántica se realiza con errores, por lo que sólo los SS que cumplen con $S_j = 00, 11$ o $S_j = 11, 11$ son considerados.

Luego de la fase de *subframes reuse*, el atacante obtiene f^{rec}, correspondiente al listado de frames donde sus MRs fueron recuperados. La orientación de las bases de medición asociadas a cada MR de un frame $f_j \in f^{rec}$ se denota como O_j, donde O_j^1, O_j^2 y O_j^3 corresponden a la orientación de cada par $m_i \in f_j$. Para un $f_k \notin f^{rec}$, dado que su orientación se desconoce, utilizaremos un subíndice adicional para referirnos a su posible MR, es decir, O_{k_1} y O_{k_2}. En la fase de *single pair reuse*, generalizamos lo obtenido en LL20, aplicándolo para cualquier par de frames donde al menos un par se reutilice. Pongamos un ejemplo:

$$\underbrace{\begin{pmatrix} 1_X & - \\ - & 1_Z \\ - & 0_Z \end{pmatrix}}_{O_j} - \underbrace{\begin{pmatrix} - & 1_Z \\ - & 1_Z \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 1_X & - \\ 1_X & - \\ - & 0_Z \end{pmatrix}}_{O_{k_1}, O_{k_2}}.$$

Aquí, conocemos el MR de un frame $f_j \in f^{rec}$ tal que $S_j = 111, 110$. Si $f_j^1 = f_k^2$, $f_j^1 = f_k^2$ o $f_j^3 = f_k^3$, donde $f_k \notin f^{rec}$ y $S_k = 001, 110$, entonces $MR_k = 100$. Por otro lado, si $f_j^2 = f_k^1$ o $f_j^2 = f_k^2$, entonces $MR_k = 101$. La distinguibilidad puede extenderse a un caso general para los frames 3 x 2, donde la recuperación del MR del frame f_k puede realizarse si y solo si se cumple que, al compartir un par en las posiciones x e y:

$$O_j^x = O_{k_1}^y, O_{k_1}^y \neq O_{k_2}^y, \tag{3.1}$$

o bien,

$$O_j^x = O_{k_2}^y, O_{k_1}^y \neq O_{k_2}^y. \tag{3.2}$$

Podemos extender el resultado si los frames f_j y f_k comparten 2 pares, en las posiciones x, v e y, w respectivamente. La recuperación del MR de f_k se realiza si y solo si se cumple una de las siguientes 4 ecuaciones:

$$O_j^x = O_{k_1}^y, O_{k_1}^y \neq O_{k_2}^y, \tag{3.3}$$

$$O_j^x = O_{k_2}^y, O_{k_1}^y \neq O_{k_2}^y, \tag{3.4}$$

$$O_j^v = O_{k_1}^w, O_{k_1}^w \neq O_{k_2}^w, \tag{3.5}$$

$$O_j^v = O_{k_2}^w, O_{k_1}^w \neq O_{k_2}^w. \tag{3.6}$$

Combinando las fases *subframes reuse* y *single pair reuse*, podemos recuperar la llave compartida entre Alice y Bob en su totalidad. El Algoritmo 2 describe el pseudocódigo de las fases *subframes reuse* y *single pair reuse*. Además, las Figuras 3.5 y 3.6 muestran los resultados de ejecución del ataque, considerando un escenario donde no existen errores en el canal cuántico, por lo que todos los SS son considerados. La gráfica demuestra que el ataque es factible, por lo que en presencia de errores en el canal cuántico, también lo será, ya que en la etapa de Reconciliación, diversos SS serán descartados.

3.3. *Conjugate-Pairs reuse attack*

El protocolo LL21 es vulnerable al *Conjugate-Pairs reuse attack*, una variante del *Pairs reuse attack* donde además se consideran los casos donde el MR es único y los *double matchings* en '0' pueden ser recuperados. Así, al igual que en LLS21, el ataque es tan efectivo que permite recuperar la llave criptográfica en su totalidad. Recordemos la construcción de los *Composed Sifting Strings* (CSS):

$$CSS = \text{1er sifting bit} \,\|\, \text{2do sifting bit} \,,\, \text{1er sifting bit conjugado} \,\|\, \text{2do sifting bit conjugado} \,.$$

La Tabla 2.14 describe todas las posibles orientaciones de las bases de Bob en un frame f_j, dado C_j. Lo primero y más importante es notar que el *Pairs reuse attack* de LL20 sigue siendo aplicable en LL21, ya que:

$$\underset{C_j=11,00}{\begin{pmatrix} 1_X & - \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 1_X & - \end{pmatrix}} - \underset{C_k=10,10}{\begin{pmatrix} 1_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ 1_X & - \end{pmatrix}},$$

$$\underset{C_j=11,00}{\begin{pmatrix} 1_X & - \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 1_X & - \end{pmatrix}} - \underset{C_k=10,01}{\begin{pmatrix} 1_X & - \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ 1_X & - \end{pmatrix}},$$

$$\underset{C_j=11,00}{\begin{pmatrix} 1_X & - \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 1_X & - \end{pmatrix}} - \underset{C_k=01,01}{\begin{pmatrix} - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ - & 1_Z \end{pmatrix}},$$

$$\underset{C_j=11,00}{\begin{pmatrix} 1_X & - \\ - & 1_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 1_X & - \end{pmatrix}} - \underset{C_k=01,10}{\begin{pmatrix} - & 1_Z \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ - & 1_Z \end{pmatrix}}.$$

Al igual que en LLS21, es posible recuperar los MRs asociados a ambos frames f_j y f_k, ya que los CSS permiten distinguir las bases de medición en los casos donde se realizan *double matchings* en '0'. Veamos un ejemplo con $S_j = C_j = 10, 01$:

$$\underset{S_j=10,01}{\begin{pmatrix} - & 0_Z \\ 1_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ 1_X & - \end{pmatrix}} \implies \underset{C_j=10,01}{\begin{pmatrix} 1_X & - \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ 1_X & - \end{pmatrix}}.$$

Así pues, si un frame f_j con $C_j = 10, 01$ comparte un par con otro frame f_k con $C_k = 10, 10$, entonces podemos recuperar los MRs asociados a f_j con total certeza, ya que:

$$\underset{C_j=10,01}{\begin{pmatrix} 1_X & - \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ 1_X & - \end{pmatrix}} - \underset{C_k=10,10}{\begin{pmatrix} 1_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ 1_X & - \end{pmatrix}}.$$

Sin embargo, hay un detalle, y es que también podemos recuperar los MRs asociados a los frames f_k con $C_k = 10, 10$, pero no por el hecho de que exista una reutilización de los pares, sino más bien porque existe sólo un único MR posible para este CSS. Lo anterior corresponde a un problema de seguridad, que fue descrito previamente en el artículo original del protocolo L23, declarando que LL21 es inseguro. En el artículo original de LL21, los autores declaran que los frames f_j donde $C_j = 10, 10$ y $C_j = 01, 01$ deben ser descartados, sin embargo lo anterior no impide que estos frames sean utilizados para recuperar potenciales frames que formarán parte de la llave compartida.

Para dar completitud al análisis, a continuación se describen el resto de frames donde la reutilización de pares conjugados permite recuperar sus MRs:

$$\underset{C_j=10,01}{\begin{pmatrix} 1_X & - \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ 1_X & - \end{pmatrix}} - \underset{C_k=01,01}{\begin{pmatrix} - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ - & 1_Z \end{pmatrix}},$$

$$\underset{C_j=01,10}{\begin{pmatrix} - & 1_Z \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ - & 1_Z \end{pmatrix}} - \underset{C_k=10,10}{\begin{pmatrix} 1_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ 1_X & - \end{pmatrix}},$$

$$\underset{C_j=01,10}{\begin{pmatrix} - & 1_Z \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ - & 1_Z \end{pmatrix}} - \underset{C_k=01,01}{\begin{pmatrix} - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ - & 1_Z \end{pmatrix}},$$

Algorithm 2 *Pairs reuse attack* en LLS21

Input: frames f y *Sifting Strings* S.
Input: Diccionario D que contiene los subframes que permiten recuperar los MRs de un par de frames f_j, f_k, en función de S_j y S_k.
Input: Diccionario P que contiene los pares que permiten recuperar el MR de un frame f_k mediante $f_j \in f^{rec}$, en función de S_j y S_k.
Output: frames recuperados f^{rec}.

$f^{rec} \leftarrow []$
for all $j, k \in \binom{|f|}{2}$ **do**
  if $f_j, f_k \notin f^{rec}$ **then**
    $C \leftarrow D(S_j, S_k)$
    if C **then**
      $s_j \leftarrow$ *subframes* de f_j
      $s_k \leftarrow$ *subframes* de f_k
      for all $i_j, i_k, MR_j, MR_k \in C$ **do**
        if $s_j[i_j] = s_k[i_k]$ **then**
          f^{rec}.add(f_j, MR_j)
          f^{rec}.add(f_k, MR_k)
        end if
      end for
    end if
  end if
end for
for all $j, k \in \binom{|f|}{2}$ **do**
  if $f_j \in f^{rec}$ **and** $f_k \notin f^{rec}$ **then**
    $C \leftarrow P(S_j, S_k)$
    for all $i, MR_j, MR_k \in C$ **do**
      if $f^{rec}[f_j] = MR_j$ **and** $f_j^i = f_k^i$ **then**
        f^{rec}.add(f_k, MR_k)
      end if
    end for
  end if
end for
return f^{rec}

$$\underset{C_j=00,11}{\begin{pmatrix} 0_X & - \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ 0_X & - \end{pmatrix}} - \underset{C_k=10,10}{\begin{pmatrix} 1_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ 1_X & - \end{pmatrix}},$$

$$\underset{C_j=00,11}{\begin{pmatrix} 0_X & - \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ 0_X & - \end{pmatrix}} - \underset{C_k=01,01}{\begin{pmatrix} - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ - & 1_Z \end{pmatrix}},$$

$$\underset{C_j=00,00}{\begin{pmatrix} 1_X & - \\ 1_X & - \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ - & 1_Z \end{pmatrix}} - \underset{C_k=10,10}{\begin{pmatrix} 1_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 0_X & - \\ 1_X & - \end{pmatrix}}.$$

$$\underset{C_j=00,00}{\begin{pmatrix} 1_X & - \\ 1_X & - \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ - & 1_Z \end{pmatrix}} - \underset{C_k=01,01}{\begin{pmatrix} - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 0_Z \\ - & 1_Z \end{pmatrix}}.$$

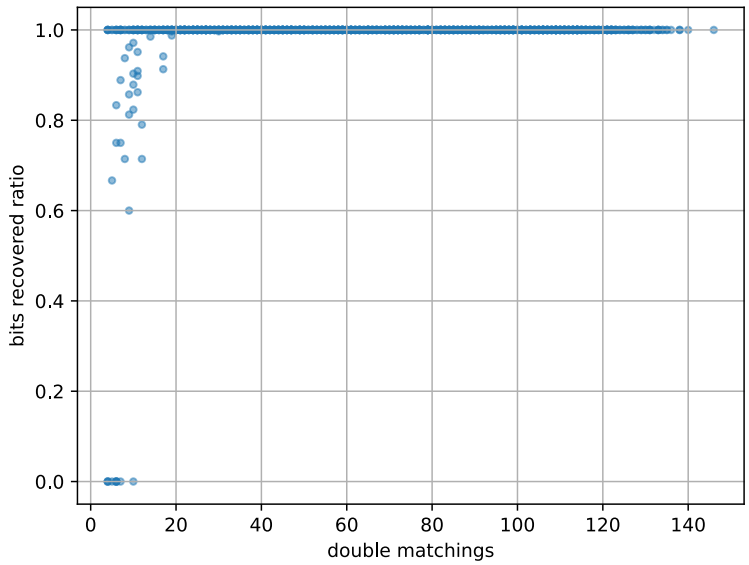


Figura 3.5: Razón entre la cantidad de bits recuperados y los totales, luego de ejecutar el *Pairs reuse attack* en LLS21 respecto a los *double matchings*, utilizando 1000 muestras aleatorias. En este caso, se asume que la transmisión cuántica se realiza sin errores, por lo que todos los SS son considerados.

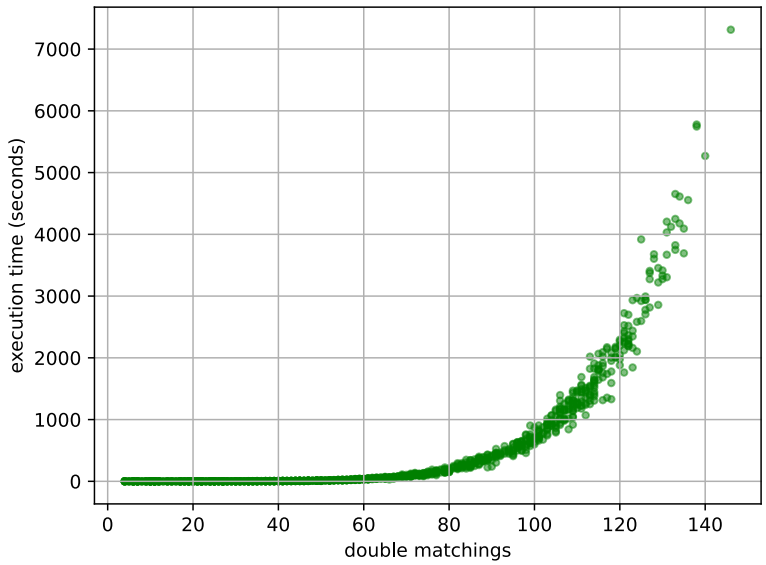


Figura 3.6: Tiempo de ejecución (en segundos) del *Pairs reuse attack* en LLS21 respecto a los *double matchings*, utilizando 1000 muestras aleatorias. En este caso, se asume que la transmisión cuántica se realiza sin errores, por lo que todos los SS son considerados.

Los Algoritmos 3 y 4 describen el pseudocódigo para el *Conjugate-Pairs reuse*, el cual permite recuperar la llave compartida en su totalidad. Adicionalmente, las Figuras 3.5 y 3.8 ilustran los resultados de ejecución del ataque, donde se demuestra que el protocolo LL21 es inseguro. La implementación se apoya del *Pairs reuse attack* para recuperar los frames f_j donde $C_j = 11, 00$, para posteriormente utilizar los f^{rec} de la misma manera que en el caso de LL20. Así, la recuperación de los f_j y f_k se realizan en iteraciones separadas.

3.4. *Avalanche-Effect attack*

En L23, a pesar de no estar definidos los SS o CSS, nuestro objetivo sigue siendo recuperar los MRs de frames $f_j \in L_1$, los cuales coinciden con los bits secretos. La seguridad de L23 se basa en la dificultad de obtener las bases de medición de Bob, Alice es la única que puede recuperarlas ya que ella conoce cuáles frames son f_1 y f_5 dentro de la lista L_1. Sin embargo, es importante recordar que la derivación de la llave compartida se realiza con un único frame, a través de su pivote p.

Podríamos pensar que la seguridad de L23 se reduce a un problema de búsqueda de p dentro de la lista L_1. En realidad, la formulación del problema es mucho más sencilla, ya que independiente del pivote que se escoja en un frame f_1 o f_5, la llave criptográfica de Alice coincidirá con la de Bob. Así pues, dado que el pivote p pudo ser medido en la base X o Z, podemos asumir uno de los casos, obteniendo una posible llave compartida. A continuación, vamos a demostrar que, si la suposición es incorrecta, la llave compartida resultante se encuentra invertida (en bits) respecto a la real.

Supongamos que tenemos los siguientes frames en L_1 y L_2:

$$L_1 = \left[\begin{pmatrix} 3 & 7 \\ 5 & 2 \end{pmatrix}, \begin{pmatrix} 5 & 2 \\ 6 & 9 \end{pmatrix}, \begin{pmatrix} 1 & 4 \\ 3 & 2 \end{pmatrix}, \dots \right],$$

$$L_2 = \left[\begin{pmatrix} 6 & 9 \\ 1 & 4 \end{pmatrix}, \begin{pmatrix} 3 & 2 \\ 5 & 9 \end{pmatrix}, \begin{pmatrix} 3 & 7 \\ 6 & 9 \end{pmatrix}, \dots \right].$$

Cada color representa pares iguales (que utilizaremos para el ejemplo), situación esperable dadas las $\binom{|P|}{2}$ combinaciones que realiza Bob. Vamos a definir el pivote p como el primer par del primer elemento de L_1. Del punto de vista de un atacante, las bases de medición de p se desconocen, por lo que asumiremos que se encuentra medido en la base Z. Así, vamos a tomar el primer par del segundo frame de L_1 para construir un frame de prueba f_T:

$$f_T = \begin{pmatrix} 3 & 7 \\ 5 & 2 \end{pmatrix} \in L_1 \implies \begin{pmatrix} 5 & 2 \\ 6 & 9 \end{pmatrix} = \begin{pmatrix} X & - \\ - & Z \end{pmatrix} = 1.$$

Sin embargo, si asumimos que p se encuentra medido en la base X, el bit se invierte:

$$f_T = \begin{pmatrix} 3 & 7 \\ 5 & 2 \end{pmatrix} \in L_1 \implies \begin{pmatrix} 5 & 2 \\ 6 & 9 \end{pmatrix} = \begin{pmatrix} - & Z \\ X & - \end{pmatrix} = 0.$$

Luego, para un frame de prueba f_T que se encuentra en L_2, se obtiene un comportamiento equivalente, asumiendo que p es medido en la base Z:

$$f_T = \begin{pmatrix} 3 & 7 \\ 6 & 9 \end{pmatrix} \in L_2 \implies \begin{pmatrix} 5 & 2 \\ 6 & 9 \end{pmatrix} = \begin{pmatrix} X & - \\ - & Z \end{pmatrix} = 1.$$

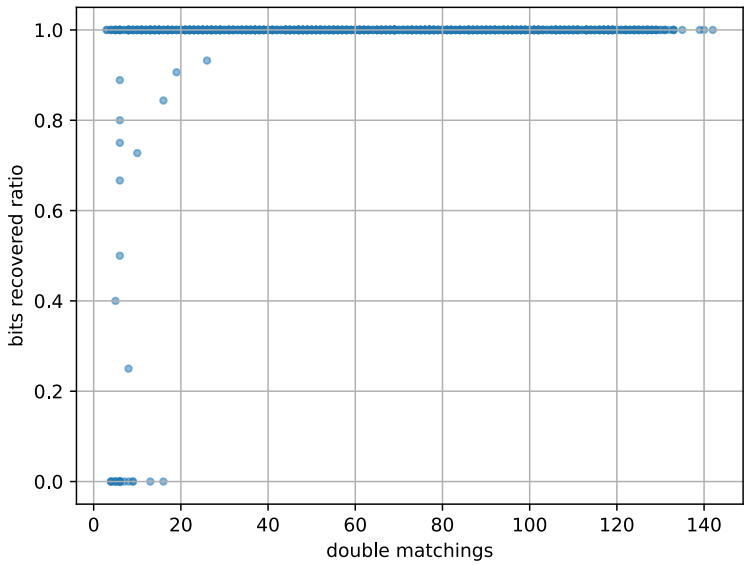


Figura 3.7: Razón entre la cantidad de bits recuperados y los totales, luego de ejecutar el *Conjugate-Pairs reuse attack* en LL21 respecto a los *double matchings*, utilizando 1000 muestras aleatorias. En este caso, se asume que la transmisión cuántica se realiza sin errores

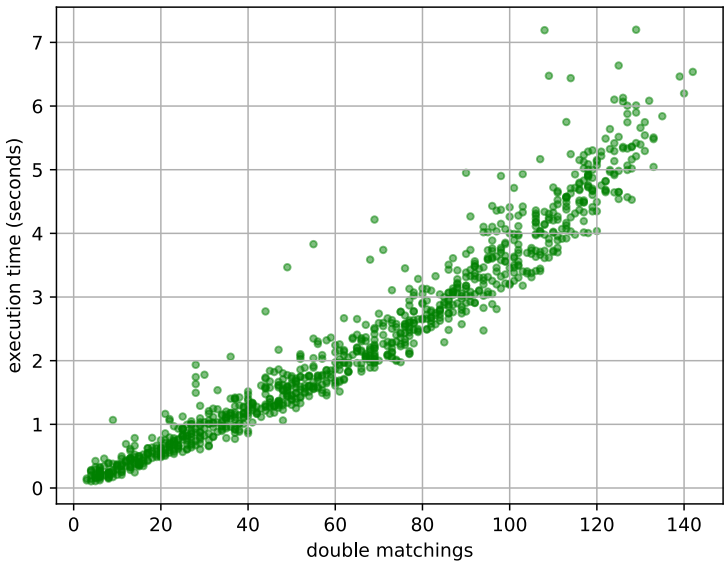


Figura 3.8: Tiempo de ejecución (en segundos) del *Conjugate-Pairs reuse attack* en LL21 respecto a los *double matchings*, utilizando 1000 muestras aleatorias. En este caso, se asume que la transmisión cuántica se realiza sin errores

Algorithm 3 *Conjugate-Pairs reuse attack* en LL21

Input: frames f y *Composed Sifting Strings* C.
Output: frames recuperados f^{rec}..

$f^{rec} \leftarrow []$
$f_{CSS} \leftarrow [f_j \in f \mid C_j = CSS]$
for $f_j, f_k \in f_{10,01} \times f_{10,10}$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **then**
    f^{rec}.add$(f_j, 11)$
  else if $f_j^2 = f_k^1$ **or** $f_j^2 = f_k^2$ **then**
    f^{rec}.add$(f_j, 10)$
  end if
end for
for $f_j, f_k \in f_{01,10} \times f_{10,10}$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **then**
    f^{rec}.add$(f_j, 10)$
  else if $f_j^2 = f_k^1$ **or** $f_j^2 = f_k^2$ **then**
    f^{rec}.add$(f_j, 11)$
  end if
end for
for $f_j, f_k \in f_{00,11} \times f_{10,10}$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **then**
    f^{rec}.add$(f_j, 11)$
  else if $f_j^2 = f_k^1$ **or** $f_j^2 = f_k^2$ **then**
    f^{rec}.add$(f_j, 10)$
  end if
end for
for $f_j, f_k \in f_{00,00} \times f_{10,10}$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **or** $f_j^2 = f_k^1$ **or** $f_j^2 = f_k^2$ **then**
    f^{rec}.add$(f_j, 00)$
  end if
end for
for $f_j, f_k \in f_{10,01} \times f_{01,01}$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **then**
    f^{rec}.add$(f_j, 10)$
  else if $f_j^2 = f_k^1$ **or** $f_j^2 = f_k^2$ **then**
    f^{rec}.add$(f_j, 11)$
  end if
end for
for $f_j, f_k \in f_{01,10} \times f_{01,01}$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **then**
    f^{rec}.add$(f_j, 10)$
  else if $f_j^2 = f_k^1$ **or** $f_j^2 = f_k^2$ **then**
    f^{rec}.add$(f_j, 11)$
  end if
end for
for $f_j, f_k \in f_{00,11} \times f_{01,01}$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **then**
    f^{rec}.add$(f_j, 10)$
  else if $f_j^2 = f_k^1$ **or** $f_j^2 = f_k^2$ **then**
    f^{rec}.add$(f_j, 11)$
  end if
end for

Algorithm 4 *Cont.*

for $f_j, f_k \in f_{00,00} \times f_{01,01}$ **do**
  if $f_j^1 = f_k^1$ **or** $f_j^1 = f_k^2$ **or** $f_j^2 = f_k^1$ **or** $f_j^2 = f_k^2$ **then**
    f^{rec}.add$(f_j, 01)$
  end if
end for
for $f_j \in f_{10,10}$ **do**
  f^{rec}.add$(f_j, 00)$
end for
for $f_j \in f_{01,01}$ **do**
  f^{rec}.add$(f_j, 01)$
end for
return f^{rec}

Por último, si asumimos que el pivote p fue medido en la base X, el bit se invierte:

$$f_T = \begin{pmatrix} 3 & 7 \\ 6 & 9 \end{pmatrix} \in L_2 \implies \begin{pmatrix} 5 & 2 \\ 6 & 9 \end{pmatrix} = \begin{pmatrix} - & Z \\ X & - \end{pmatrix} = 0.$$

Entonces, si tomamos un frame $f_j \in L_1$ aleatorio y definimos el pivote p, el atacante puede calcular 2^r llaves posibles, donde r es el número de rondas. Para una ejecución completa del protocolo L23, $r = 2$, por lo que existen sólo $2^2 = 4$ llaves posibles.

Dado que sólo se requiere de un único frame $f_j \in L_1$ para recuperar k_a o k_b, denominamos a este ataque como *Avalanche Effect*. El Algoritmo 5 describe el pseudocódigo para el ataque, donde la llave compartida puede recuperarse completamente, independiente del tipo de transmisión (con o sin errores) y la cantidad de *double matchings*.

Algorithm 5 *Avalanche Effect attack*. Se asume que p se encuentra medido en la base Z

Input: Listas L_1, L_2
Output: k_a o k_b

$k \leftarrow []$
$f_j \leftarrow$ random(L_1)
$p \leftarrow f_j^1$
for all $f_j \in L_1$ **do**
  if $p \neq f_j^1$ **then**
    $f_T \leftarrow [p, f_j^1]$
    if $f_T \in L_1$ **then**
      $k_i \leftarrow 1$
    else if $f_T \in L_2$ **then**
      $k_i \leftarrow 0$
    end if
  else
    $k_i \leftarrow 0$
  end if
end for
return k

4 | Directrices de Seguridad

En el Capítulo 3 demostramos, de manera teórica y experimental, que los protocolos LL20, LLS21, LL21 y L23 no son seguros y por ende no deben ser utilizados para ningún propósito industrial o entorno productivo de una aplicación digital. Sin embargo, es importante tener en cuenta que las vulnerabilidades encontradas explotan aspectos que pueden ser corregibles como la reutilización de pares, representando hasta el momento un *trade-off* entre seguridad y el aumento cuadrático (o cúbico) de la llave compartida, además de la probabilidad de detectar y corregir errores.

Así pues, en este capítulo, directrices de seguridad son descritas, con el propósito de ser la primera referencia para el desarrollo de protocolos QKD basados en frames que sean más seguros. considerando las fallas de seguridad que fueron encontradas y explotadas previamente, en el canal clásico. Vamos a comenzar con un análisis del aspecto más importante del diseño de los protocolos QKD basados en frames, correspondiente a la reutilización de pares. Es importante notar que cada directriz mencionada a continuación podría afectar considerablemente al diseño de los protocolos, ya que la reutilización de pares permite tanto aumentar el tamaño de la llave compartida como detectar y corregir errores en el canal cuántico.

4.1. Seguridad en bits

En [11], el *security strength*, de ahora en adelante bits de seguridad, es un valor asociado a la cantidad de trabajo requerido para romper un algoritmo criptográfico. Dicho trabajo puede ser medido a través del número de operaciones, el cual es expresado en bits.

Seguridad (en bits)	AES	RSA	ECDH
128	128	3072	256 - 383
192	192	7680	384 - 511
256	256	15360	512+

Tabla 4.1: Bits de seguridad para diferentes tamaños de llaves de AES (cifrado simétrico), RSA (cifrado asimétrico) y ECDH (intercambio de llaves).

Los bits de seguridad son calculados respecto al ataque más eficiente encontrado hasta el momento, sin considerar el uso de la computación cuántica. En general, se espera que un protocolo posea mínimo 128 bits de seguridad, por lo que utilizaremos esta métrica para evaluar a los protocolos QKD basados en frames.

4.2. Reutilización de pares

Tanto el *Pairs reuse attack* como el *Conjugate-Pairs reuse attack* son vulnerabilidades que se corrigen siempre y cuando se dejen de reutilizar pares. De esta manera, en LL20 y LL21, Alice debe calcular como máximo $\frac{|m|}{2}$ frames, en vez de las $\binom{|m|}{2}$ combinaciones. Así, la cantidad de frames usables se reduce considerablemente ya que, en LL20:

$$T'_{LL20} = \frac{1}{4} \cdot \frac{|m|}{2} \cdot \left(\frac{1}{2} - \frac{1}{3}e\right). \tag{4.1}$$

Sin la reutilización de pares, no es posible detectar ni corregir posibles errores del canal cuántico, a través del método de los frames auxiliares. Adicionalmente, una reutilización parcial de los pares, es decir, reutilizar sólo una parte de ellos, tampoco podría aumentar la seguridad de LL20, ya que los $S_j \in \{00, 11; 11, 11\}$ sólo pueden ser corregidos si existen frames auxiliares f_k con $S_k \in \{01, 10; 10, 10\}$ que comparten pares con los frames usables f_j. Así pues, dado que el *Pairs reuse attack* aún es factible para una reutilización parcial de los pares, es necesario utilizar métodos alternativos para la corrección de errores y definir una nueva expresión para T'_{LL20}.

Respecto a LLS21, tenemos una situación equivalente. Si se dejan de reutilizar pares, la cantidad de frames usables se reduce considerablemente:

$$T'_{LLS21} = \frac{3}{8} \cdot \frac{|m|}{3} \cdot \left(\frac{1}{3} - \frac{2}{7}e\right). \tag{4.2}$$

Al igual que en LL20, dado que los frames auxiliares deben reutilizar pares con los frames usables para detectar posibles errores del canal cuántico, no es factible utilizar este método de forma segura. La reutilización parcial de pares es un comportamiento que debe estudiarse en LLS21, ya que el artículo original no especifica cuáles deben ser los *Sifting Strings* (SS) de los frames auxiliares que reutilizan pares con los frames usables. Adicionalmente, el mismo vector de ataque del *Pairs reuse attack* en LL20 puede ser aplicado en LLS21:

$$\underset{S_j=111,110}{\begin{pmatrix} 1_X & - \\ - & 1_Z \\ - & 0_Z \end{pmatrix}, \begin{pmatrix} - & 1_Z \\ 1_X & - \\ 0_X & - \end{pmatrix}} - \underset{S_k=100,100}{\begin{pmatrix} 1_X & - \\ 0_X & - \\ 0_X & - \end{pmatrix}, \begin{pmatrix} 1_X & - \\ - & 0_Z \\ 0_X & - \end{pmatrix}}.$$

En este ejemplo, podemos notar que si $f_j^1 = f_k^1$, entonces podemos derivar el MR del frame f_j, pero no el del frame f_k, al igual que en LL20. Por otro lado, si $f_j^3 = f_k^3$, entonces la conclusión es equivalente, para el siguiente MR posible de f_j. Así pues, este aspecto debe ser considerado para analizar la factibilidad de una reutilización parcial de los pares que sea segura, definiendo una nueva expresión para T'_{LLS21}.

Por último, en LL21, al igual que sus predecesores, la cantidad de frames usables se ve reducida, considerando que Alice no reutiliza pares de estados cuánticos:

$$T'_{LL21} = \frac{1}{2} \cdot \frac{|m|}{2} = \frac{|m|}{4}. \tag{4.3}$$

A pesar de descartar la reutilización de pares, aún es factible utilizar el método corrector de errores de LL21. Según el artículo original, la corrección de errores se realiza mediante los frames auxiliares, caso equivalente respecto de LL20 y LLS21, pero también mediante los frames usables. La Tabla 4.2 muestra los CSS erróneos que permiten detectar los MRs de un frame usable f_j, sin la necesidad de reutilizar pares con un frame auxiliar. Sin embargo, utilizar únicamente los frames usables implica que es necesario definir una nueva expresión para T'_{LL21}, ya que los errores pueden ser corregidos siempre y cuando existan pares de estados cuánticos sin errores. Dicho de otra manera, la nueva expresión para T'_{LL21} dependerá del QBER e.

En conclusión, los protocolos LL20 y LLS21 utilizan métodos correctores de errores que deben ser modificados para que su seguridad pueda ser cuantificada, dada la amenaza del *Pairs reuse attack* en sus diseños, principalmente por el descarte de una gran cantidad de frames y la recuperación efectiva de MRs. LL21, en cambio, es un protocolo que puede tolerar el *Conjugate-Pairs reuse attack* siempre y cuando no se reutilicen pares de estados cuánticos, ya que parte del método corrector de errores no requiere del uso de frames auxiliares. Sin embargo, para que la modificación de LL21 sea segura, el tamaño de la llave compartida pasaría a ser lineal respecto a la cantidad de *double matchings* $|m|$, en el caso de no utilizar métodos diferentes para la derivación de la llave compartida.

CSS erróneo	Frame	MR
01,10	f_2, f_{14}	10
01,10	f_6	11
10,01	f_3, f_{13}	11
10,01	f_4, f_9, f_{10}	10

Tabla 4.2: Tabla de *Composed Sifting Strings* (CSS) erróneos y sus correspondientes *Measurement Results* (MRs). Parte del método corrector de errores en L21 permite que únicamente los frames usables f_j con los CSS y MRs indicados sean suficientes para la detección y corrección.

4.3. Derivación de la llave compartida

A pesar de que en L23 se reutilicen pares de estados cuánticos, el problema fundamental de la seguridad de este protocolo es el método de derivación de la llave compartida. Recordemos que, en él, Alice busca frames f_1 y f_5 en la lista L_1 y, a partir de los frames encontrados, sólo uno es suficiente para derivar la llave compartida en cada ronda. Lo anterior supone un cambio radical en el diseño de los protocolos QKD basados en frames, ya que en LL20, LLS21 y LL21, la llave compartida se deriva por frame, independientemente de si se reutilizan pares de estados cuánticos o no.

El *Avalanche-Effect attack* explota el hecho de que, dado que Alice toma un pivote p que corresponde a un par de estados cuánticos medido en la base X o Z, donde cada estado colapsó a un bit '1', la elección de p no es única, dado que cualquier frame de F_1 y F_5 es un candidato a ser un pivote. Dada la no unicidad de p, nosotros demostramos que para un frame $f_k \notin F_1, F_5$, su correspondiente pivote permite derivar la llave compartida, por lo que la seguridad de L23 se ve comprometida, independiente del largo de la lista L_1. Así pues, si Bob no reutiliza los pares de estados cuánticos, no es posible derivar la llave compartida y por ende es necesario modificar el diseño de L23.

El problema de seguridad de L23 se justifica también por la derivación de la llave privada a través de la pública. Recordemos que tanto RSA como ECDH fueron diseñados de tal manera de que su seguridad se basa en un problema matemático, justamente al relacionar la llave privada con la pública. Incluso, en los protocolos LL20, LLS21 y LL21, la llave privada se compone del conocimiento de los estados cuánticos que se envían (del lado de Alice) y las bases de medición que permiten obtener cada *double matching* (del lado de Bob). Así, se considera difícil computacionalmente derivar las bases de medición utilizando únicamente la llave pública de Alice (listas f y f') y Bob (lista S o C). Sin embargo, como pudimos demostrar, el uso de la llave pública en estos protocolos permite que, a través de la reutilización de pares, la llave privada se pueda recuperar desde la pública. En L23, las llaves privadas de Alice y Bob son equivalentes a las de los protocolos LL20, LLS21 y LL21, no así la llave pública, que corresponde a las listas L_1 y L_2, colapsando la seguridad de L23 a un problema matemático trivial.

4.4. El propósito de los frames

En resumen, se proponen dos directrices de seguridad que deben tomarse como referencia para desarrollar protocolos QKD basados en frames que sean más seguros:

1. Unicidad de los frames: Durante el intercambio de información clásica entre Alice y Bob, los pares de estados cuánticos no deben reutilizarse entre frames, es decir, cada frame debe ser único. La reutilización incontrolada de pares de estados cuánticos puede provocar que el protocolo QKD sea vulnerable al *Pairs reuse attack* y/o *Conjugate-Pairs reuse attack*.
2. Generación apropiada de llaves: Debe ser difícil computacionalmente obtener la llave compartida a través de la llave pública. Es importante que la seguridad del protocolo QKD sea superior a 128 bits.

La segunda directriz es genérica, así que vamos a explicarla a continuación. Desde BB84, las llaves privadas de Alice y Bob se han relacionado mediante el fenómeno cuántico, donde Alice conoce los estados cuánticos que envía, y Bob las bases de medición que utilizó y los bits obtenidos. Dependiendo del protocolo, las bases de medición (BB84 y BBM92), los bits obtenidos (LL20, LLS21 y L23) u otra información clásica (LL21) es expuesta, donde en el caso de los protocolos QKD basados en frames, la llave compartida está directamente relacionada con las bases de medición de Bob. Por ello, la recuperación de las bases de medición de Bob implica una recuperación de la llave compartida, la segunda directriz indica que debe ser difícil computacionalmente, obtener las bases de medición de Bob a través de la llave pública, pero se describe de forma genérica para considerar posibles variantes de los protocolos QKD basados en frames que se propongan en un futuro.

Desde LL20, el propósito del uso de los frames en QKD ha sido, en el canal cuántico, mitigar los ataques *Photon Number Splitting* (PNS) e *Intercept-Resend* (IR). En el canal clásico, el potencial de los frames recae en la factibilidad de aumentar, polinomialmente, el tamaño de la llave compartida, donde además se puedan detectar y corregir errores en el canal cuántico que permitan aumentar la eficiencia de los métodos respecto a otras propuestas. Es importante que las directrices de seguridad propuestas permitan continuar el desarrollo de protocolos QKD basados en frames sin que afecten a su propósito planteado desde un inicio, pero sí consideren la gran relevancia que tiene el criptoanálisis clásico para estos nuevos paradigmas que se encuentran actualmente revolucionando las comunicaciones cuánticas, tal y como lo hizo BB84 en su momento. Así, la unicidad de los frames y la generación apropiada de llaves representan la primera referencia para el desarrollo seguro de protocolos QKD basados en frames, siendo imprescindible su actualización de acuerdo a las nuevas propuestas venideras.

5 | Conclusiones

En el presente trabajo de Tesis se definieron las primeras directrices de seguridad para los protocolos de Distribución Cuántica de Llaves o *Quantum Key Distribution* (QKD) basados en la reconciliación por frames, matrices *n* x *m* que agrupan pares de estados cuánticos enviados y medidos por Alice (transmisor) y Bob (receptor), respectivamente. El método empleado para definir estas directrices se conoce como criptoanálisis clásico, donde se realiza un proceso de ingeniería inversa para comprender los fundamentos de los protocolos y así estudiar sus posibles vulnerabilidades. Se identificaron 3 ataques que permiten comprometer la seguridad de la totalidad de los protocolos QKD basados en frames publicados hasta el momento, llamados *Pairs reuse*, *Conjugate-Pairs reuse* y *Avalanche-Effect*. La factibilidad de cada ataque fue demostrada de manera teórica y experimental, utilizando Qiskit para simular la generación y medición de estados cuánticos. Así, la formulación de las directrices de seguridad son robustas.

En el Capítulo 1, se presentó el contexto y la relevancia de la criptografía como pilar fundamental de la seguridad de la información en el ámbito digital. La criptografía no solo garantiza la confidencialidad, integridad y autenticidad de la información, sino que también permite el intercambio seguro de llaves criptográficas, crucial para la comunicación segura entre dos partes, como Alice y Bob. El avance de la computación cuántica ha planteado nuevos desafíos a la criptografía clásica, subrayando la necesidad de alternativas como PQC y QKD. Sin embargo, los protocolos QKD basados en la reconciliación por frames supone un nuevo paradigma para las comunicaciones cuánticas, donde es fundamental que el criptoanálisis clásico se estudie con mayor profundidad, razón de la existencia de este trabajo de Tesis.

El Capítulo 2 abordó el análisis del estado del arte de la QKD, una tecnología capaz de intercambiar llaves criptográficas de manera segura utilizando fenómenos de la física cuántica. Se describieron los protocolos pioneros BB84 y BBM92, destacando su impacto en la evolución de las comunicaciones seguras. Además, se introdujo el concepto de frame y cómo estos han revolucionado la QKD, presentando cuatro propuestas de protocolos basados en ellos: LL20, LL21, LLS21 y L23. La inclusión de frames en los protocolos QKD tiene el potencial de mejorar la eficiencia y seguridad del intercambio de llaves, lo que constituye un avance significativo en el marco de la criptografía cuántica.

Luego, el Capítulo 3 se centró en el ejercicio de criptoanálisis de los protocolos QKD basados en frames, identificando tres vulnerabilidades críticas: el ataque de reutilización de pares (*Pairs reuse attack*), el ataque de reutilización de pares conjugados (*Conjugate-Pairs reuse attack*) y el ataque del efecto avalancha (*Avalanche-Effect attack*). Cada uno de estos ataques fue descrito teóricamente, y su impacto en los protocolos fue evaluado a través de simulaciones computacionales en Qiskit, demostrando la factibilidad de estos ataques. La capacidad de recuperar completamente la llave compartida a partir de estas vulnerabilidades subraya la necesidad urgente de fortalecer los protocolos QKD basados en frames contra tales amenazas, considerando que el canal clásico se encuentra autenticado.

El *Pairs reuse attack* explotó la reutilización de pares de estados cuánticos en LL20 y LLS21, donde demostramos que la llave compartida puede ser recuperada en su totalidad para una comunicación con y sin errores, respectivamente. En el *Conjugate-Pairs reuse* se demostró la factibilidad de recuperar la llave compartida de LL21 mediante los principios del *Pairs reuse attack* y el determinismo de ciertos *Composed Sifting Strings* (CSS). Finalmente, el *Avalanche-Effect attack* explotó una falla de diseño de L23, donde la derivación de la llave compartida puede realizarse de manera trivial para un atacante, ya que un único frame es suficiente para determinarla. Los resultados de las simulaciones asociadas a cada protocolo proporcionaron una validación empírica de los riesgos teóricos identificados.

Tomando las vulnerabilidades encontradas, en el Capítulo 4 se desarrollaron las primeras directrices de seguridad

específicas con el objetivo de construir protocolos QKD basados en frames que sean más seguros. Estas directrices abarcan diversos aspectos del diseño de protocolos, incluyendo la derivación de la llave compartida y la reutilización de pares de estados cuánticos para el cómputo de frames. Así pues, se concluye que los protocolos LL20, LLS21 y L23 deben modificar su diseño para que su seguridad pueda ser cuantificada, caso contrario de LL21, el cual puede seguir siendo utilizado quitando la reutilización de pares de estados, donde es necesario estudiar sus implicancias respecto al tamaño de la llave compartida. La implementación de estas directrices es crucial para mitigar los riesgos asociados con las vulnerabilidades detectadas y garantizar la autenticidad y confidencialidad de las comunicaciones cuánticas. Sin embargo, la constante actualización de estas directrices es imprescindible para evaluar la seguridad de nuevas propuestas futuras.

El criptoanálisis realizado en este trabajo no solo ha permitido identificar debilidades específicas en los protocolos QKD basados en frames, sino que también ha sentado las bases para el desarrollo de estrategias de mitigación efectivas. Al proponer directrices de seguridad claras y aplicables, este trabajo contribuye de manera significativa a la creación de protocolos más seguros, que puedan resistir tanto las amenazas actuales como las futuras en el ámbito de la QKD. Este enfoque proactivo es esencial para mantener la confianza en los sistemas de comunicación cuántica y asegurar que puedan ofrecer una protección robusta contra los ataques clásicos.

Las contribuciones de este trabajo de Tesis son de gran relevancia para la comunidad académica y la industria. El criptoanálisis detallado de los protocolos QKD basados en frames y las directrices de seguridad propuestas ofrecen un marco valioso para futuros desarrollos en este campo. Además, se sugiere que investigaciones futuras exploren la implementación práctica de estas directrices y evalúen su efectividad en entornos reales. Asimismo, el estudio de nuevas vulnerabilidades y la constante actualización de las directrices de seguridad serán cruciales considerando los avances tecnológicos y las amenazas emergentes.

Bibliografía

[1] Christof Paar y Jan Pelzl. "Understanding Cryptography: A Textbook for Students and Practitioners". En: 1st. Berlin, Heidelberg: Springer Publishing Company, Incorporated, 2009, págs. 1-4.

[2] Morris J. Dworkin. *Advanced Encryption Standard (AES)*. en. 2023-05-09 04:05:00 de 2023. DOI: https://doi.org/10.6028/NIST.FIPS.197-upd1.

[3] Kathleen Moriarty et al. *PKCS #1: RSA Cryptography Specifications Version 2.2*. RFC 8017. Nov. de 2016. DOI: 10.17487/RFC8017.

[4] Quynh Dang. *Secure Hash Standard*. en. 2015-08-04 de 2015. DOI: https://doi.org/10.6028/NIST.FIPS.180-4.

[5] Dr. Hugo Krawczyk, Mihir Bellare y Ran Canetti. *HMAC: Keyed-Hashing for Message Authentication*. RFC 2104. Feb. de 1997. DOI: 10.17487/RFC2104.

[6] Michael B. Jones, John Bradley y Nat Sakimura. *JSON Web Token (JWT)*. RFC 7519. Mayo de 2015. DOI: 10.17487/RFC7519.

[7] Priyadarshini Patil et al. "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish". En: *Procedia Computer Science* 78 (2016). 1st International Conference on Information Security Privacy 2015, págs. 617-624. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2016.02.108.

[8] Eric Rescorla. *Diffie-Hellman Key Agreement Method*. RFC 2631. Jun. de 1999. DOI: 10.17487/RFC2631.

[9] Kevin Igoe, David McGrew y Margaret Salter. *Fundamental Elliptic Curve Cryptography Algorithms*. RFC 6090. Feb. de 2011. DOI: 10.17487/RFC6090.

[10] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Ago. de 2018. DOI: 10.17487/RFC8446.

[11] Elaine Barker. *Recommendation for key management:* mayo de 2020. DOI: 10.6028/nist.sp.800-57pt1r5.

[12] Paul Benioff. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". En: *Journal of Statistical Physics* 22.5 (mayo de 1980), págs. 563-591. ISSN: 1572-9613. DOI: 10.1007/bf01011339.

[13] Isaac L. Chuang, Neil Gershenfeld y Mark Kubinec. "Experimental Implementation of Fast Quantum Searching". En: *Phys. Rev. Lett.* 80 (15 abr. de 1998), págs. 3408-3411. DOI: 10.1103/PhysRevLett.80.3408.

[14] Michael A. Nielsen e Isaac L. Chuang. "Quantum Computation and Quantum Information: 10th Anniversary Edition". En: Cambridge, England: Cambridge University Press, 2010, págs. 13-17.

[15] Michael A. Nielsen e Isaac L. Chuang. "Quantum Computation and Quantum Information: 10th Anniversary Edition". En: Cambridge University Press, 2010. Cap. 1, pág. 14.

[16] Yudong Cao et al. "Quantum Chemistry in the Age of Quantum Computing". En: *Chemical Reviews* 119.19 (ago. de 2019), págs. 10856-10915. ISSN: 1520-6890. DOI: 10.1021/acs.chemrev.8b00803.

[17] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". En: *SIAM Journal on Computing* 26.5 (1997), págs. 1484-1509. DOI: 10.1137/S0097539795293172.

[18] ID Quantique. *White paper: Understanding Quantum Cryptography*. Inf. téc. Ch. de la Marbrerie 3 1227, Carouge, Switzerland: ID Quantique SA, mayo de 2020. URL: https://marketing.idquantique.com/acton/attachment/11868/f-020d/1/-/-/-/-/Understanding%20Quantum%20Cryptography_White%20Paper.pdf.

[19] Charles H. Bennett y Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing". En: *Theoretical Computer Science* 560 (2014). Theoretical Aspects of Quantum Cryptography – celebrating 30 years of BB84, págs. 7-11. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2014.05.025.

[20] Mujtaba Zahidy et al. "Quantum key distribution using deterministic single-photon sources over a field-installed fibre link". En: *npj Quantum Information* 10.1 (ene. de 2024). ISSN: 2056-6387. DOI: 10.1038/s41534-023-00800-x.

[21] Victor Lovic. "Quantum Key Distribution: Advantages, Challenges and Policy". En: (2020). DOI: 10.17863/CAM.58622.

[22] Miralem Mehic et al. "Error Reconciliation in Quantum Key Distribution Protocols". En: *Reversible Computation: Extending Horizons of Computing: Selected Results of the COST Action IC1405*. Ed. por Irek Ulidowski et al. Cham: Springer International Publishing, 2020, págs. 222-236. ISBN: 978-3-030-47361-7. DOI: 10.1007/978-3-030-47361-7_11.

[23] Limei Guo, Hsiao-Chun Wu y Duan Huang. "Novel intelligent blind information reconciliation for LDPC codes in quantum key distribution systems". En: *Physical Communication* 64 (2024), pág. 102348. ISSN: 1874-4907. DOI: https://doi.org/10.1016/j.phycom.2024.102348.

[24] Luis A. Lizama-Pérez et al. "Quantum Flows for Secret Key Distribution in the Presence of the Photon Number Splitting Attack". En: *Entropy* 16.6 (2014), págs. 3121-3135. ISSN: 1099-4300. DOI: 10.3390/e16063121.

[25] Luis Adrian Lizama-Pérez, José Mauricio López y Eduardo De Carlos López. "Quantum Key Distribution in the Presence of the Intercept-Resend with Faked States Attack". En: *Entropy* 19.1 (2017). ISSN: 1099-4300. DOI: 10.3390/e19010004.

[26] Luis Adrián Lizama-Pérez, J. Mauricio López R. y Emmanuel H. Samperio. "Beyond the Limits of Shannon's Information in Quantum Key Distribution". En: *Entropy* 23.2 (2021). ISSN: 1099-4300. DOI: 10.3390/e23020229.

[27] Luis Adrián Lizama-Perez. "Reverse Reconciliation for Optimal Error Correction in Quantum Key Distribution". En: *Symmetry* 15.3 (2023). ISSN: 2073-8994. DOI: 10.3390/sym15030710.

[28] Gilles van Assche. "Privacy amplification using hash functions". En: *Quantum Cryptography and Secret-Key Distillation*. Cambridge, England: Cambridge University Press, 2006, págs. 101-112.

[29] Hong-Yi Su. "Simple analysis of security of the BB84 quantum key distribution protocol". En: *Quantum Information Processing* 19.6 (abr. de 2020). ISSN: 1573-1332. DOI: 10.1007/s11128-020-02663-z.

[30] Abdulbast A. Abushgra. "Variations of QKD Protocols Based on Conventional System Measurements: A Literature Review". En: *Cryptography* 6.1 (2022). ISSN: 2410-387X. DOI: 10.3390/cryptography6010012.

[31] Charles H. Bennett, Gilles Brassard y N. David Mermin. "Quantum cryptography without Bell's theorem". En: *Phys. Rev. Lett.* 68 (5 feb. de 1992), págs. 557-559. DOI: 10.1103/PhysRevLett.68.557.

[32] Aitor Brazaola-Vicario et al. "Quantum key distribution: a survey on current vulnerability trends and potential implementation risks". En: *Opt. Continuum* 3.8 (ago. de 2024), págs. 1438-1460. DOI: 10.1364/OPTCON.530352.

[33] Luis A. Lizama-Perez y J. Mauricio López. "Quantum Key Distillation Using Binary Frames". En: *Symmetry* 12.6 (2020). ISSN: 2073-8994. DOI: 10.3390/sym12061053.

[34] Luis Adrián Lizama-Pérez y José Mauricio López-Romero. "Perfect Reconciliation in Quantum Key Distribution with Order-Two Frames". En: *Symmetry* 13.9 (2021). ISSN: 2073-8994. DOI: 10.3390/sym13091672.

[35] Daniel Espinoza Figueroa. *FramesQKD, Key Recovery attacks on Frame-based QKD protocols*. 2024. URL: https://github.com/D-Cryp7/FramesQKD.

[36] Ali Javadi-Abhari et al. *Quantum computing with Qiskit*. 2024. DOI: 10.48550/arXiv.2405.08810.