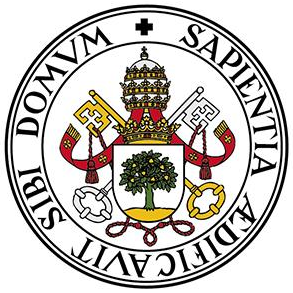


UNIVERSIDAD DE VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE TELECOMUNICACIÓN

Fundamentos de la distribución cuántica de claves (QKD) y experimentación con el simulador QKDNetSim

Autor:
Dña. Carmen Canet Pérez

Tutores:
D. Ignacio de Miguel Jiménez
D. Ramón J. Durán Barroso

Valladolid, septiembre de 2025

TÍTULO: **Fundamentos de la distribución cuántica de claves (QKD) y experimentación con el simulador QKDNetSim**

AUTOR: **Dña. Carmen Canet Pérez**

TUTORES: **D. Ignacio de Miguel Jiménez**
D. Ramón J. Durán Barroso

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

TRIBUNAL

PRESIDENTE: **D. Ignacio de Miguel Jiménez**

VOCAL: **D. Ramón J. Durán Barroso**

SECRETARIO: **Dña. Noemí Merayo Álvarez**

SUPLENTE: **D. Juan Carlos Aguado Manzano**

SUPLENTE: **D. Patricia Fernández del Reguero**

FECHA: **Septiembre 2025**

CALIFICACIÓN:

Resumen del TFG

La seguridad de la información afronta un cambio de paradigma debido al desarrollo de la computación cuántica, que amenaza con comprometer los algoritmos criptográficos actuales. La distribución cuántica de claves (QKD) surge como tecnología estratégica, capaz de garantizar a largo plazo seguridad teórica de la información (ITS) gracias a las características que aportan los principios físicos de la cuántica como la superposición, el entrelazamiento y la no clonación. El sistema proporciona a dos entidades la capacidad de establecer claves secretas simétricas mediante el intercambio de estados cuánticos en un canal dedicado, por fibra o espacio libre, donde cualquier intento de interceptación será detectable. Las claves generadas se utilizarán posteriormente por una aplicación en procesos de cifrado clásico junto con OTP o AES.

Este trabajo presenta, en primer lugar, los principales fundamentos cuánticos y criptográficos que se relacionan con QKD, así como los principales métodos, protocolos, componentes, interfaces, procesos y capas que forman parte de una red QKD. Posteriormente, a nivel experimental, se presenta la herramienta de simulación QKDNetSim, que permite analizar redes de distribución de claves y comprender el comportamiento de QKD en condiciones controladas bajo parámetros configurables. Mediante distintos escenarios de prueba, se evalúa cómo influyen variables como el tamaño de clave, la tasa de generación o la configuración de los nodos en la disponibilidad de clave y el rendimiento del sistema.

Palabras clave

Qubit, superposición, bases, seguridad ITS, OTP, criptografía cuántica, módulo QKD, BB84, generación de clave, SAE, KMS, interfaz ETSI 014, interfaz ETSI 004, nodo de confianza, QKDN, simulador QKDNetSim.

Abstract

Information security is facing a paradigm shift due to the development of quantum computing, which threatens to compromise current cryptographic algorithms. Quantum key distribution (QKD) is emerging as a strategic technology capable of guaranteeing long-term theoretical information security (ITS) thanks to the characteristics provided by the physical principles of quantum mechanics, such as superposition, entanglement, and non-cloning theorem. The system provides two entities with the ability to establish symmetric secret keys by exchanging quantum states over a dedicated channel, via fiber or free space, where any attempt at interception will be detectable. The generated keys will then be used by an application in classical encryption processes together with OTP or AES.

This work first presents the main quantum and cryptographic fundamentals related to QKD, as well as the main methods, protocols, components, interfaces, processes, and layers that form part of a QKD network. Subsequently, at the experimental level, the QKDNetSim simulation tool is presented, which allows the analysis of quantum key distribution networks and provides insights into QKD behavior under controlled conditions with configurable parameters. Through different test scenarios, it evaluates how variables such as key size, generation rate, and node configuration influence key availability and system performance.

Keywords

Qubit, superposition, bases, ITS security, OTP, quantum cryptography, QKD module, BB84, key generation, SAE, KMS, ETSI 014 interface, ETSI 004 interface, trusted node, QKDN, QKDNetSim simulator.

Agradecimientos

En primer lugar, quiero agradecer a Nacho por su paciencia, disposición y buena voluntad para acompañar el desarrollo de este TFG especialmente en los momentos más decisivos.

Más allá del ámbito académico, quiero expresar mi profunda gratitud a toda mi familia y amistades, que han sido un pilar fundamental y que tanto han cuidado de mí durante todos estos años.

También, en un plano más personal, a esa niña que soñaba con entender el mundo a su alrededor, gracias por no soltar la idea, mantener la ilusión y seguir creyendo en ti misma.

Este trabajo está asociado al proyecto de investigación ONOFRE-4, el cual ha sido financiado por el Ministerio de Ciencia, Innovación y Universidades, la Agencia Estatal de Investigación y por FEDER/UE (proyecto PID2023-148104OB-C41 financiado por MICIU/AEI/10.13039/501100011033 y por FEDER/UE).

Declaración de uso de IA generativa

Durante la preparación de este trabajo se utilizó ChatGPT y DeepL para mejorar la gramática y la legibilidad de algunas frases. Después de utilizar estas herramientas/servicios, la autora revisó y editó el contenido en la medida necesaria y asume toda la responsabilidad por el contenido de la publicación.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

1. Introducción

1.1. Contexto

La seguridad de la información constituye un pilar fundamental en el crecimiento actual de la sociedad digital, donde el desarrollo de la computación cuántica plantea desafíos significativos a los mecanismos criptográficos convencionales que deben proteger dicha información.

En este contexto, la criptografía cuántica, en particular la distribución cuántica de claves QKD (*Quantum Key Distribution*), se presenta a modo de solución que aprovecha los principios físicos de la mecánica cuántica, como la superposición y el entrelazamiento, con el fin de ser una alternativa viable que proteja las comunicaciones y garantice la confidencialidad de las comunicaciones frente a posibles ataques. Por ello, el principal objetivo de QKD es proporcionar comunicaciones con seguridad teórica de la información o, por sus siglas en inglés, ITS (*Information Theoretical Secure*) [1], lo que significa que su seguridad no depende de limitaciones computacionales, sino de las leyes de la física.

Tanto en el actual mundo industrial como en el académico, el desarrollo de la investigación en tecnología basada en cuántica ha llevado a conseguir dispositivos y sistemas cuánticos de gran interés que esperan el momento para ser comercializados en gran escala. Sin embargo, la complejidad técnica, el coste de los equipos y la falta de infraestructuras consolidadas hacen que, por ahora, su despliegue a gran escala sea limitado. Debido a ello, el desarrollo de simuladores en este campo, adquieren un papel fundamental al permitir modelar, analizar y optimizar distintos escenarios, sin tener que trabajar con sus componentes físicos.

1.2. Objetivos planteados

Este trabajo tiene como primer objetivo realizar un estudio conceptual de la distribución cuántica de clave, QKD, abarcando desde sus fundamentos en la cuántica y la criptografía, hasta la estructura y particularidades de su implementación en redes.

Una vez planteada la complejidad del sistema, el segundo objetivo de este trabajo consiste en abstraer el comportamiento teórico visto. Para ello, se realizará un estudio de la plataforma que facilita el simulador de red denominado QKDNetSim, presentando las posibilidades que ofrece y exponiendo el planteamiento de varias configuraciones de prueba elementales para analizar el comportamiento del entorno

frente a diversas configuraciones externas de su interfaz y simular así sistemas basados en la distribución cuántica de clave, QKD.

1.3. Estructura de la memoria

La estructura planteada para desarrollar los objetivos planteados en este trabajo consiste en seis capítulos.

Capítulo 1. Se presenta una introducción general de la temática, la presentación de los objetivos generales planteados y el resumen de la estructura del documento.

Capítulo 2. Expone los fundamentos teóricos de la mecánica cuántica como los conceptos de qubit, estado, superposición, medida, linealidad, entrelazamiento y no clonación. También se describe su relación con la seguridad y la confidencialidad de la información bajo técnicas criptográficas clásicas y cuánticas.

Capítulo 3. Este capítulo aborda los fundamentos de un sistema QKD, sus formas de implementación, los medios de transmisión, el protocolo BB84, los componentes fundamentales y las particularidades de la arquitectura de red.

Capítulo 4. Se presenta la herramienta QKDNetSim, especificando su origen, características, arquitectura, parámetros que se modelan, además de la configuración de un caso de uso como base para presentar la salida obtenida del simulador.

Capítulo 5. En este apartado, siguiendo la línea introducida por el anterior capítulo, se presentan varios escenarios de simulación, analizando los resultados obtenidos al variar el número de elementos del sistema bajo la variación de distintos parámetros para analizar su comportamiento e influencia.

Capítulo 6. Recoge las conclusiones del trabajo y se plantean posibles líneas de investigación futura.

Capítulo 2

2. Fundamentos teóricos

La mecánica cuántica es la rama de la física que describe el comportamiento de la materia y la energía a escalas microscópicas, donde las leyes clásicas dejan de ser aplicables. A diferencia de la física clásica, donde el estado de un sistema es determinista, en el ámbito cuántico los sistemas se representan mediante estados cuánticos, que pueden estar en superposición de varias configuraciones posibles. Esto constituye la base de los primeros protocolos de distribución cuántica de claves, como el BB84. Otro fenómeno característico es el entrelazamiento, en el que dos partículas comparten un estado conjunto con correlaciones que no pueden explicarse por la física clásica, recurso que más adelante sirvió para ser aplicado en protocolos alternativos de QKD.

El desarrollo de la cuántica aplicada a la computación y las redes ha marcado un punto de inflexión en el panorama de la seguridad de la información, amenazando con superar el planteamiento actual de los sistemas criptográficos tradicionales.

La criptografía clásica, basada en algoritmos de clave simétrica y asimétrica, se ve amenazada por los algoritmos cuánticos como el de Shor y el de Grover, que tienen el potencial de romper estos sistemas de forma eficiente y en un tiempo inviable para una computadora clásica. Ante esta situación, la irrupción de la criptografía cuántica ayuda a evitar este problema de la mano de las leyes y principios de la física de la mecánica cuántica.

En este caítulo se van a desarrollar en una primera parte, los conceptos y procesos esenciales de las propiedades que ofrece la mecánica cuántica, con un enfoque técnico principalmente basado en los libros de Wong [2] y VanMeter [3]. Otras dos referencias clave son, el artículo introductorio de Golec et al. [4], que ofrece una visión teórica más compacta, y el de Hoofnagle y Garfinkel [5] que ofrece otra perspectiva más extensa y profunda de la temática. Posteriormente, en segundo lugar, se introducen los desafíos que presenta la criptografía clásica frente a la cuántica, identificando los algoritmos de procesos de cifrado y de autentificación, para finalizar introduciendo el desafío de la distribución de claves y la solución que presenta QKD.

2.1. Fundamentos de la mecánica cuántica

2.1.1. Qubit y estado

En la computación clásica la unidad básica de información es el bit adoptando como estado los valores binarios 0 o 1. En cambio, el qubit o "quantum bit" en la computación cuántica, transforma radicalmente el concepto clásico del estado de la información. Para describirlo, se utiliza la notación de Dirac, al ser Paul Dirac uno de los fundadores de la mecánica cuántica y quien introdujo la notación "*bra-ket"* [2].

Un *ket* genérico, cuya notación es representada mediante una barra vertical y un corchete angular, se denota como $|\psi\rangle$ y representa un estado cuántico genérico como un vector columna, de forma tal que:

$$|\psi\rangle = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{bmatrix}$$

Si el estado cuántico está definido por un solo qubit, la matriz solo tendrá dos elementos. Al aumentar el número de qubits, la representación requiere cada vez más coeficientes, como se verá más adelante.

Un estado cuántico también se puede definir como otro elemento denominado, *bra*, que se representa mediante un corchete angular seguido de una barra vertical, con un propósito principalmente matemático y operacional a la hora de tratar los qubits, ya que un *bra* es el vector conjugado transpuesto de un *ket*, quedando como se describe a continuación.

$$\langle\psi| = [a_0^* \quad a_1^* \quad \dots \quad a_{N-1}^*]$$

Tanto en el interior de un *ket* como en el de un *bra*, también se representan los estados base del sistema, $|0\rangle$ y $|1\rangle$, de tal manera que la representación matricial convencional de estos es la siguiente:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \; ; \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Pero un qubit no está limitado a representar simplemente el estado de 0 o de 1, sino que, gracias a adaptar los fundamentos de la física cuántica, ahora el estado del qubit se describe como una superposición lineal de ambos valores al mismo tiempo, bajo unos coeficientes determinados como α y β, de tal manera que el estado de un qubit genérico quedaría definido como una superposición de ambos.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Y en forma matricial:

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

2.1.2. Superposición: Interpretación visual en Esfera de Bloch

Para facilitar la interpretación visual de los qubit, se presenta una herramienta fundamental para entender los estados, la esfera de Bloch, que sirve como una representación geométrica tridimensional del espacio en donde se sitúan los estados de un qubit.

Los estados base $|0\rangle$ y $|1\rangle$, se encuentran ubicados en el eje Z, de forma que los extremos del eje que atraviesa la esfera de Bloch por los polos de norte a sur, dejando situados al $|0\rangle$ en el extremo superior del hemisferio norte y al $|1\rangle$ en el extremo inferior del hemisferio sur. Un qubit puede ser representado como cualquier punto en la superficie de la esfera de Bloch, gracias al comportamiento intrínseco de la superposición que poseen los estados cuánticos.

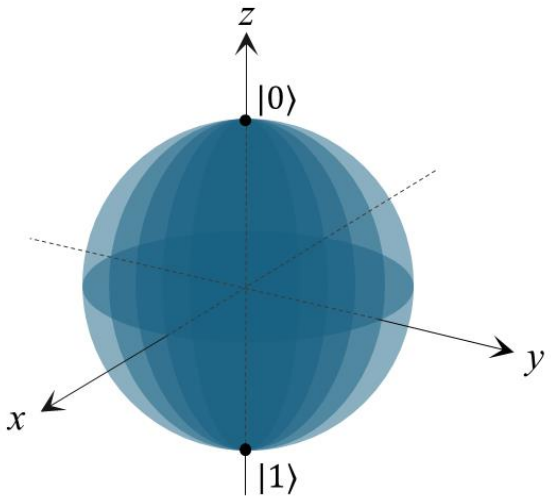
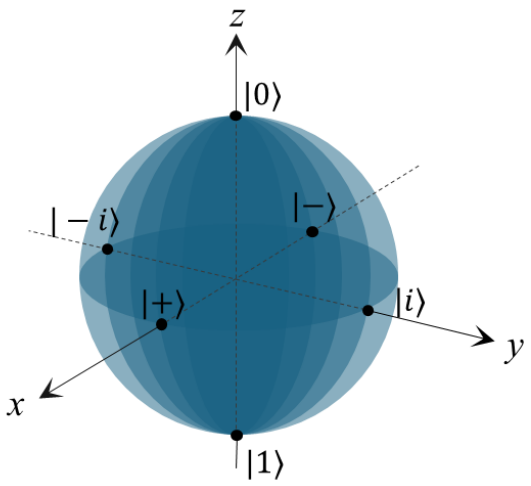


Figura 1. Esfera de Bloch con ejes y estados base

La superposición es un concepto que permite, gracias a los fundamentos de las leyes de la mecánica cuántica, que el estado de un qubit sea una combinación lineal de $|0\rangle$ y $|1\rangle$, en lo que se denomina un estado de superposición. De tal manera que, cuando un estado es representado sobre el ecuador de la esfera de Bloch se encuentra en una superposición equilibrada de los estados base $|0\rangle$ y $|1\rangle$, es decir, desde el punto de vista geométrico cualquier punto en la línea del ecuador está a la misma distancia de los polos del eje Z, reflejando una composición simétrica de ambos componentes.

Se definen específicamente los puntos coincidentes entre el ecuador y los dos ejes X e Y, ya que, al ser utilizados con gran frecuencia, están definidos bajo su propio nombre como $|+\rangle$, $|-\rangle$, $|i\rangle$ y $|-i\rangle$.



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

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

$$|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$$

$$|-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$$

Figura 2. Esfera de Bloch con estados de superposición básicos

Para determinar la posición de un qubit en la esfera de Bloch considerando un estado general $|\psi\rangle$ con amplitudes α y β, se parametrizan las amplitudes en función de dos ángulos θ y ϕ. El angulo θ, se denomina ángulo polar y mide la inclinación desde el polo norte hacia abajo, mientras que φ, denominado ángulo acimutal, mide la rotación transversal en el plano XY desde el eje X.

La representación gráfica del estado en la esfera, teniendo en cuenta que $0 \leq \theta \leq \pi$ y $0 \leq \phi < 2\pi$, queda descrita por la siguiente ecuación [2]:

$$|\psi\rangle = cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

A modo de ejemplo para ilustrar geométricamente la representación de un qubit en la esfera de Bloch, si se considera un estado cuántico $|\psi\rangle$ con los ángulos $\theta = \frac{\pi}{2}$ y $\phi = \frac{\pi}{4}$, quedaría definido el siguiente estado de superposición:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1+i}{2}|1\rangle$$

Como se puede observar a continuación en la Figura 3, en este ejemplo el qubit se ubica sobre la esfera en el plano XY del ecuador de Z, debido al ángulo θ, y desplazado según ϕ del eje X.

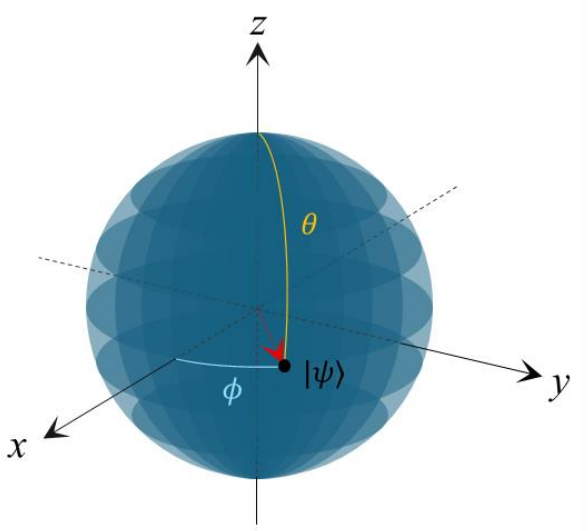


Figura 3. Ejemplo de representación de un estado cuántico

Los qubits, además de en la circunferencia del ecuador, también pueden ser representados en cualquier otro lugar, pudiendo ubicarse en toda la superficie de la esfera. Estos qubits quedarán situados de forma más cercana al $|0\rangle$ en el hemisferio norte, o al $|1\rangle$ o en el hemisferio sur de la esfera, dependiendo del valor de sus componentes. Esta ubicación es significativa, ya que con ella se va a poder interpretar el comportamiento final del estado del qubit al ser medido.

2.1.3. Medición y colapso

La mecánica cuántica establece que un qubit puede encontrarse en una superposición de los estados base $|0\rangle$ y $|1\rangle$. Sin embargo, también postula que al medir un qubit en esa base, para leer su resultado, se obtiene un único valor definido, $|0\rangle$ o $|1\rangle$, con probabilidades determinadas por las amplitudes de la superposición. Concretamente esta probabilidad viene dada por la norma al cuadrado de los coeficientes α y β, de tal manera que la probabilidad de que al medir el estado del qubit, este sea $|1\rangle$, viene dada por la norma al cuadrado del coeficiente α y la probabilidad de que al medir el estado del qubit, este sea $|0\rangle$ la describe la norma al cuadrado de β.

Además, se debe de respetar que un estado cuántico ha de estar normalizado, por lo que la probabilidad total de medir $|0\rangle$ o $|1\rangle$, tiene que ser igual a 1.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$p_{|0\rangle} = |\alpha|^2 \;\; ; \;\; p_{|1\rangle} = |\beta|^2$$

$$|\alpha|^2 + |\beta|^2 = 1$$

El proceso de medir un qubit cobra importancia, ya que provoca que el qubit colapse a uno de sus estados base. El colapso es la consecuencia del proceso de medida, ocasionando que el qubit deje de estar superpuesto y adopte un estado definido. Los

estados situados en el ecuador como, por ejemplo, el estado $|+\rangle$, al ser medidos van a tener un 50% de probabilidades de colapsar al estado $|0\rangle$ y un 50% de hacerlo al $|1\rangle$.

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

$$p_{|0\rangle} = p_{|1\rangle} = \left(\frac{1}{\sqrt{2}}\right)^2 = 0.5$$

Es por esto por lo que, gracias a la representación en la esfera de Bloch, cuando el qubit no está en el ecuador, ayuda a interpretar visualmente hacia qué estado es más probable que colapse el qubit tras realizar una medida representando la distinta probabilidad de colapso a $|0\rangle$ o $|1\rangle$ según se encuentre en el hemisferio norte o en el hemisferio sur del eje Z.

Una vez un el estado de un qubit ha sido determinado, si inmediatamente se quiere volver a medir su estado en la misma base, este ya ha perdido su comportamiento probabilístico, es decir, si en un primer lugar la medida resultó colapsar en el estado $|0\rangle$, con la segunda medida el resultado será de nuevo $|0\rangle$ con una probabilidad del 100%.

Por otra parte, cabe destacar que en la esfera de Bloch se pueden tomar dos puntos opuestos cualesquiera como polo norte y polo sur, esto es válido para cualquier conjunto de valores opuesto formando lo que se denomina una base. Por esta razón, tomando los puntos opuestos conocidos que delimitan los ejes de coordenadas, al conjunto formado por $\{|0\rangle, |1\rangle\}$ se le denomina base Z, en cambio si se toman $\{|+\rangle, |-\rangle\}$, se denomina base X, y si es trata del conjunto $\{|i\rangle, |-i\rangle\}$, se denomina base Y. Pudiendo medir con respecto a cualquiera de estas bases, o con respecto a dos estados cualesquiera en lados opuestos de la esfera de Bloch.

2.1.4. Algebra lineal aplicable a los estados cuánticos

Para operar con los estados de qubits se utiliza el lenguaje matemático del algebra lineal, lo cual es idóneo para realizar los cálculos con sistemas de algebra computacional. Dos de los operadores básicos necesarios son el producto interno y el producto exterior.

El producto interno, o producto escalar, de un qubit consigo mismo se puede simplificar operándose en su forma vectorial obteniendo la siguiente expresión.

$$\langle\psi|\psi\rangle = (\alpha^*\beta^*)\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = |\alpha|^2 + |\beta|^2 = 1$$

Este caso sirve para calcular la probabilidad total del estado de superposición, si este está normalizado debe ser igual a 1. Pero en caso de ser aplicado a dos qubits opuestos, como por ejemplo $\langle +|-\rangle$, tendrá un producto escalar igual a 0, denominándose estados ortogonales, u ortonormales si se cumplen ambas opciones.

Esa operación es de utilidad para realizar cambios de base y como otra manera de obtener las amplitudes de los estados cuánticos principalmente al medirse sobre otras bases. Como, por ejemplo, podemos escribir el estado de un qubit $|\psi\rangle$ definido en $\{|0\rangle, |1\rangle\}$ midiéndolo en la base ortonormal $\{|i\rangle, |-i\rangle\}$, calculado el valor de la amplitud final para cada estado de esta base con extremos en $|i\rangle$ y $|-i\rangle$, mediante el producto interno de $\langle i|\psi\rangle$ y de $\langle -i|\psi\rangle$ respectivamente, como se puede ver en la siguiente expresión.

$$|\psi\rangle = \langle i|\psi\rangle|i\rangle + \langle -i|\psi\rangle|-i\rangle$$

El producto exterior es otra operación importante que da lugar a matrices, su nomenclatura se dispone al contrario que en el producto interno. Para dos qubits genéricos como ψ, con amplitudes α y β, y otro como ϕ con amplitudes γ y δ, el producto exterior se define de la siguiente manera.

$$|\psi\rangle\langle\phi| = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}(\gamma^* \delta^*) = \begin{pmatrix}\alpha\gamma^* & \alpha\delta^*\\ \beta\gamma^* & \beta\delta^*\end{pmatrix}$$

2.1.5. Puertas cuánticas, linealidad y reversibilidad

La representación matricial es muy útil ya que en computación cuántica se implementan las puertas cuánticas, que actúan como operadores fundamentales que permiten manipular el estado de los qubits de forma similar a como actúan las puertas lógicas clásicas sobre los bits. No obstante, al tratarse los sistemas cuánticos de procesos de distinta naturaleza, las puertas cuánticas presentan una estructura matemática que permite adaptarse a estas necesidades empleando matrices para realizar transformaciones precisas sobre el estado del qubit.

En particular, se requiere que estas transformaciones sean lineales y unitarias. Como operadores unitarios, su forma genérica es representadas mediante la letra U de "unitaria" [2]. Las puertas cuánticas han de satisfacer que:

$$U^\dagger U = I$$

Una matriz que satisface esta igualdad es una matriz unitaria válida, siendo I la matriz identidad, y $U^\dagger$, la conjugada transpuesta de U. Si tenemos un qubit y aplicamos una puerta cuántica U, podemos deshacer la puerta aplicando $U^\dagger$. Por lo que, una puerta cuántica U es siempre reversible, y su inversa es $U^\dagger$.

$$U^{-1} = U^\dagger$$

La linealidad es una propiedad esencial para preservar la estructura de superposición de los estados cuánticos. Que su proceso deba ser lineal, indica que la acción sobre $|\psi\rangle$ se distribuye de tal manera que se preserva su estructura de superposición

manteniendo la probabilidad total del estado normalizado a 1, si previamente el estado lo estaba.

$$U|\psi\rangle = |U\psi\rangle$$

$$U(\alpha|0\rangle + \beta|1\rangle) = \alpha U|0\rangle + \beta U|1\rangle$$

2.1.5.1. Puertas cuánticas fundamentales

Las puertas cuánticas fundamentales realizan transformaciones precisas sobre el estado de un único qubit de manera que se consigue convertir los estados mediante un proceso reversible. Estas puertas se interpretan geométricamente como rotaciones en la esfera de Bloch. A continuación, se describen las principales puertas de un qubit y sus propiedades relevantes, resumidas posteriormente en la Tabla 1 [2]:

- **Puerta de identidad *I*:** Actúa como el operador neutro del conjunto de puertas cuánticas dejando inalterado el estado del qubit, es decir, no produce ninguna transformación, pero es de utilidad, ya que permite expresar formalmente operaciones que afectan solo a una parte del sistema, dejando el resto intacto.
- **Puerta de Pauli *X*:** Conocida como NOT cuántico, produce un intercambio de los estados base. Invierte la posición del estado $|0\rangle$ con el $|1\rangle$, dejando sin variar los estados $|+\rangle$ y $|-\rangle$. Geométricamente corresponde con una rotación de 180° alrededor del eje X en la esfera de Bloch. De tal manera que aplicando la puerta dos veces, se volvería al estado inicial, al estar aplicando un giro de 360º, de tal manera que $X^2 = I$.
- **Puerta de Pauli *Y*:** Transforma los estados base incluyendo factores de fase complejos, dejando esta vez sin variación los estados $|i\rangle$ y $|-i\rangle$. Representa una rotación de 180° alrededor del eje Y, por lo que aplicándose dos veces vuelve al estado original; $Y^2 = I$.
- **Puerta de Pauli *Z*:** Conserva los estados base en módulo, pero introduce un cambio de fase relativa mediante una rotación de 180° alrededor del eje Z, por lo que se vuelve al estado original mediante $Z^2 = I$.
- **Puerta de fase *S*:** Equivale a la raíz cuadrada de la Z, introduce una fase compleja mediante una rotación de 90° alrededor del eje Z. Aplicarla dos veces significaría una transformación de Pauli Z, por lo que para volver al punto original se necesitan cuatro, de tal manera que $S^4 = I$.
- **Puerta *T*:** También conocida como puerta π/8, es la raíz cuadrada de la puerta S. Corresponde a una rotación de 45° alrededor del eje Z. Aplicándose de dos a cuatro veces se consigue equivaler a las puertas de fase S y Pauli Z, como $T^2 = S$; $T^4 = Z$.
- **Puerta de Hadamard *H*:** Una de las más utilizadas en computación cuántica, transforma los estados base en estados en superposición. Geométricamente,

representa una rotación de 180° alrededor del plano diagonal X+Z. Aplicándose dos veces se consigue la original $H^2 = I$.

Nombre de la puerta cuántica	Matriz unitaria	Acción
Identidad	$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$I\|0\rangle = \|0\rangle$ $I\|1\rangle = \|1\rangle$
Pauli *X*	$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$X\|0\rangle = \|1\rangle$ $X\|1\rangle = \|0\rangle$
Pauli *Y*	$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$Y\|0\rangle = i\|1\rangle$ $Y\|1\rangle = -i\|0\rangle$
Pauli *Z*	$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$Z\|0\rangle = \|0\rangle$ $Z\|1\rangle = -\|1\rangle$
Fase *S*	$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	$S\|0\rangle = \|0\rangle$ $S\|1\rangle = i\|1\rangle$
Puerta *T*	$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$	$T\|0\rangle = \|0\rangle$ $T\|1\rangle = e^{i\frac{\pi}{4}}\|1\rangle$
Hadamard *H*	$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$H\|0\rangle = \frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle) = \|+\rangle$ $H\|1\rangle = \frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle) = \|-\rangle$

Tabla 1. Representación matricial y acción de las principales puertas cuánticas

2.1.6. Múltiples qubits y entrelazamiento

Un sistema puede estar formado por múltiples qubits. Por ejemplo, con dos qubits ambos en estado $|1\rangle$, en estos casos se puede describir el estado general del sistema como un producto tensorial de los estados individuales de cada qubit, el estado del sistema se representa como el producto tensorial de los estados individuales de cada qubit:

$$|1\rangle \otimes |1\rangle = |11\rangle$$

En el algebra lineal, el producto tensorial es el producto Kronecker, que multiplica cada elemento de la primera matriz o vector por la correspondiente matriz o vector del segundo componente en su conjunto. Este comportamiento se extrapola desde 2 qubits a n qubits, obteniendo $N = 2^n$ estados base posibles. Por ejemplo, para dos qubits se corresponden 4 estados base de manera que la base Z quedaría como $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. Así pues, un estado general que conste de dos qubits es una superposición de estos estados base con 4 amplitudes, $c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle$. En un estado producto, la medición de un qubit no afecta a los demás, se pueden medir individualmente o en conjunto y en ambos casos se obtendrán las

mismas probabilidades. Por otro lado, para tantas amplitudes no existe forma de representación mediante la esfera de Bloch [2].

Según aumenta el número de qubits, para facilitar la notación se pueden describir los estados en decimal como $|0\rangle, |1\rangle, |2\rangle, \dots, |N-1\rangle$. De tal manera que un estado general de *n*-qubits es una superposición que se describe de forma genérica con la siguiente expresión.

$$|\psi\rangle = \sum_{j=0}^{N-1} c_j|j\rangle = c_0|0\rangle + c_1|1\rangle + \dots + c_{N-1}|N-1\rangle = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N-1} \end{pmatrix}$$

Existen estados cuánticos con múltiples bits que pueden factorizarse como el producto tensorial de qubits individuales (como el ejemplo de $|11\rangle$, que podía factorizarse en $|1\rangle$ y $|1\rangle$). Ahora bien, hay ciertos estados cuánticos que no pueden factorizarse en un producto de estados, y en estos casos se habla de entrelazamiento. Por ejemplo, se puede demostrar que el siguiente estado cuántico de dos qubits.

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \begin{pmatrix} 1/\sqrt{2} \\ 0 \\ 0 \\ 1/\sqrt{2} \end{pmatrix},$$

no puede representarse como el producto tensorial de dos qubits individuales,

$$|\phi_1\rangle = \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \quad \text{y} \quad |\phi_0\rangle = \begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix}.$$

Si un estado cuántico de *n* qubits es separable, es decir, puede representarse como una factorización de *n* qubits, entonces puede describirse de forma compacta con solo $2n$ coeficientes (2 por qubit) en lugar de los 2^n de la representación general. Por el contrario, un estado entrelazado general debe representarse con 2^n coeficientes. La existencia de estados entrelazados explica por qué un computador cuántico no puede ser simulado eficientemente por un computador clásico [2].

En los estados entrelazados, los qubits dejan de ser completamente independientes, pues la medición de un qubit puede afectar a los otros qubits (algo que no ocurre en los estados producto) [6].

2.1.6.1. Estados de Bell y puerta CNOT

Los estados de Bell son la forma más simple y fundamental de representar un entrelazamiento máximo entre un par de qubits. Entrelazar máximamente, significa que, la medida de un qubit entrelazado determina completamente el estado del segundo qubit, ya que la correlación entre sus propiedades es perfecta.

Con dos qubits existen cuatro estados de máximo entrelazamiento, que son los estados de Bell, formando una base ortonormal denominada base de Bell.

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

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

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

La puerta CNOT (o puerta NOT controlada) es una puerta lógica cuántica de dos qubits. Esto significa que opera sobre dos qubits simultáneamente. De forma general, se define como CNOT $_{ij}$ de tal manera que, en *i* o *j*, se indica cuál de las dos posiciones de un estado se fija como qubit de control con un '1' y cuál tendrá la función de objetivo con un '0'.

En este caso por defecto, es la CNOT_{10} por lo que para un estado |xy⟩ al que se quiere aplicar esta puerta, se controla el qubit izquierdo y se aplican los cambios al qubit derecho. De forma que cuando se aplique la puerta CNOT a un estado |xy⟩ si el qubit izquierdo es 1 el resultado en la salida será mantener dicho 1 y aplicar una puerta NOT al qubit derecho y si el qubit izquierdo es 0 se mantienen ambos a la salida.

$$\text{CNOT}|00\rangle = |00\rangle$$

$$\text{CNOT}|01\rangle = |01\rangle$$

$$\text{CNOT}|10\rangle = |11\rangle$$

$$\text{CNOT}|11\rangle = |10\rangle$$

Esa puerta ofrece una transformación crucial porque es el medio más simple para crear un estado entrelazado partiendo de dos qubits, como se puede ver con la siguiente demostración de ejemplo (y de forma similar para conseguir el resto de los estados de Bell).

$$\text{CNOT}|+\rangle|0\rangle = \text{CNOT}\ \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\phi^+\rangle$$

2.1.1. No clonación

El teorema de no clonación establece que es imposible crear una copia idéntica de un estado cuántico desconocido arbitrario $|\psi\rangle$ en estado de superposición, lo que impide que un espía duplique los qubits sin ser detectado.

La clonación sería posible si existiera una puerta *U* tal que al aplicarla sobre un qubit desconocido arbitrario $|\psi\rangle$ provocara una transformación en un qubit auxiliar $|0\rangle$, de manera que se obtuvieran dos copias iguales en vez de únicamente la original, esto es,

$$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$$

Existen muchas soluciones al sistema matricial resultante, pero en ellas se requiere conocer las amplitudes $\alpha\ y\ \beta$ del estado original $|\psi\rangle$ y esas amplitudes son desconocidas, porque para conocerlas habría que medir y eso colapsaría el estado. Así pues, no existe un operador *U* que nos permita copiar un estado cuántico general desconocido.

Por otro lado, si el estado fuera conocido, sí que sería posible clonarlo, pero en este caso sería más correcto referirse a que se realiza una copia, lo que es útil para ciertos procesos de verificación de operaciones en situaciones controladas.

Se pueden aplicar distintas puertas para llegar a transformar los estados como si fueran copias controladas. Como podría ser, la propia puerta $\text{CNOT}|10\rangle = |11\rangle$ o aplicar operaciones con puertas simples para conseguir, por ejemplo, mediante las puertas Identidad y Hadamard, la siguiente transformación.

$$(I \otimes H)(|-\rangle|1\rangle) = |-\rangle|-\rangle$$

2.2. Criptografía y computación cuántica

La criptografía es la ciencia o disciplina general que estudia, mediante diversas técnicas matemáticas, cómo proteger la información mediante algoritmos y protocolos, utilizados para procesos como cifrados, firmas digitales o funciones hash. Mantener la seguridad y la confidencialidad de la información, se ha convertido en un objeto principal de estudio frente a las crecientes amenazas, influyendo en numerosas áreas de investigación llegando hasta la computación cuántica.

En este apartado se presentan y comparan los desafíos que suponen los desarrollos de la computación cuántica frente a la criptografía clásica y cómo surgen nuevas soluciones.

2.2.1. Desafíos de la criptografía y el cifrado

Según la referencia [7], la evolución histórica del desarrollo cuántico destaca dos eventos importantes. El primero, un artículo publicado en 1982, donde Richard Feynman planteó que un ordenador basado en principios cuánticos podría simular fenómenos físicos de forma eficiente aprovechando propiedades como la superposición y el entrelazamiento, cuyas características hemos planteado en el apartado 2.1. Años después, en 1994, Peter Shor presentó un algoritmo cuántico

capaz de factorizar números enteros de gran tamaño en un tiempo polinomial. Esto presenta un escenario con un gran salto frente a la capacidad de cálculo clásica, cuya complejidad sigue siendo superpolinómica es decir, que pueden volverse impracticables debido a las enormes cantidades de tiempo que tardan en ejecutarse [8], [2].

La importancia de esto es crítica, ya que los sistemas criptográficos modernos como el cifrado asimétrico, utiliza dos claves matemáticamente relacionadas, una pública para cifrar y una privada para descifrar, de tal manera que su seguridad depende de la dificultad de cálculo clásica de la factorización de enteros, como ocurre con el algoritmo RSA (*Rivest, Shamir, and Adleman*) y en la dificultad de resolver el problema del logaritmo discreto, como ocurre con ECC (*Elliptic Curve Cryptography*), DH (*Diffie–Hellman*) y DSA (*Digital Signature Algorithm*), por lo que estos algoritmos criptográficos no son seguros tras la implementación práctica del algoritmo de Shor

Posteriormente, en 1996, se propuso el algoritmo de Grover [9], que permite realizar búsquedas en bases de datos no estructuradas o encontrar colisiones de funciones hash, donde dos entradas distintas producen la misma salida, reduciendo el tiempo de búsqueda con *N* elementos desde un orden clásico de $O(N)$ a un orden cuántico de $O(\sqrt{N})$, el equivalente a una mejora cuadrática en la complejidad de la búsqueda.

Con respecto a los cifrados basados en cifrado simétrico, su proceso utiliza cifrado en bloques de tal manera que una sola clave se utiliza tanto para cifrar como para descifrar, lo son, AES (*Advanced Encryption Standard*), 3DES (*Triple Data Encryption Standard*), "Blowfish" o RC4. En este caso, su seguridad se basa en la dificultad computacional de encontrar la clave por fuerza bruta. Frente a la computación cuántica, los algoritmos simétricos son más robustos porque la reducción de complejidad que ofrece el algoritmo de Grover mantiene el problema en el ámbito polinómico [8], reduciendo su complejidad de búsqueda de 2^N a $2^{N/2}$ [7], que es otra manera de expresar la mejora cuadrática.

Por lo tanto, una clave de 128 bits acabaría ofreciendo frente a un atacante cuántico 64 bits de seguridad efectiva. De esta manera, la criptografía simétrica no está en peligro inmediato por la computación cuántica, sino que requiere ajustes en los tamaños de clave para garantizar seguridad a largo plazo. Tanto [7] como [8], consideran que AES-256 y funciones hash de superiores 256 bits siguen siendo seguras en el contexto cuántico ofreciendo 128 bits de seguridad.

El único esquema de cifrado con seguridad teórica de la información (ITS), es decir, una seguridad absoluta e independiente de la capacidad de cómputo del atacante, es el cifrado de Vernam o de "One-Time Pad" (OTP). Utiliza una clave del mismo tamaño que el mensaje, completamente aleatoria y siendo utilizada una única vez, criterios que estableció Claude Shannon en 1949 [10]. Cumpliendo esto, el OTP garantiza confidencialidad perfecta [8]. En la práctica, puede ser difícil de gestionar si se aplican claves muy grandes, que compliquen su gestión, por lo que es poco práctico.

Por otro lado, surge la criptografía postcuántica, que describe un conjunto de algoritmos que han demostrado ser seguros frente a los ataques cuánticos conocidos hasta el momento. Estás basados en técnicas de retículos (*Lattice-based*), basados en hash (*hash-based*), o cuadráticas multivariantes (*multivariate Quadratic*), y son desarrollos para garantizar la resistencia criptográfica de las comunicaciones frente a futuros avances en la computación cuántica [7].

La Tabla 2, compara las amenazas que se imponen a los diferentes sistemas descritos, en presencia de algoritmos cuánticos, si se quiere información adicional, o profundizar sobre la criptografía y los distintos tipos mencionados. Para profundizar, se recomienda la referencia [11].

Categoría	Tipos	Base de seguridad	Amenaza cuántica	Estado frente a computación cuántica
Asimétrica	RSA, ECC, DH, DSA	Factorización de enteros (RSA) y Logaritmo discreto (DH/ECC)	Shor, factoriza y resuelve el log. discreto en tiempo polinomial	Insegura
Simétrica	AES, 3DES, Blowfish, RC4	Fuerza bruta sobre el espacio de claves	Grover, búsqueda cuadráticamente más rápido	Segura si sube su tamaño de clave (AES-256 ≈ 128 bits de seguridad efectiva cuántica)
	OTP	Shannon	Ninguna	Seguro absoluto; cifrado XOR con clave aleatoria del mismo tamaño
Postcuántica (PQC)	Lattice-based, Code-based, Hash-based	Problemas no vulnerables a Shor/Grover	No conocido	Seguras en teoría (pendiente estandarización NIST)

Tabla 2. Tipos de criptografía frente a computación cuántica

2.2.1.1. Funciones hash y algoritmos de autentificación

El cifrado, o encriptación, es el proceso de transformar un mensaje legible, como texto plano, en uno ilegible como texto cifrado utilizando en el proceso un algoritmo y una clave. Una comunicación segura puede necesitar también apoyarse sobre una autentificación para verificar la validez de la transmisión entre dos partes.

El algoritmo de Hash Seguro (SHA) se define como una familia de algoritmos con el objetivo fundamental de garantizar la integridad de los datos, ya que un cambio mínimo en la entrada produce una salida completamente diferente. Toman una entrada de datos de cualquier tamaño y producen una salida de longitud fija conocida

como resumen hash, o "*digest*" [12]. La denominada, función hash, es la que se encarga de obtener dicha cadena de bits de salida de longitud fija.

Están diseñados para ser unidireccionales, lo que significa que es computacionalmente inviable revertir el proceso para obtener la entrada original. Se divide en SHA-1, SHA-2 y SHA-3; en concreto actualmente el más utilizado es SHA-2, que incluye SHA-224, SHA-256, SHA-384 y SHA-512. Estas versiones, no tienen "ataques de colisión eficientes", lo que significa que no es factible encontrar dos entradas distintas que generen la misma salida de manera práctica y que pueda aprovechar una atacante. Se considera robusta al tener una probabilidad de colisión en la salida inferior a $1/2^{64}$. El número, por ejemplo, de SHA-256, indica que la longitud del hash de salida tiene un tamaño fijo de 256 bits y utiliza palabras de 32 bits. Mas detales pueden encontrarse en su especificación del NIST [12]. El algoritmo de Grover puede acelerar la búsqueda de fuerza bruta, pero la tarea de encontrar el mensaje original a partir de su hash sigue siendo computacionalmente inviable para estas salidas.

Su versión más moderna SHA-3 es una familia de funciones de hash seguro para datos binarios, seleccionada por el NIST como un estándar criptográfico. Utiliza una estructura nueva, conocida como la construcción de esponja "sponge construction" del algoritmo KECCAK, pero con el mismo fin que SHA-2, conseguir una salida única. Más detalles de su funcionamiento pueden encontrarse en su especificación [13].

Por otro lado, también existen algoritmos de autentificación de mensajes, MAC (*Message Authentication Code*), como VMAC. Concretamente sigue el estilo expuesto por Wegman-Carter empleando una función hash universal (VHASH) para generar una cadena corta [14]. Combinadas con un cifrado de un solo uso su resultado sería la etiqueta de autentificación MAC [15]. Puede producir etiquetas de 64 o 128 bits, donde que un atacante consiga falsificar una etiqueta es inferior a $1/2^{60}$ y $1/2^{120}$ respectivamente [16].

La Tabla 3, expone los algoritmos de autentificación expuestos y su estado frente a la computación cuántica.

Categoría	Ejemplos	Base de seguridad	Estado frente a computación cuántica
Funciones hash	SHA-2, SHA-3	Resistencia a colisiones y mejora de diseño con algoritmo KECCAK (en SHA-3)	Seguro frente a Grover con salidas ≥ 256 bits
MAC universal hashing	VMAC	Hashing universal + seguridad del cifrado asociado	Seguro frente a Grover con claves largas. Recomendado AES (≥256 bits)

Tabla 3. Tipos de funciones hash y MAC frente a computación cuántica

2.2.2. Consolidación de QKD

El problema central de la criptografía moderna no es tanto diseñar algoritmos de cifrado, sino cómo establecer de manera segura las claves secretas que estos algoritmos requieren. Este problema, conocido como acuerdo de claves secretas, busca que dos usuarios distantes, obtengan un mismo valor secreto garantizando que ningún tercero pueda conocerlo o interferir sobre él.

La distribución cuántica de claves (QKD, "Quantum Key Distribution"), Tabla 4, constituye una solución de alta tecnología en el ámbito de la criptografía y de las comunicaciones seguras y fiables, ya que emplea los principios fundamentales vistos en el apartado 2.1, como la superposición o el entrelazamiento cuántico, estableciendo entre dos partes un protocolo de intercambio seguro de claves criptográficas como BB84 o EE91, para posteriormente emplear un algoritmo de cifrado clásico, como el cifrado simétrico, para proteger los mensajes confidenciales que se van a transferir entre dos puntos.

La seguridad que proporciona es teóricamente invulnerable, lo que la hace resistente a una amplia gama de ataques, ampliando aquellos a los que ya se enfrentan actualmente los sistemas de seguridad presentes en la computación clásica. El proceso se fundamenta principalmente en las leyes de la naturaleza descritas por la física cuántica. No se necesita ninguna suposición computacional, por lo tanto, el protocolo es inmune a cualquier atacante independientemente de su poder computacional.

Pese a esto, si bien el ámbito de la criptografía cuántica ofrece una seguridad teóricamente inquebrantable, en general su adopción a gran escala se ve limitada por barreras tecnológicas como su rango de uso, los altos costes y la necesidad de equipos especializados en ciertos casos difíciles de miniaturizar. A pesar de los desafíos, QKD se posiciona como la tecnología de criptografía cuántica más madura habiendo empezado ya con su comercialización, lo que demuestra su viabilidad para aplicaciones prácticas [17].

Categoría	Ejemplos	Base de seguridad	Estado frente a computación cuántica
Criptografía cuántica	QKD (BB84, E91, etc.)	Seguridad basada en principios cuánticos	Protege la información frente a terceros aun siendo cuánticos. Requiere aplicar autenticación clásica y cifrado seguro para proteger el conjunto del sistema.

Tabla 4. Seguridad de criptografía cuántica con QKD

Capítulo 3

3. Fundamentos de un sistema QKD

En este capítulo se van a tratar los apartados esenciales para entender la estructura general correspondiente a un sistema completo que utilice QKD: su objetivo, medios de transmisión, enfoques de implementación, protocolos aplicables, componentes esenciales, procesos aplicados y la división de la arquitectura del modelo de red.

3.1. Objetivo de QKD

La distribución cuántica de claves QKD, se dispone como una de las aplicaciones más relevantes de los sistemas cuánticos, ya que permite a dos partes el establecimiento de una clave secreta con seguridad garantizada principalmente por las leyes de la mecánica cuántica. El principal objetivo es establecer un procedimiento o método para generar y distribuir claves criptográficas simétricas entre usuarios separados geográficamente unidos entre sí por una red que cumpla con el requisito de conseguir la seguridad teórica de la información, ITS [18].

Se requiere la capacidad de producir, manipular, transmitir y medir señales cuánticas. En el caso de las telecomunicaciones, estas señales cuánticas son principalmente fotones que, a través de su polarización, pueden definir estados cuánticos distintos según esta sea, polarización horizontal, vertical, o cualquier superposición de ambas, siendo esto un sistema análogo a lo explicado en el apartado 2.1.2. No obstante, en teoría cualquier sistema cuántico con dos estados distintos puede ser utilizado como qubit de la misma manera, como por ejemplo, los distintos niveles de energía de iones, las propiedades del spin de un electrón o ciertas características de circuitos superconductores [2].

La forma en que QKD permite generar claves secretas con seguridad incondicional, es combinándose con un algoritmo de cifrado, vistos en el apartado 2.2.1, como OTP, y así se obtiene un protocolo compuesto que es igual de incondicionalmente seguro. Esto se debe a que QKD, por sí mismo, no cifra directamente los mensajes que las partes desean intercambiar, sino que únicamente distribuye las claves secretas. Para que dichas claves tengan un sentido práctico manteniendo la seguridad, es necesario aplicar un protocolo de cifrado clásico que permita proteger la confidencialidad de los datos transmitidos a través de un canal público. Esto es posible gracias al marco de composabilidad universal (UC, *universal composability*) de seguridad de QKD, que asegura que, si este se integra con otros protocolos igualmente seguros, el sistema resultante sigue manteniendo esa seguridad [8].

3.1.1. Medios de transmisión

Según el punto de vista ofrecido por [19] y [6], se analizan dos medios convencionales de transmisión utilizados como canales para implementar QKD, la fibra óptica y el espacio libre. Sus característica y matices se presentan brevemente a continuación, se puede encontrar más información técnica de implementaciones en [20].

- **Fibra óptica**

La tecnología QKD que se despliega utilizando fibra óptica ha logrado un gran avance y ya está disponible en el mercado. La razón principal de su éxito es que la fibra óptica ofrece una baja pérdida de señal y una alta estabilidad, lo que la convierte en el medio ideal para transmitir las complejas señales cuánticas. El objetivo de los estudios de investigación centrados en QKD sobre fibra es ampliar la distancia alcanzable de la red y aumentar la tasa a la que se pueden generar las claves seguras para suministrar a posibles aplicaciones más exigentes [21].

La degradación de las señales cuánticas debida a la absorción y al ruido supone una limitación en la distancia alcanzable entre los enlaces punto a punto de la red, limitándose teóricamente a unos cientos de kilómetros. La pérdida habitual de la fibra es de aproximadamente 0,21 dB/km para las longitudes de onda típicas de telecomunicaciones [1] como 1550 nm. Debido a ello, según [22], los sistemas QKD comerciales toleran alrededor de 20-30 dB de perdida en distancias de entre 100 y 150 kilómetros.

Esto impide que las comunicaciones cuánticas alcancen simultáneamente altas velocidades y largas distancias [23]. Este reto se ve agravado por las pérdidas inherentes a posibles dispositivos intermedios como divisores, filtros y multiplexores, así como por el ruido de los detectores ópticos [1], [22]. Lo que, en consecuencia, limita más la distancia máxima que se puede implementar con QKD, disminuyendo a medida que aumentan las pérdidas.

- **Espacio libre**

Este proceso de distribución cuántica de clave a través del aire, o espacio libre, está en una fase de desarrollo menos madura que los sistemas QKD basados en fibra óptica. Aunque se ha demostrado su viabilidad en laboratorios con distancias cada vez mayores hasta 500 kilómetros, actualmente se sigue investigando con el objetivo de lograr aplicaciones prácticas más robustas, que incluso plantean y realizan esquemas híbridos donde fibra óptica y espacio libre se integran juntas como una red de comunicación cuántica espacio-tierra [24], consiguiendo cubrir una distancia de 4600 kilómetros, hacia una idea de red global.

Su transmisión debe tener en cuenta que unas condiciones como, una situación atmosférica adecuada, una trayectoria despejada y una relación señal-ruido (SNR) aceptable. Lo que acaba limitando estrictamente el tiempo y rango de uso disponible [1].

La relación que ofrece [19] y [6] entre ambos medios, se pueden observar a continuación en la Tabla 5, las características que ofrecen de forma general, según su coste, madurez, flexibilidad, estabilidad y comercialización.

Características	Espacio libre	Fibra óptica
Madurez	Baja	Alta
Estabilidad	Baja	Alta
Flexibilidad	Alta	Baja
Coste	Alto	Bajo
Comercialización	En desarrollo	Disponible
Futuro	Complementarse hacia una red global	

Tabla 5. Comparativa de medios de transmisión en QKD

3.2. Implementación discreta y continua

Dentro de los esquemas de QKD, existen dos enfoques principales: los protocolos de variable discreta (DV-QKD) y los de variable continua (CV-QKD). Ambos se apoyan en principios cuánticos comunes, pero difieren en la forma de codificar y procesar la información.

3.2.1. DV-QKD

Los sistemas DV-QKD (*Discrete Variable QKD*) son aquellos que utilizan propiedades discretas. La información se representa en estados cuánticos discretos como los de un fotón individual utilizando, su polarización, fase o intervalo temporal [19].

En este marco, el transmisor idealmente debería generar fotones únicos, aunque en la práctica se emplean pulsos láser atenuados para aproximar esta condición. El receptor utiliza detectores de fotones individuales, y la seguridad del protocolo depende en gran medida de la eficiencia de estos dispositivos que, además, como hemos visto, también condicionan la distancia máxima alcanzable [19].

Un enfoque aplicable se basa en la idea de preparar y posteriormente medir dichos fotones polarizados, de tal manera que, cuando un fotón polarizado se mide en la misma base en la que fue preparado, el resultado es determinista, ya que el receptor siempre obtiene el valor esperado. Sin embargo, si la medición se realiza en una base

distinta como por ejemplo sería, medir un fotón que fue preparado con polarización horizontal utilizando la base diagonal, el resultado es aleatorio con probabilidad uniforme, debido a que, tras la medición, el estado del fotón colapsa en la nueva base y se destruye el estado original [25]. Aplicando los conceptos algebraicos expuestos en el apartado 2.1.4, podríamos ver matemáticamente los mismos resultados, que se muestra a continuación en la Figura 4. Este comportamiento será utilizado más adelante en el apartado 3.3.1, cuando se trate el principal protocolo de variables discretas BB84.

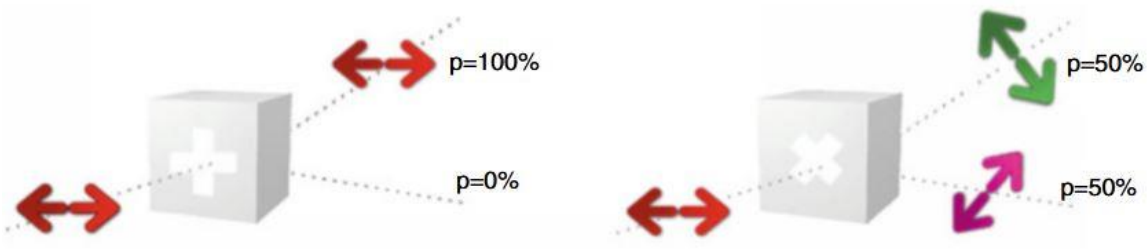


Figura 4. Probabilidad de detectar fotón horizontal con base rectilínea o diagonal [25]

De estas propiedades se deriva también la seguridad que aporta la imposibilidad de clonar los estados cuánticos desconocidos. Un espía no puede copiar un fotón arbitrariamente polarizado para medirlo sin introducir perturbaciones detectables, alterando el estado y provocando alteraciones en la secuencia de comunicación.

Por tanto, la combinación de preparación en bases según la teoría cuántica y la naturaleza destructiva de la medición, garantizan que cualquier intento de interceptar los fotones por la red genere errores que pueden ser identificados por los participantes legítimos. En conjunto, estas características de los fotones polarizados constituyen el fundamento de los protocolos de tipo "*prepare-and-measure"* utilizado en variables discretas, como es el caso de BB84, ya mencionado.

El enfoque aplicado en protocolos de distribución cuántica de clave de variable discreta, no se limita únicamente a la preparación y medición de estados cuánticos. De hecho, otros protocolos, como el E91, se basan en aplicar las propiedades que ofrece el entrelazamiento cuántico.

Estos sistemas basan su seguridad en las fuertes correlaciones entre partículas entrelazadas, que permiten a las partes generar una clave compartida a partir de mediciones consistentes [20], utilizando por ejemplo las bases de Bell introducidas en el apartado 2.1.6.1. También, los fotones deben crearse en pares entrelazados, lo que es más difícil y repercute en un rendimiento menor que cuando se utiliza una producción de fotones individuales mediante la atenuación de un pulso láser [22].

3.2.1. CV-QKD

Otra opción más reciente, es utilizar el enfoque que presenta la variable continua, CV-QKD (*Continuous Variable QKD*). En este caso se utilizan las cuadraturas del campo eléctrico de la onda electromagnética, su amplitud y fase, y se prescinde del uso de qubits codificados en fotones individuales [19]. Son variables continuas en forma de señal de tal manera que los efectos cuánticos sean accesibles para codificar información cuántica y comprobar si se encuentran perturbaciones de un posible espía, cumpliendo con el propósito de ITS.

A diferencia de los sistemas de variable discreta (DV-QKD) que dependen de detectores de fotón único, como SDPs de silicio o InGaAs [26]. CV-QKD emplea detección homodina o heterodina, basa en la superposición de la señal cuántica con un oscilador local clásico [25]. Esta técnica traslada métodos utilizados en las telecomunicaciones clásicas al ámbito cuántico. Su principal ventaja es que aprovecha la tecnología ya existente, lo que permite reducir significativamente los costes de producción y facilita la integración a gran escala de estos sistemas con los componentes optoelectrónicos, mediante un esquema práctico de preparación y medición de estados coherentes.

Este modelo exige condiciones de bajo ruido para preservar la integridad de la señal cuántica, pero sus propiedades le permiten la co-propagación del canal cuántico, es decir, junto con canales de comunicación clásicos en la misma fibra óptica, una capacidad que no es viable en la mayoría de los sistemas DV-QKD. Esta multiplexación de canales posibilita la integración de la CV-QKD en infraestructuras de telecomunicaciones existentes sin necesidad de desplegar nuevas redes de fibra dedicadas, lo que la convierte en una solución prometedora para redes de comunicación seguras a gran escala [27].

3.3. Protocolos QKD

Una implementación QKD se puede llevar a cabo utilizando varios protocolos. Estos se pueden dividir en diferentes tipos según si utilizan explícitamente el entrelazamiento o no. También, según si se trabaja con un protocolo de preparación y medición del qubit, o según si se utilizan variables discretas o continuas. Su implementación es muy diferente mostrando diferentes fortalezas y debilidades, pero desde el punto de vista de la seguridad, todos ellos han sido demostrados como ser seguros [22].

3.3.1. BB84

Es un protocolo de QKD publicado por Bennett and Brassard en 1984 [28], y es el primer protocolo que implementa la QKD. Opera con tres actores principales: un emisor (Alice), un receptor (Bob) y un potencial espía (Eve).

Necesita un canal cuántico, por el que compartir las señales cuánticas y un canal clásico público autenticado entre Alice y Bob, para que se comuniquen mediante mensajes utilizando métodos tradicionales. Eve, puede escuchar esa conversación pública, pero, como se explicará más adelante, no puede participar ni obtener información. Cualquier intento de intervenir, en el canal cuántico alteraría las señales, lo que puede ser detectado por Alice y Bob. Es por esto por lo que se definen los siguientes métodos y procesos para garantizar el éxito del protocolo QKD.

El protocolo BB84 codifica la información en qubits mediante cuatro estados distintos, que se organizan en dos bases no ortogonales, una rectilínea (horizontal y vertical) y otra diagonal (de ±45°) [19], resultando como R{$|H\rangle, |V\rangle$} y D{$|45°\rangle, |-45°\rangle$}, que serían el equivalente matemático en la esfera de Bloch a R{$|0\rangle, |1\rangle$} y D{$|+\rangle, |-\rangle$} por sus propiedades probabilísticas y que corresponden a los bits clásicos 0, 1, 1 y 0, respectivamente como se muestra en la Figura 5.

Esto significa que, como se introdujo en el apartado de DV-QKD, un espía al medir el fotón no va a poder determinar con absoluta certeza cual fue el estado inicial enviado, introduciendo incertidumbre al sistema.

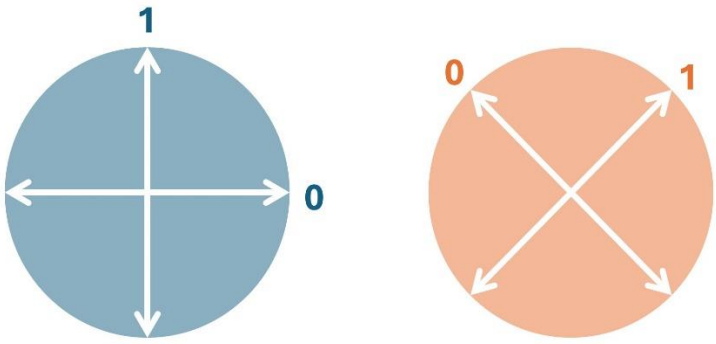


Figura 5. Polarización del fotón en las bases rectilínea y diagonal

El proceso consta de 5 fases, reflejadas en Figura 7 junto con los componentes principales del protocolo [19], en primer lugar:

1. **Codificación cuántica (Preparación del qubit, transmisión y medida):**

   Alice genera una cadena de bits clásicos, denominada como clave en bruto "*raw key*", para codificarlos en un flujo de fotones individuales, de manera que cada fotón posee uno de los cuatro estados de polarización de las dos bases R y D.

   Alice procede a enviar los qubits a Bob a través del canal cuántico. Bob los recibe, los mide al azar en una base de polarización y procede a registrar las bases de medición utilizadas con los resultados.

2. **Cribado (“*Sifting*”):**

   Alice y Bob comparten las bases que utilizaron para medir a través de un canal clásico autenticado.

   En esta discusión pública, ambos anuncian sus bases de polarización, pero no sus resultados. A continuación, descartan los eventos en los que no concuerdan. Los qubits restantes correspondientes a las bases coincidentes se decodifican en un flujo de bits denominado clave cribada o “*sifted key*”, como se ve en la Figura 6, denominado únicamente (“*key*”) [7].

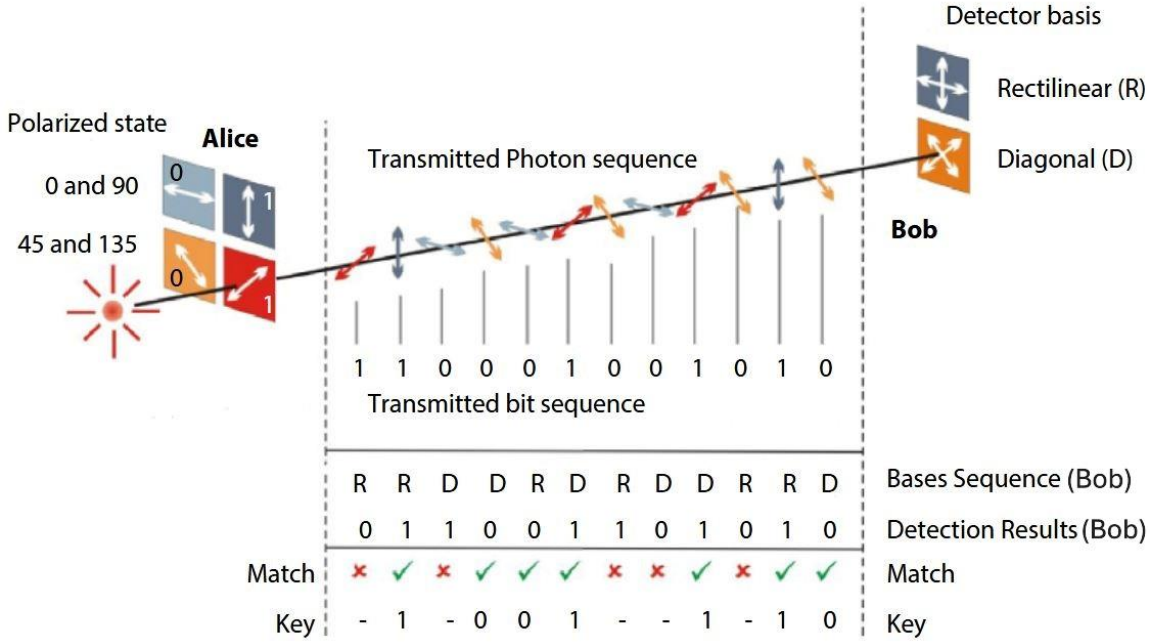


Figura 6. Proceso de obtención de clave con BB84 [7]

En este momento, se han podido producir errores debido a fotones de fondo, ruido del detector, imperfecciones de la polarización o por influencia de un espía [29]. Es por esto por lo que se procede al siguiente paso:

3. **Estimación de errores:**

   Antes de avanzar, Alice y Bob deben verificar la calidad de la clave cribada obtenida, estimando la tasa de error de bits cuánticos QBER (*Quantum Bit Error Rate*) [19]. Esta estimación, se utiliza para obtener un indicador que revele la presencia de un espía en la clave cribada.

   El proceso consiste en seleccionar aleatoriamente un subconjunto de bits de la clave cribada para sacrificarlos anunciándolos públicamente y así comprobar si hay alguna discrepancia, contabilizando los errores encontrados entre ambas partes y dividiéndolos entre el total compartido.

$$QBER = \frac{Nº\ de\ bits\ erroneos\ encontrados}{Nº\ de\ bits\ totales\ del\ subconjunto\ seleccionado}$$

Si la discrepancia se encuentra por debajo de un valor umbral predeterminado, generalmente menor al 5%, se continúa; en caso contrario, el proceso de distribución de clave debe cancelarse y reiniciarse desde la primera etapa [30]. Existen métodos alternativos a este proceso para estimar la QBER [6].

La probabilidad teórica de que Eve logre espiar sin ser detectada en una muestra de *n* bits, se denomina probabilidad de fallo de la clave "*key failure probability*" y típicamente se busca que sea inferior a 10^{-10} [19]. Se refleja en la siguiente fórmula basada en el comportamiento probabilístico de medir las bases aleatoriamente, donde para cada bit, Eve tiene un 50 % de elegir la base correcta sin introducir error y pese a elegir la base equivocada, aún conserva un 50 % de no generar discrepancia, esto equivale a 3 de 4 posibilidades en las que Eve no introduce error en un bit de prueba, repitiendo el proceso para *n* bits de prueba comparados [2] se obtiene:

$$p = \left(\frac{3}{4}\right)^n$$

De manera teórica, una cantidad de 81 bits cumpliría este valor.

$$(n = 81)\ ;\ p = 7{,}59 \cdot 10^{-11} < 10^{-10}$$

Para asegurar la clave final con mayor seguridad, hay que tener en cuenta otras dos fases, que se comentan a continuación, en donde se procede a mejorar la seguridad de la clave final.

4. **Postprocesamiento:**

A través del canal clásico Alice y Bob realizan una serie de procesos para seguir mejorando la clave final obtenida, estas técnicas se engloban generalmente en tres grupos denominados corrección de errores ("*Error Key Reconciliation*"), verificación de clave y amplificación de la privacidad ("*Privacy Amplification")*. La cadena final de bits de clave resultante será denominada finalmente como clave secreta, "*secret key*" [6], [19].

Se aplica la corrección de errores para corregir cualquier error restante en los bits de clave, ya que incluso si Alice y Bob determinan la ausencia de Eve con una estimación de QBER baja, tendrán que corregir errores residuales causados por efectos naturales, como las imperfecciones de los medios ópticos, las fuentes, los detectores de señales u otros fenómenos en el transporte y la medición de fotones, por ejemplo, con los métodos Cascade o LDPC [25], [29]. Posteriormente se confirma que la corrección de errores ha sido exitosa verificando una pequeña muestra de clave.

Una vez considerada libre de errores, la técnica de ampliación de la privacidad se aplica para terminar de asegurar que Eve no tenga información. Se puede aplicar una función hash unidireccional, como las introducidas en el apartado 2.2.1.1, para extraer la clave final a partir de los bits de clave resultantes del proceso de corrección de errores. Si se sigue el procedimiento correctamente, se extrae una clave secreta final más corta, pero completamente segura [30].

5. **Autenticación:**

Es importante señalar la necesidad de autenticar las comunicaciones del canal clásico. De lo contrario, se podría llevar a cabo un ataque de tipo "*man in the middle*" [22], ciberataque en el que un atacante situado entre Alice y Bob intercepta el mensaje de Alice y envía su propio mensaje a Bob, mientras que tanto Alice como Bob creen que están comunicándose directamente entre ellos. Es por esto por lo que, el emisor y el receptor deben acordar un método de autenticación. La seguridad del canal cuántico solo funciona si Alice sabe que está hablando con Bob y viceversa.

Se lleva a cabo desde la primera comunicación de QKD por el canal clásico, ya que la primera sesión de QKD se autentica la conversación utilizando una clave secreta previamente compartida entre Alice y Bob [6]. Se puede utilizar hash o MAC y sirve para verificar que los mensajes del canal clásico son auténticos y no manipulados.

Las sesiones posteriores de QKD se pueden autenticar utilizando una pequeña parte de las claves secretas acordadas para evitar el ataque de "*man in the middle*" de Eve [19].

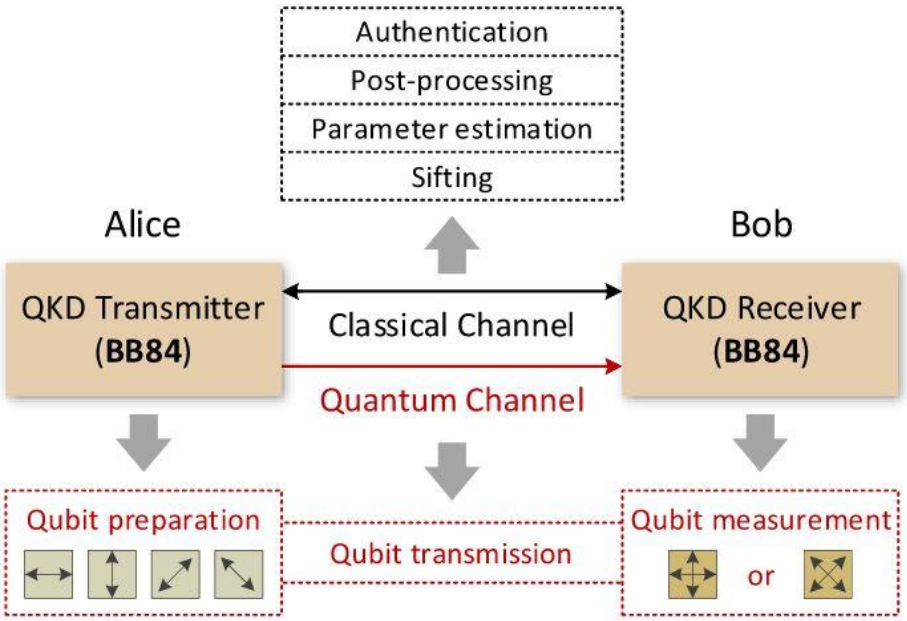


Figura 7. Esquema de componentes y fases del protocolo BB84 [19]

En el protocolo QKD BB84, la distancia física entre nodos es un factor determinante: cuanto mayor es la longitud de fibra óptica o el canal cuántico, mayores son las pérdidas de fotones y el ruido, lo que incrementa el QBER y reduce drásticamente la tasa de clave secreta final.

3.3.1. Otros enfoques

Dentro de los protocolos de DV-QKD se encuentran también los protocolos basados en fotones entrelazados, que son protocolos que utilizan las propiedades de los estados cuánticos entrelazados de la mecánica cuántica. A diferencia de un protocolo de fotón único, estos protocolos se comunican con un par de fotones entrelazados entre sí. En los protocolos de fotones entrelazados, el primero fue denominado E91 ya que fue propuesto por Artur Ekert en 1991 [31]. Posteriormente en 1992, se propuso el protocolo BBM92 [32].

A continuación, se presenta una tabla de referencia de las implementaciones típicas de QKD, basada en las diferentes opciones de implementación presentadas expuestas en [6] y [19].

Nombre del Protocolo	Tipo	Enfoque	Año
BB84	DV	Preparación y medida	1984
E91	DV	Entrelazamiento	1991
BBM92	DV	Entrelazamiento	1992
GG02	CV	Preparación y medida	2002
DPS	DV	Preparación y medida	2002
Decoy-state	DV	Preparación y medida	2003–2005
SARG04	DV	Preparación y medida	2004
COW	DV	Preparación y medida	2005
MDI	DV/CV	Preparación y medida	2012
TF	DV	Preparación y medida	2018
PM	DV	Preparación y medida	2018

Tabla 6. Listado de protocolos de QKD

3.4. Componentes esenciales físicos y lógicos

Los componentes principales que forman parte de un sistema QKD básico se dividen en una parte fundamental de comunicación cuántica y otra clásica, los podemos ver representados sobre el esquema de la Figura 8.

- En la capa cuántica, encontramos al emisor (Alice) y al receptor (Bob), conectados por un canal cuántico ($\mathbb{Q}$) sobre fibra óptica o espacio libre, como se expuso en el aparatado 3.1.1, por donde viajará la información cuántica, por ejemplo, los estados cuánticos mediante fotones individuales.
- Además, se requiere un canal clásico ($\mathbb{C}$) autenticado, que permite a las partes intercambiar determinada información auxiliar de manera convencional sin riesgo de manipulación externa.

Se define también un tercer componente, en este caso actuando como un malicioso espía en la comunicación entre Alice y Bob, que se denomina Eve y sirve para plantear posibles escuchas que interfieran en la comunicación entre Alice y Bob, la Figura 8 muestra esta interferencia como $\rho \rightarrow \rho'$.

Un módulo QKD es una solución completa que integra las funciones cuánticas y clásicas, concretamente, es un conjunto de componentes de hardware y software contenidos dentro de un límite criptográfico definido, que implementa funciones criptográficas y procesos óptico-cuánticos como protocolos de distribución cuántica de claves QKD, sincronización y postprocesado para la generación de claves seguras [33]. En la literatura también se encuentra referenciado como "*QKD device*", ya que funciona como un dispositivo de QKD que se puede presentar todo junto, dividido en un subsistema cuántico y otro clásico, o como QKDE refiriéndose a entidades QKD.

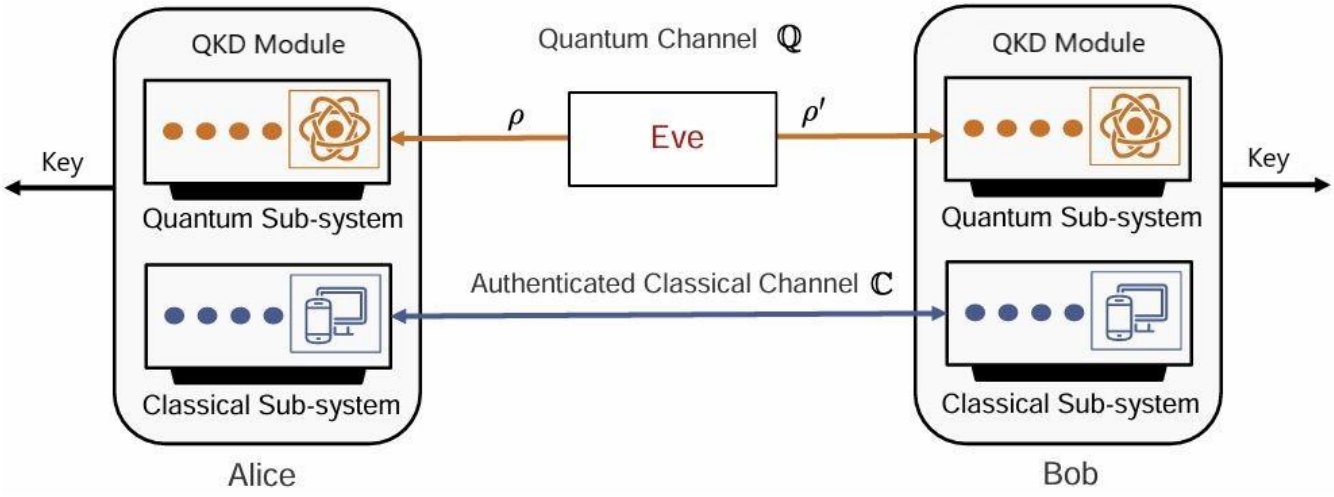


Figura 8. Sistema QKD con canales, módulos, emisor, receptor y espía [34]

El enlace QKD, por su parte, es como se denomina al conjunto lógico que conecta entre si un par de módulos QKD remotos mediante un canal cuántico y otro clásico, permitiéndoles realizar un protocolo QKD, es importante establecerlo de tal manera que la seguridad de las claves simétricas generadas no dependa de los componentes que lo forman [35].

Como primer elemento separado del propio módulo QKD de generación de claves, se encuentra el sistema de gestión de claves KMS (*Key Manager System*). Es un componente crítico encargado de la gestión y el aprovisionamiento de claves a los usuarios de las aplicaciones que van a querer utilizar el sistema QKD. Es donde, una vez generada la clave final, se gestiona el ciclo de vida de las claves cuánticas, descartándolas finalmente cuando es necesario, representado en la Figura 9 .

Las tareas del KMS incluyen almacenar claves en un búfer que actúa como una memoria segura gestionando una cantidad determinada de claves hasta que sean requeridas, para transmitirlas a través de la red QKD y suministrarlas a las aplicaciones según sea requerido siguiendo los pasos establecidos por las recomendaciones para el intercambio de claves utilizando interfaces como ETSI QKD 014 o ETSI QKD 004. Las interfaces se presentan más adelante en el texto y se describirán en detalle en el apartado 3.5.3.

Teóricamente al KMS también se le denominan con un nombre más genérico como, entidad de gestión de claves KME (*Key Management Entity*), refiriéndose a la unidad que gestiona las claves de una red en cooperación con una o más entidades de gestión de claves [35], o incluso simplemente KM [36].

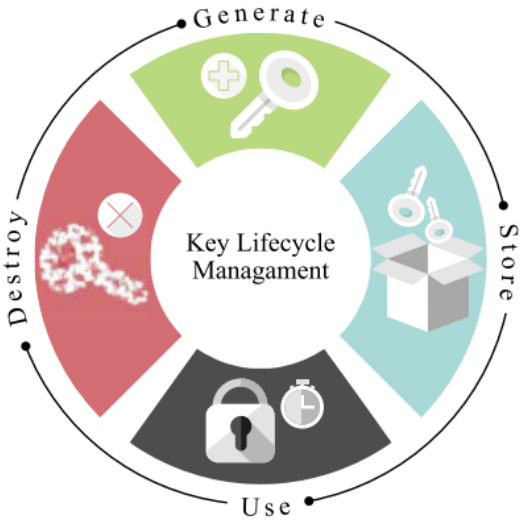


Figura 9. Ciclo de vida de la clave QKD [21]

A continuación, como se ha mencionado, la aplicación QKD es la entidad encargada de consumir las claves QKD generadas por el módulo QKD y dispuestas por el sistema

de gestión de claves, para aplicarlas a su fuente de datos a proteger. Para ello el sistema QKD necesita incorporar un algoritmo criptográfico, que es un procedimiento computacional bien definido utilizado para el cifrado de datos de la aplicación, tomando como entradas las claves criptográficas generadas por QKD y los datos originales de la aplicación, para producir una salida que cumpla con el objetivo de mantener los datos seguros [37].

Asimismo, se denomina entidad de aplicación segura SAE (*Secure Application Entity*), al módulo que solicita una o más claves a una entidad de gestión de claves (KME) mediante APIs (*Application Programming Interface*) de suministro de clave, para una o más aplicaciones que se ejecutan en cooperación con una o más entidades de aplicación seguras [37], buscando establecer una comunicación segura gracias a las claves derivadas del sistema QKD. Es el componente que toma los datos de la aplicación junto con la clave dispuesta para realizar un proceso de cifrado, que generalmente se refleja integrado al módulo de aplicación QKD final como puede verse en la Figura 10 [33].

El siguiente esquema muestra de manera general cómo funciona un sistema de distribución cuántica de claves (QKD) integrado con cifrado clásico para que se comuniquen dos aplicaciones. Se emplean módulos QKD para generar claves secretas, que luego son cifradas para proteger los mensajes de la aplicación a través de un canal público [33].

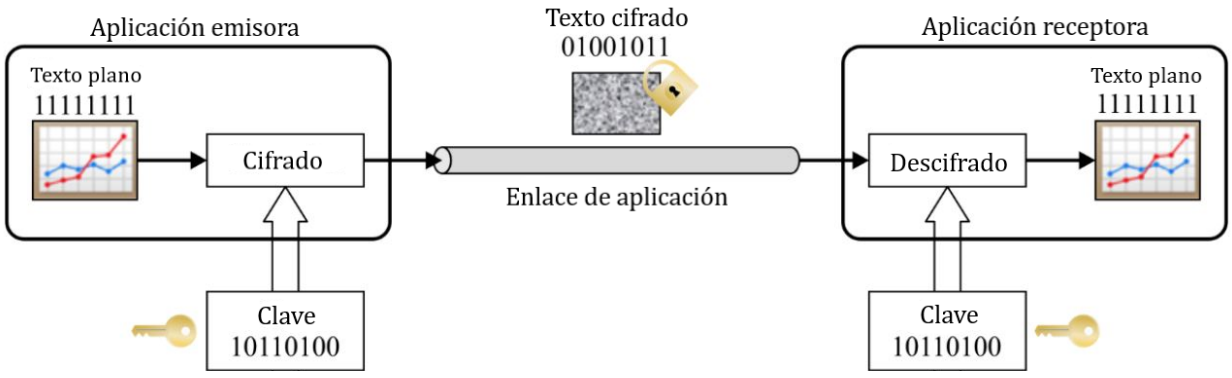


Figura 10. Proceso de cifrado y descifrado por aplicación

Por otra parte, el concepto de interfaz de aplicación QKD, se refiere a la conexión lógica entre un gestor de claves QKD (KMS) y una o varias entidades de aplicación (SAEs), representados como usuarios finales ("*End-User Terminal*") en la Figura 11, donde las dos interfaces reconocidas como ETSI 004 y ETSI 014, definen la forma de comunicación entre dichos elementos mediante distintos atributos que son relevantes desde el punto de vista de la red, utilizando protocolos seguros como

HTTPS para asegurar la entrega de la clave durante la comunicación. Se tratarán en detalle en el apartado 3.5.3.

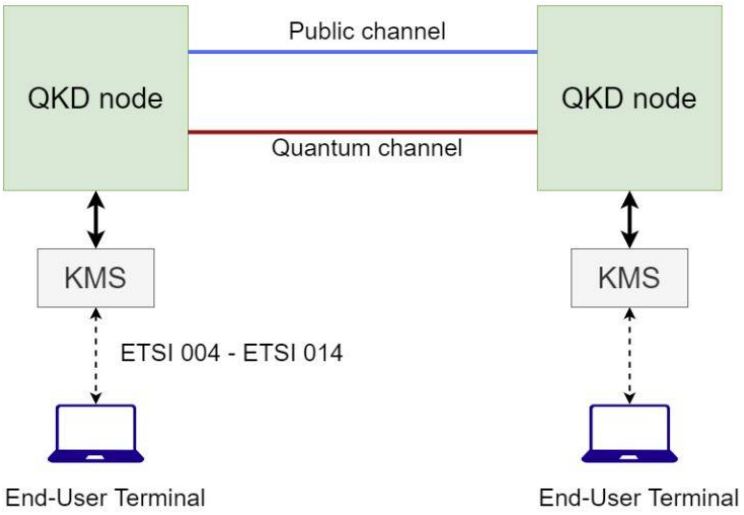


Figura 11. Relación de interfaz, KMS y aplicación [38]

3.5. Arquitectura de red QKD

Las redes de telecomunicaciones convencionales se basan principalmente en la transmisión de datos binarios a través de canales como fibra óptica, cables de cobre o radioenlaces. Su diseño se complementa con el uso de repetidores y amplificadores que mantener la señal en intervalos regulares. Esto permite que los datos viajen grandes distancias controlando problemas como el ruido y atenuación de la señal. Un enlace tradicional es robusto y puede extenderse globalmente gracias a esta infraestructura, creando redes complejas y escalables.

Las propiedades de la transmisión cuántica impiden el uso de amplificadores clásicos, lo que en parte limita el alcance de la tecnología [21]. Esto se debe a que el proceso del amplificador, al intentar amplificar los fotones, destruye la estructura del estado cuántico de estas partículas, perdiendo la información y violando la condición de mantener la seguridad fundamental.

En esta sección se presentan los conceptos y estructuras que constituyen una red QKD para solventar este problema a la hora de distribuir las claves, sus componentes, técnica de retransmisión, estandarización recomendada para su organización interna y las capas de arquitectura de red correspondientes.

3.5.1. Nodos de confianza en QKDN

La red QKD, o también denominada QKDN ("*QKD network*"), es el conjunto de elementos que cumplen con la funcionalidad de distribuir claves secretas simétricas

incondicionalmente seguras a cualquier par de usuarios que accedan a la red de forma legítima [8].

Según su definición [37], se define la estructura de la red QKD como un sistema compuesto por dos o más nodos de confianza. Lo que crea una red QKD basada en nodos de confianza que seguirá un proceso denominado "trusted relaying" o de retransmisión fiable [8], que se detallará en el siguiente apartado.

Se define un nodo de confianza TN ("*Trusted Node*"), o nodo QKD, al módulo que contiene los distintos equipos seguros, incluyendo una o más entidades de gestión de claves KMEs y uno o más módulos QKD, situados dentro de un perímetro de seguridad [18], como podemos ver en la Figura 12 en azul y en la Figura 13 representado mediante un círculo verde. Dispuestos de tal manera que establezcan una o más conexiones a otros TNs a través de enlaces QKD.

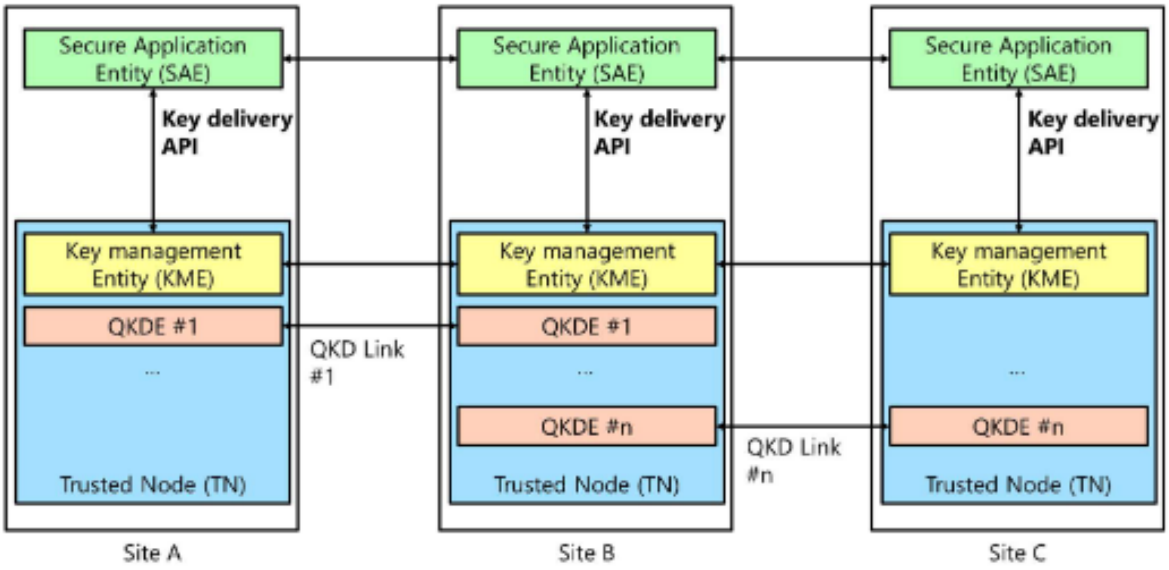


Figura 12. Esquema conceptual de red QKD con nodos de confianza (TN) [18]

Por otra parte, el segundo elemento que conecta dos nodos de confianza, son los enlaces KM, que son las conexiones entre las distintas entidades de gestión de clave. Para ello el KME puede subdividirse en dos apartados denominados KMA ("*Key management agent"*) y KSA ("*Key supply agent"*) con enlaces dedicados interconectando cada función o pueden agruparse en un único enlace KM [36]. Su objetivo es conseguir que los KME pueda realizar sus funciones de sincronización de clave, verificación de integridad y las tareas de retransmisión de claves que se abordan en el siguiente apartado. El desglose específico de los enlaces KM en enlaces KMA y KSA, puede verse representado al final del capítulo en la Figura 24 [19].

Sin embargo, como también se puede ver en la Figura 13, entre dos nodos de confianza consecutivos, se pueden integrar otros mecanismos de red que aporten flexibilidad y alcance de la red.

Por un lado, los "*Optical switch/splitter*", que permiten redirigir o dividir dinámicamente el tráfico cuántico entre distintos módulos, facilitando el establecimiento de claves bajo demanda entre múltiples usuarios en redes multipunto o metropolitanas. En segundo lugar, los "*Quantum relay point*" basados en protocolos como MDI-QKD (*Measurement-Device-Independent QKD*) [39] o TF-QKD (*Twin-Field QKD*)[40], funcionan como estaciones de medida intermedia que no requieren ser parte de la red de confianza ya que no acceden a la clave en sí misma y la seguridad se garantiza por el propio protocolo. Este nodo recibe fotones de dos nodos de la red obteniendo información sobre la correlación entre los fotones y en base al resultado el par de nodos genera clave, lo que permite extender la distancia operativa de la red sin introducir vulnerabilidades.

En conjunto, la combinación de estas técnicas ofrece la posibilidad de una red más escalable y segura, complementando el papel de los nodos de confianza utilizando funciones de enrutamiento óptico y puntos de retransmisión cuánticos con protocolos de nueva generación [19], [33].

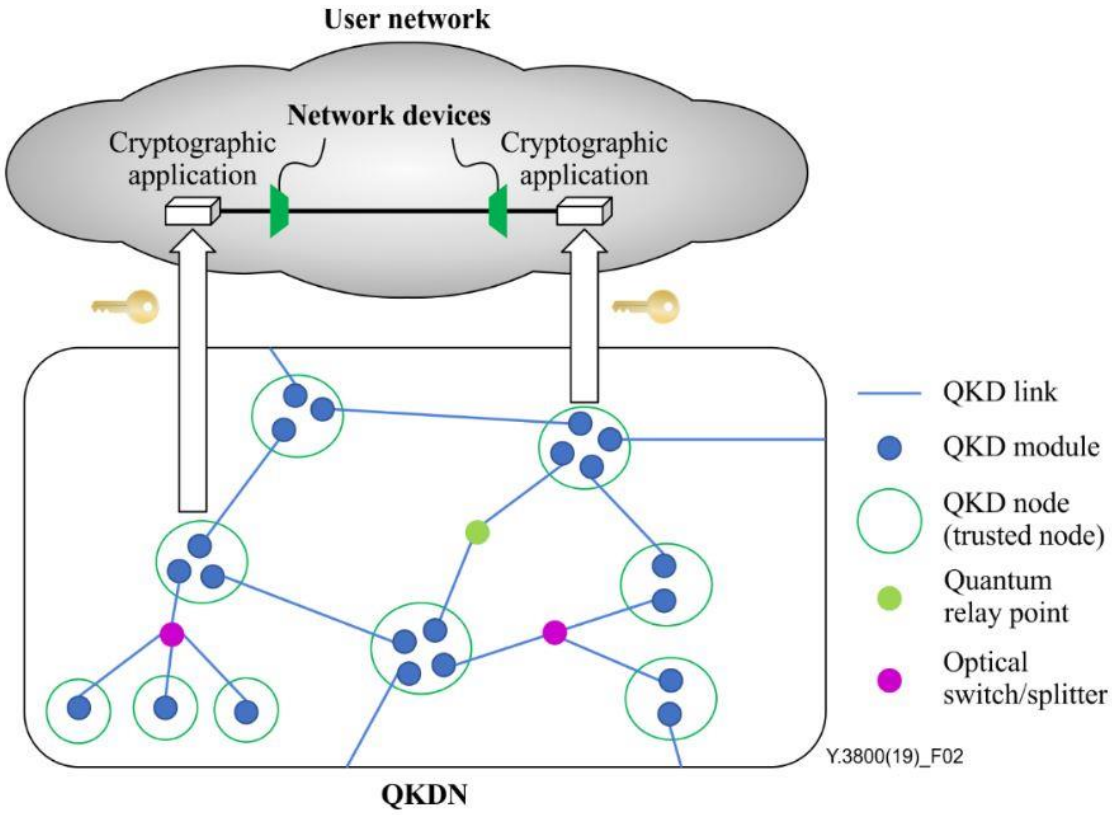


Figura 13. Esquema de red QKD [33]

3.5.2. Redes de enlaces punto a punto mediante retransmisión fiable de clave

Para llevar a cabo la comunicación a través de varios nodos, las redes QKD se basan en el supuesto de que todos los nodos son fiables cuando la comunicación se realiza de forma "*hop-by-hop*" [1], es decir, salto a salto, en un proceso de retransmisión fiable o "*trusted relaying*".

El modelo de implementación de red más sencillo solo requiere una conexión física directa entre dos nodos, en lo que sería una tecnología punto a punto [21]. De manera que, si se quiere escalar el proceso y establecer una red mediante conexiones salto a salto, estas pueden considerarse como la conexión en serie de sistemas QKD independientes punto a punto. En estas redes, el material de clave seguro se transporta de un extremo a otro según sea necesario a través de los nodos de confianza intermedios introducidos en el apartado interior, actuando como repetidores de confianza en un proceso denominado "*key relay*" o retransmisión de clave.

Para que una red QKD funcione, los módulos QKD adyacentes deben conectarse a través de los enlaces lógicos QKD, definidos en el apartado 3.4, de tal manera que el canal cuántico mantenga conexión directa entre módulos QKD, para que los fotones viajen directamente sin problemas entre dos nodos, pero el canal público, se pueda formar mediante una conexión común con un número arbitrario de dispositivos intermedios que verifiquen y procesen los datos intercambiados.

La técnica principal en la que se sustenta el proceso de retransmisión fiable se denomina "*Store & Forward*". Traducido como almacenamiento y reenvío, consiste en que, para cada enlace de la red, la información recibida en un nodo intermedio se descifra y se vuelve a cifrar antes de ser reenviada, asegurando así la continuidad del proceso a lo largo de la cadena de nodos [8].

El proceso de retransmisión considera que cuando se quiere transmitir un paquete de clave entre dos nodos distantes a través de uno o más nodos intermedios, estos actúan como repetidores de confianza, teniendo en cuenta que cada enlace de la ruta requiere una clave independiente propia. Para ello, en cada nodo intermedio procesa el paquete recibido se descifra, se verifica mediante su etiqueta de autenticación y, a continuación, se vuelve a cifrar utilizando la clave compartida con el siguiente nodo del enlace. Siguiendo este proceso en cadena para cada nodo necesario en la transmisión del mensaje hasta el destino final, donde se recupera el mensaje deseado [1].

Este proceso se puede ver de forma conceptual en la Figura 14, donde cada color representa las diferentes asociaciones de claves entre pares de nodos fiables.

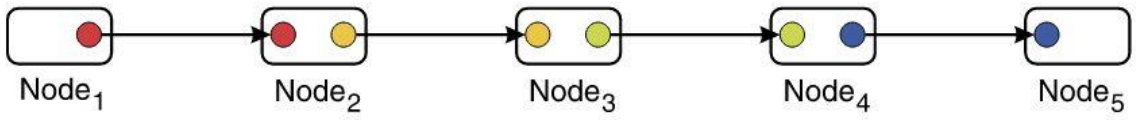


Figura 14. Retransmisión visual de clave sobre red QKD punto a punto con TN [8]

Otra representación más técnica del proceso se puede ver en la Figura 15, donde en este caso se representa la retransmisión de clave implementando el cifrado OTP con la operación XOR entre la clave del enlace y el paquete a retransmitir 'n'. De este modo, gracias a un protocolo QKD previo, el nodo A tiene compartida una clave con B, denominada 'AB', con la que procede a cifrar el paquete para enviárselo a B a través del canal clásico. Entonces B, como conoce la clave AB, recupera 'n' mediante el descifrado, aplicando de nuevo XOR, y así sucesivamente entre los nodos conectados hasta, en este caso, llegar al nodo D que recupera el paquete 'n' original.

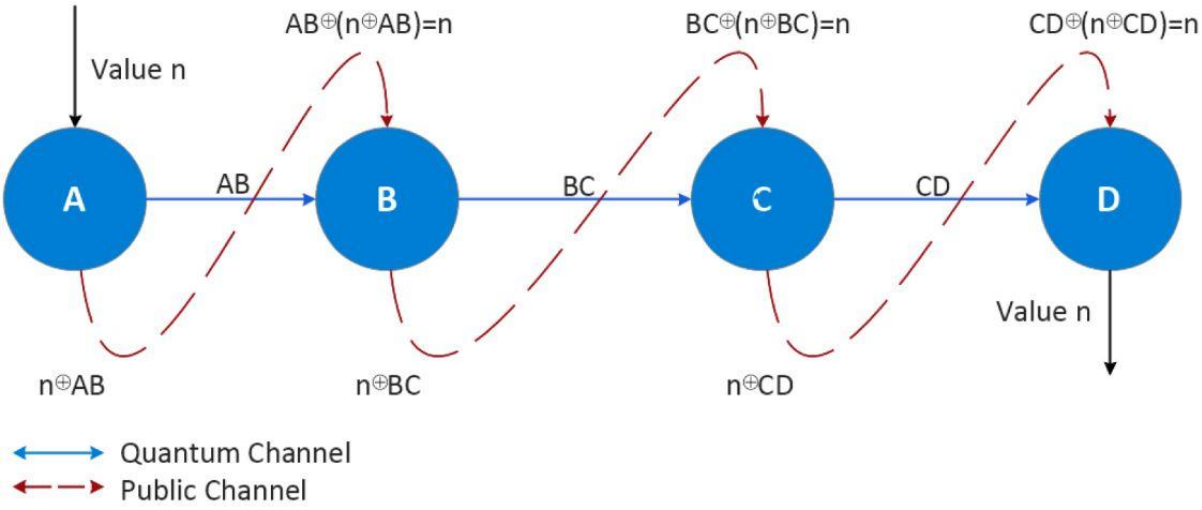


Figura 15. Retransmisión de clave sobre red QKD punto a punto con TN y OTP [1]

Su principal inconveniente es la posible congestión de un nodo, que pueda provocar que el comportamiento de la comunicación se ralentice, tardando en principio más tiempo en transferir los datos por la ruta hasta el destino [25].

Otra debilidad de este enfoque es que cada nodo de retransmisión en la ruta debe tener acceso a la clave global, lo que aumenta la probabilidad de ataque y exige confianza total en todos los intermediarios [21].

Para mitigar esta vulnerabilidad, existen otras dos opciones de distribución de claves globales propuestas por ITU-T en [36].

- Distribución de clave empleando XOR uniforme en el nodo destino

  Este proceso mantiene las claves parciales ocultas en los nodos intermedios, combinándolas con una operación XOR solo en el destino final, lo que reduce la exposición de la clave en la red y limita la necesidad de confiar en cada nodo intermedio.

- Distribución de clave empleando XOR centralizada

  En este caso, un nodo específico y altamente seguro centraliza la recolección y combinación de todas las claves parciales. Este proceso simplifica la operación, pero introduce un punto único de fallo, lo que también podría comprometer toda la red si ese nodo es atacado.

Una vez analizado este punto, cabe destacar que el conjunto de este proceso es iniciado por la aplicación, solicitando clave segura para sus datos, e iniciando con ello el proceso de retransmisión por la red entre los nodos de confianza. Para garantizar la interoperabilidad de las redes QKD con las aplicaciones QKD, es importante contar con especificaciones que estandaricen la entrega de claves entre ellos, permitiendo además la idea de interoperabilidad entre distintos fabricantes como se presenta a continuación.

3.5.3. Estandarización ETSI para sistemas QKD

Uno de los principales retos para llevar la implantación industrial de redes QKD a gran escala, es una estandarización que garantice compatibilidad de los componentes que se fabriquen a nivel global. Los principales organismos de estandarización trabajan en las normas de estandarización de QKD son por ejemplo ETSI, ITU-T, ISO/IEC, IETF, IEEE y CSA [19].

La implementación de QKD en dispositivos reales está sujeta a detalles que pueden llevar a imperfecciones que podrían llegar a reducir la seguridad del sistema. Por lo tanto, es importante que se produzca una certificación de dispositivos acorde con el nivel de seguridad previsto en el estándar.

En concreto, el Instituto Europeo de Normas de Telecomunicaciones (ETSI), estableció un Grupo de Especificaciones Industriales (ISG) sobre QKD, denominado ETSI SG-QKD, desarrollando una serie de especificaciones e informes grupales como los siguientes, que se encuentran actualmente activos y en proceso de ser actualizados.

- **ETSI GS QKD 007,** [37]**:**

  Tiene como finalidad establecer un marco común de definiciones y abreviaturas relacionadas con QKD. Surge porque la distribución cuántica de claves introduce

conceptos y tecnologías que, si bien en parte provienen de la física cuántica y la criptografía clásica, adquieren un significado más específico cuando se aplican en el contexto de las telecomunicaciones. Para evitar confusiones y asegurar que los distintos actores del sector emplean un lenguaje coherente, ETSI trata de reunir en un único lugar la terminología más relevante, funcionando como una referencia que mejora la consistencia en los informes y especificaciones.

En este sentido, tiene el propósito de servir como glosario estandarizado, garantizando que tanto investigadores como fabricantes, operadores y desarrolladores de sistemas de QKD compartan un mismo lenguaje al describir componentes, procesos y tecnologías de manera estándar. Cabe destacar que la versión actual que se consultó necesita actualizarse ya que no se encuentran todos los términos relacionados, como por ejemplo los términos mencionados en el apartado 3.5.1, KMA y KSA.

- **ETSI GS QKD QKD 011,** [41]**:**

  Se centra en establecer un marco de trabajo para que los fabricantes y desarrolladores midan y describan con precisión los componentes ópticos empleados en sistemas de QKD. Incluye en él una detallada recopilación de procedimientos de medición especificados como: frecuencia de reloj, repetición de pulsos ópticos, número medio de fotones, entre otros. Describiendo tanto el equipamiento necesario como los entornos operacionales y los métodos de cálculo para obtener los resultados necesarios.

  La finalidad que busca es garantizar que las mediciones sean comparables entre diferentes dispositivos y laboratorios, minimizando incertidumbre, y facilitando la interoperabilidad técnica. El documento busca convertirse en una guía esencial para asegurar que los componentes cuánticos puedan ser evaluados de forma consistente y reproducible, teniendo en cuenta que la versión actualmente publicada del 2016, se encuentra también a la espera de actualizarse. La versión ETSI GS QKD 013 aún sin publicar, pretende extender este aspecto.

A continuación, los documentos ETSI GS QKD 004 y ETSI GS QKD 014, introducidos en el apartado 3.4, son recomendaciones clave en el ámbito de la distribución cuántica de claves. Su función principal es definir cómo componentes de entidad de aplicación segura (SAE) y de entidad de gestión de claves (KME), se comunican e intercambian claves siguiendo su respectiva interfaz para poder utilizar los servicios de la red QKD.

3.5.3.1. Interfaz ETSI GS QKD 004

La especificación ETSI GS QKD 004 [35] define la interfaz de aplicación QKD para la gestión del servicio de claves entre las entidades de gestión de claves (KMEs). Su propósito principal es estandarizar la forma en la que dos aplicaciones seguras (SAEs), situadas en extremos opuestos de la red, puedan establecer y coordinar un flujo de claves cuánticas a través de sus respectivos KMEs.

Se define como una interfaz que puede ser implementada sin necesidad de definir un protocolo específico para la comunicación. Puede efectuarse de forma muy compacta sin bibliotecas complejas mediante lenguaje C estándar, simplificando el análisis de su seguridad. El objetivo de esta API es proporcionar a las aplicaciones un mecanismo estandarizado con reglas y comandos que la aplicación debe seguir para establecer un canal extremo a extremo. Gracias a ello, se podrá solicitar, recibir y gestionar las claves cuánticas de un KME.

Para ello introduce el concepto de sesiones con prestaciones de calidad de servicio QoS ("*Quality of Service*"), donde estos requisitos incluyen, según [25] y [35], por ejemplo:

- Tasa de clave máxima y mínima [bps]: La velocidad a la que la aplicación espera recibir bloques de clave.
- Tamaño de bloque de clave ("*Key chunk size*") [bytes]: Define cuántos bits debe contener cada bloque entregado según requiera la aplicación.
- Duración o tiempo de vida de la sesión TTL ("*Time to Live*") [s]: Define cuánto tiempo debe mantenerse el KSID ("*Key Stream ID*") activo. Este parámetro se describe más adelante.
- Otros parámetros que especifican temporizadores, la fluctuación de la entrega de la clave, el nivel de prioridad o información de metadatos.

Estos parámetros influyen en cómo el KME gestiona sus recursos, ya que, si se solicita una tasa de entrega excesivamente alta, puede que no se pueda satisfacer con el ritmo de generación de clave disponible en el sistema en ese momento. Así, se establece por ejemplo que, si se piden tamaños de bloque mayores a los soportados, la petición será rechazada

Sigue un enfoque orientado a sesiones, en el que las sesiones se identifican mediante el parámetro KSID ("*Key Stream ID*"), un identificador único proporcionado por el gestor de claves QKD a la aplicación, en donde ambos pares utilizarán el mismo valor para hacer referencia a su flujo de clave de aplicación. Cada aplicación puede gestionar varios flujos independientes, por lo que, en teoría, no hay límite en cuanto al número de ellos que puede solicitar una aplicación [25].

Se definen tres funciones API, en primer lugar, "OPEN_CONNECT", que tiene como objetivo principal establecer una sesión de flujo de claves y reservar claves de

acuerdo con las especificaciones de QoS. Con esta función, el SAE pide a su KME que prepare la conexión hacia el SAE remoto. La respuesta de esta solicitud incluye el KSID y el estado de la conexión obtenido, lo que permite que la sesión quede activa y lista para usarse en la transmisión de claves.

Cabe destacar que la documentación de ETSI QKD 004 no describe el funcionamiento interno del servicio de comunicación entre los KME, dejándolo a criterio del desarrollador. Tampoco dicta cómo un KME debe organizar sus bases de datos, manipular buffers de clave o coordinarse internamente con el módulo QKD. Como se puede ver en la Figura 16, ETSI QKD 004 tampoco establece cómo se comunican los identificadores de claves entre los SAE conectados.

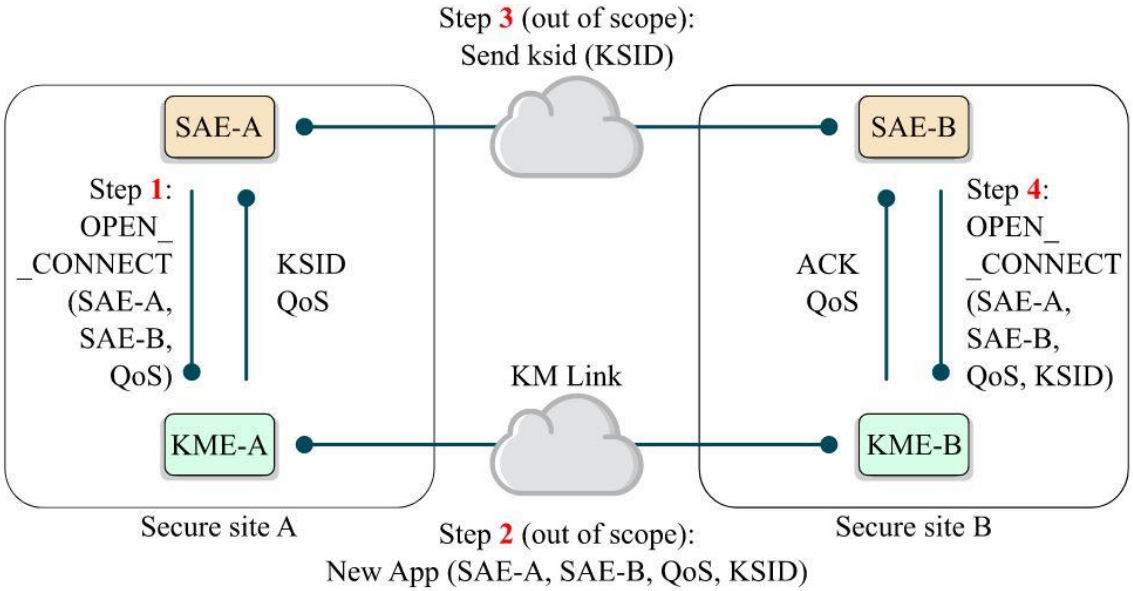


Figura 16. Interfaz de suministro de clave ETSI QKD 004 Establecimiento de sesión [21]

Las otras dos funciones, que no aparecen en la Figura 16, son “GET_KEY” y “CLOSE”. La función “GET_KEY” devuelve la cantidad necesaria de material de clave solicitado para el KSID específico junto con el estatus que informará del éxito de la solicitud. Mientras que la función “CLOSE” permite a los SAE finalizar y terminar las sesiones de flujo de claves, pueden verse en el diagrama de la Figura 17 que se introducirá a continuación.

La API está diseñada para operar en múltiples niveles de gestión de claves, desde enlaces individuales hasta redes completas, por lo que cada módulo QKD tiene su propia unidad de gestión de claves KME y un servidor de claves de nivel superior, que se comunica con otros posibles módulos QKD, dentro del nodo QKD.

La Figura 17, muestra un diagrama de secuencia de la interfaz de suministro de claves ETSI QKD 004, donde se muestran las tres funciones primitivas: “OPEN_CONNECT”,

"GET_KEY" y "CLOSE". Se puede ver como el proceso inicia la conexión con "OPEN_CONNECT", solicitando al KMS que establezca una sesión hacia el destino según parámetros de QoS. El KMS, con el apoyo de un protocolo de encaminamiento y sus buffers QKD, reservan claves y generan un identificador de sesión KSID, que se sincroniza entre ambos extremos. Una vez creada la sesión, las aplicaciones pueden obtener claves extrayendo el material de sus buffers mediante "GET_KEY". Finalmente, se observa como el proceso finaliza con la función "CLOSE" que libera los recursos y cierra la sesión. Cabe destacar que las líneas azules, representan los procesos de la comunicación que no están determinados por la recomendación [35].

Por otra parte, los mensajes de error que se envían a la aplicación lo hacen a través del parámetro de estado, que corresponde con un valor correcto, o un valor de error, que puede ocurrir debido a no haber suficientes claves disponibles, pero también porque la aplicación homóloga aún no está conectada, no hay conexión QKD disponible, el KSID ya está en uso, se superó el tiempo de espera especificado o porque no se pudieron cumplir las especificadas de QoS solicitados entre otros.

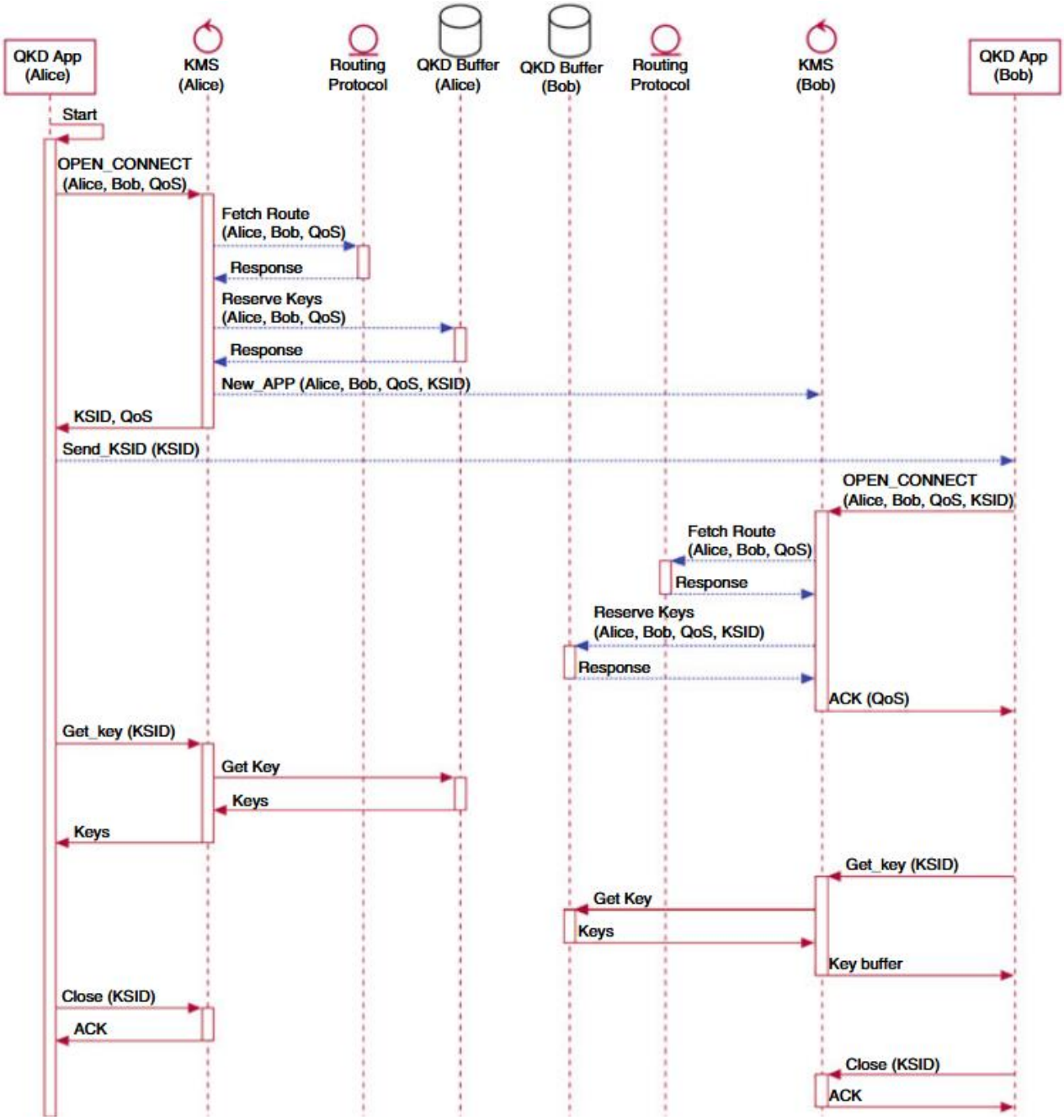


Figura 17. Diagrama de secuencia de la interfaz ETSI QKD 004 [25]

3.5.3.2. Interfaz ETSI GS QKD 014

Esta especificación de interfaz QKD [18], adopta un enfoque más moderno y accesible, ya que presenta una gran aceptación entre los proveedores de equipos cuánticos [21]. Define una API para la gestión de claves entre el KME y las aplicaciones seguras (SAEs), utilizando HTTPS como protocolo de comunicación basado en REST con formato de mensaje JSON, expresando cada operación como una URI junto a parámetros específicos, con el objetivo de facilitar la interoperabilidad entre equipos de distintos proveedores.

Al establecer la conexión, se llevará a cabo una autenticación mutua entre SAE y KME, de manera que a menos que se haya verificado la ID del KME, este rechazará la conexión del SAE. Después de la autenticación mutua, el SAE puede llamar a métodos API en el KME. Cada SAE y cada KME tendrá un ID (SAE ID, KME ID), que será único en la red QKD, pero su formato y asignación no están definidos por la especificación. A cada clave entrega por el proceso, también se le asigna un identificador único universal.

Previo a la gestión de la clave, con un proceso no delimitado por la especificación, se establece la llamada al método "GET_STATUS" para permitir a un SAE origen conocer información necesaria de un KME considerando el estado de la conexión de la red QKD, con información como los diferentes identificadores, el tamaño (mínimo, máximo o por defecto) de clave que un KME puede entregar, las claves disponibles que puede solicitar para otro SAE destino, el máximo número de claves que se pueden solicitar por petición o las SAE de destino adicionales permitidas por el KME. Un ejemplo sería el siguiente formato obtenido del propio documento [18]:

```
" {
    "source_KME_ID": "AAAABBBBCCCCDDDD",
    "target_KME_ID": "EEEEFFFFGGGGHHHH",
    "master_SAE_ID": "IIIIJJJJKKKKLLLL",
    "slave_SAE_ID": "MMMMNNNNOOOOPPPP",
    "key_size": 352,
    "stored_key_count": 25000,
    "max_key_count": 100000,
    "max_key_per_request": 128,
    "max_key_size": 1024,
    "min_key_size": 64,
    "max_SAE_ID_count": 0
} "
```

Utiliza el paradigma maestro-esclavo, de tal manera que un SAE maestro de origen puede iniciar una comunicación segura con SAE esclavo de destino dentro de un entorno protegido y autenticado, siguiendo los siguientes pasos, como muestra la Figura 18.

1. El SAE origen llama al método API "GET_KEY" de petición de claves para obtener claves de su KME, puede iniciar el ID de la aplicación de destino, el tamaño y número de claves solicitadas junto con más opciones si se quieren especificar o, si no, serán asignadas en su forma establecida por defecto. La respuesta corresponde con el material de clave solicitado junto con los ID de clave asociados "*Key ID*" que serán iguales a los del KME de destino y el número solicitado de claves proporcionadas se elimina del conjunto de claves almacenado en KME.

2. El SAE origen notifica a SAE de destino los ID de clave obtenidos. Esta comunicación entre SAEs queda fuera de la especificación.

3. Por último, el SAE de destino utiliza el método API de entrega de claves "GET_KEY_WITH_KEY_IDS" junto con el ID del SAE origen y la información de los ID de clave establecidos, para conseguir el mismo conjunto de claves de su propio KME, el cual, mediante un paquete de claves "*Key container*", devuelve el material de clave idéntico como se estableció en el KME origen donde empezó el proceso y también elimina el correspondiente conjunto de claves de su almacenamiento.

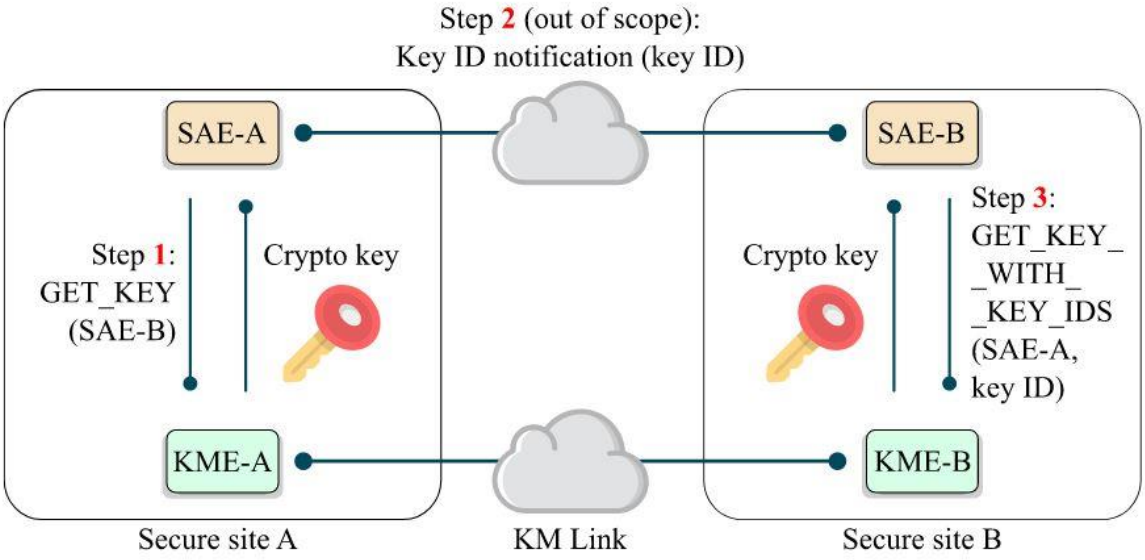


Figura 18. Interfaz de suministro de clave ETSI QKD 014 [21]

Se puede describir este proceso al igual que la interfaz 004, mediante la representación de un diagrama de secuencia de los tres métodos principales que describen el proceso de comunicación entre del SAE con el KME mediante "GET_STATUS", "GET_KEY" y "GET_KEY_WITH_KEY_IDS", se puede ver en el diagrama de la Figura 19, donde las líneas en azul representan los procesos que no establece la recomendación.

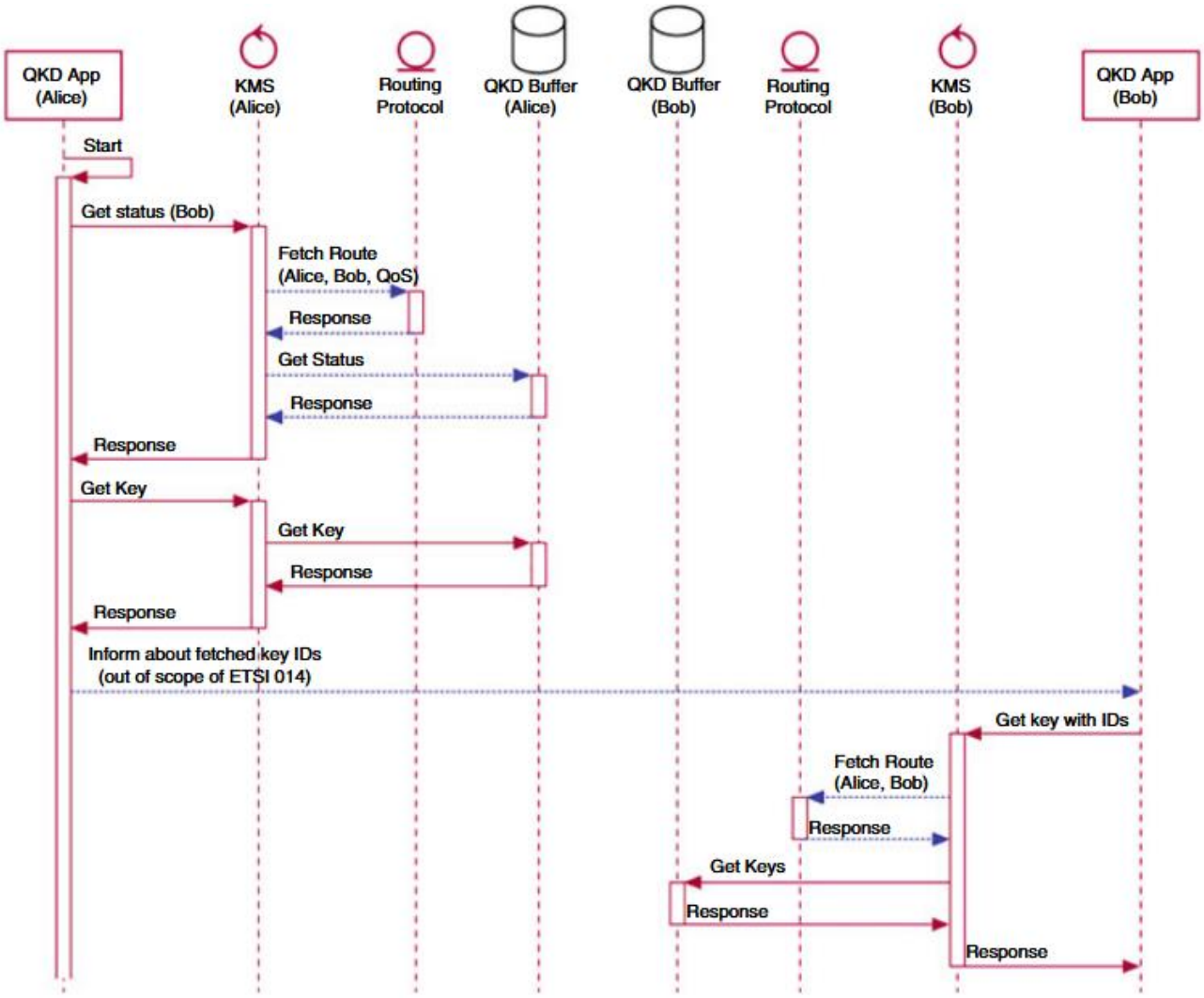


Figura 19. Diagrama de secuencia de ETSI QKD 014 [25]

3.5.4. Estructura de capas de arquitectura de red QKD

Los apartados anteriores han mostrado cómo los diversos componentes de una red QKD están interconectados y se agrupan en función de las etapas que conforman el ciclo de vida completo de las claves cuánticas desde su creación y petición inicial, hasta su retransmisión y consumo.

Dividiendo las dos funcionalidades principales, tenemos, por un lado, a la red de usuarios finales, formada por los usuarios que hacen uso de las claves en sus procesos de comunicación y, por otro lado, al conjunto de la red QKDN, que se encarga de satisfacer la petición de clave a través de sus procesos internos llevados a cabo por sus componentes mediante protocolos QKD, procesos de retransmisión e interfaces de aplicación, vistos en los apartados 3.3, 3.5.2 y 3.5.3 respectivamente.

Para facilitar la visualización del proceso de comunicación, las redes QKD se definen a menudo mediante divisiones de capas. Según [19], existen diferentes interpretaciones de la estructura dependiendo el nivel de definición que se quiera mostrar, puede interpretarse como una arquitectura dividida en 3, 4, 5 o incluso 6 capas. En la Figura 21, podemos ver cómo se representa la versión ofrecida por la recomendación de ITU-T [33], y en la Figura 20, la versión compacta y visual que ofrece [19] o [42] con únicamente 3 capas.

El modelo de tres capas de [19], corresponde con las siguientes subdivisiones con sus correspondientes funciones:

- **Capa de infraestructura**

  Reúne los dispositivos físicos de QKD en los denominados nodos QKD, que representan los nodos de confianza vistos en el apartado 3.5.1, interconectados mediante enlaces QKD y enlaces KM. Cada par de nodos genera y almacena sus claves secretas simétricas, junto con detalles de la clave como su identificador, tamaño, tasa de generación, tipo de clave, identificador de dispositivo y marca temporal, además de parámetros de enlace QKD como longitud y tipo.

- **Capa de control y gestión**

  Consta de un módulo controlador ("*QKD Network Controller*") y otro gestor de red ("*QKD Network Manager*"), el primero se encarga de activar, calibrar y mantener los nodos QKD de la capa inferior, además, según [33], de funciones de vigilancia como control de enrutamiento para la retransmisión de clave o control de autenticación, autorización y QoS. El segundo, supervisa y gestiona la red en su conjunto, recopilando parámetros de estado de los nodos QKD y sus enlaces, de manera que cada cierto tiempo extrae parámetros de claves y enlaces para registrarlos en bases de datos y asegurar un control del sistema actualizado. Las claves secretas nunca son accesibles para ninguno de estos elementos, lo que preserva la seguridad.

- **Capa de aplicación**

  Situado sobre las anteriores capas, se encuentran las aplicaciones criptográficas QKD interconectadas mediate sus propios enlaces de aplicación. Los usuarios solicitan claves con características específicas de tamaño, tasa o periodo de renovación. El gestor valida la disponibilidad de clave y ordena al controlador la entrega en el formato requerido. Una vez recibidas, las claves pasan a ser responsabilidad de la aplicación, para ser utilizadas en procesos de cifrado. Si el gestor no valida disponibilidad, la aplicación debe esperar a la regeneración de claves, por lo que la capacidad de la red depende del equilibrio entre la generación de claves y las demandas de los usuarios.

En este modelo, también se aprecia en la Figura 20, como las distintas capas se comunican de manera efectiva entre ellas gracias a las interfaces habilitadas.

- **La interfaz de gestión:** Conecta al gestor QKD con nodos, al controlador y a las aplicaciones, permitiendo monitorizar el estado de sus dispositivos.
- **La interfaz de control:** Enlaza al controlador de red QKD, con los nodos para configurar las conexiones, el enrutamiento y QoS.
- **La interfaz de aplicación:** Coordina la petición y entrega de claves secretas desde los nodos a las aplicaciones, definida en ETSI GS QKD 004 y 014 como hemos visto en el apartado 3.5.3.

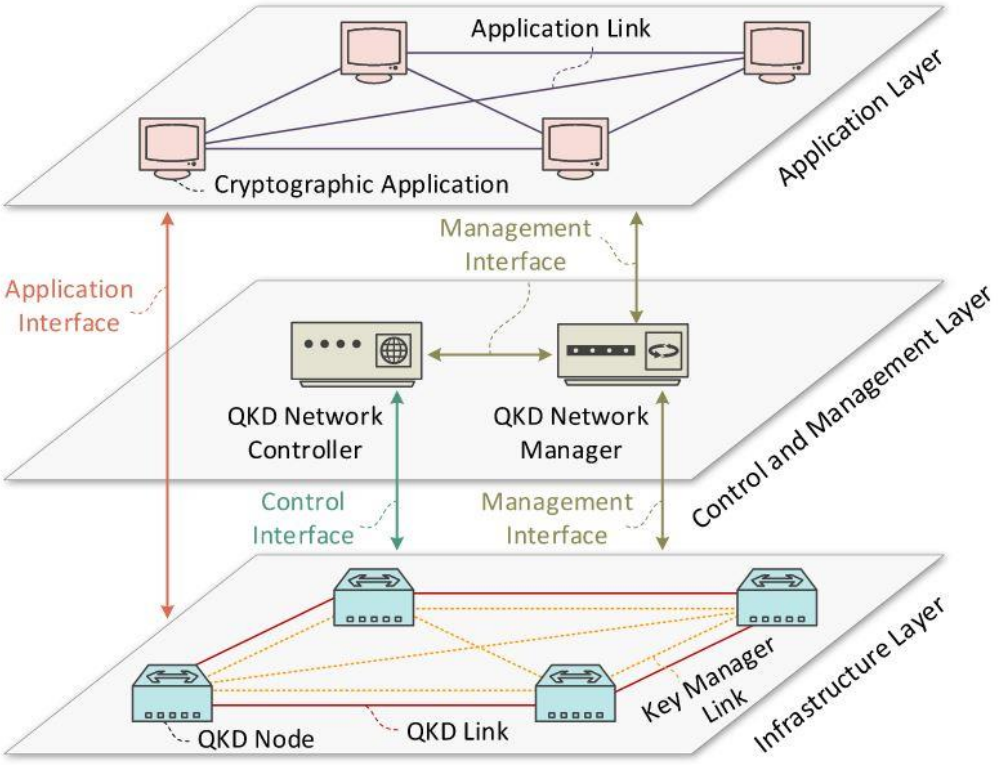


Figura 20. Modelo de red QKD con 3 capas [19]

La diferencia con la visión ofrecida por el modelo de 5 capas se encuentra en,

- Subdividir la capa de infraestructura en dos partes, una capa cuántica, con los módulos y enlaces QKD, y una capa de gestión de clave, con los KME. Lo que sería el equivalente a subdividir la lógica del nodo confianza.
- La capa de control y gestión, se dividen cada una por separado. Donde el controlador QKD en la capa de control de red, se puede implementar de manera centralizada con solo un controlador para toda la red como se puede ver en la Figura 21, o distribuida, con un controlador en cada nodo,

permitiendo una gestión de red flexible [33], como se puede ver en la Figura 22 y Figura 23. De tal manera que el gestor QKD también queda separado, pero mostrándose unido al resto de capas de la red QKD, mostrando como ya se ha mencionado, su función de monitorizar y manejar la propia red QKD.

- Por último, la capa de aplicación queda inalterada, pero bajo el nombre de capa de servicio, sumando finalmente las 5 capas.

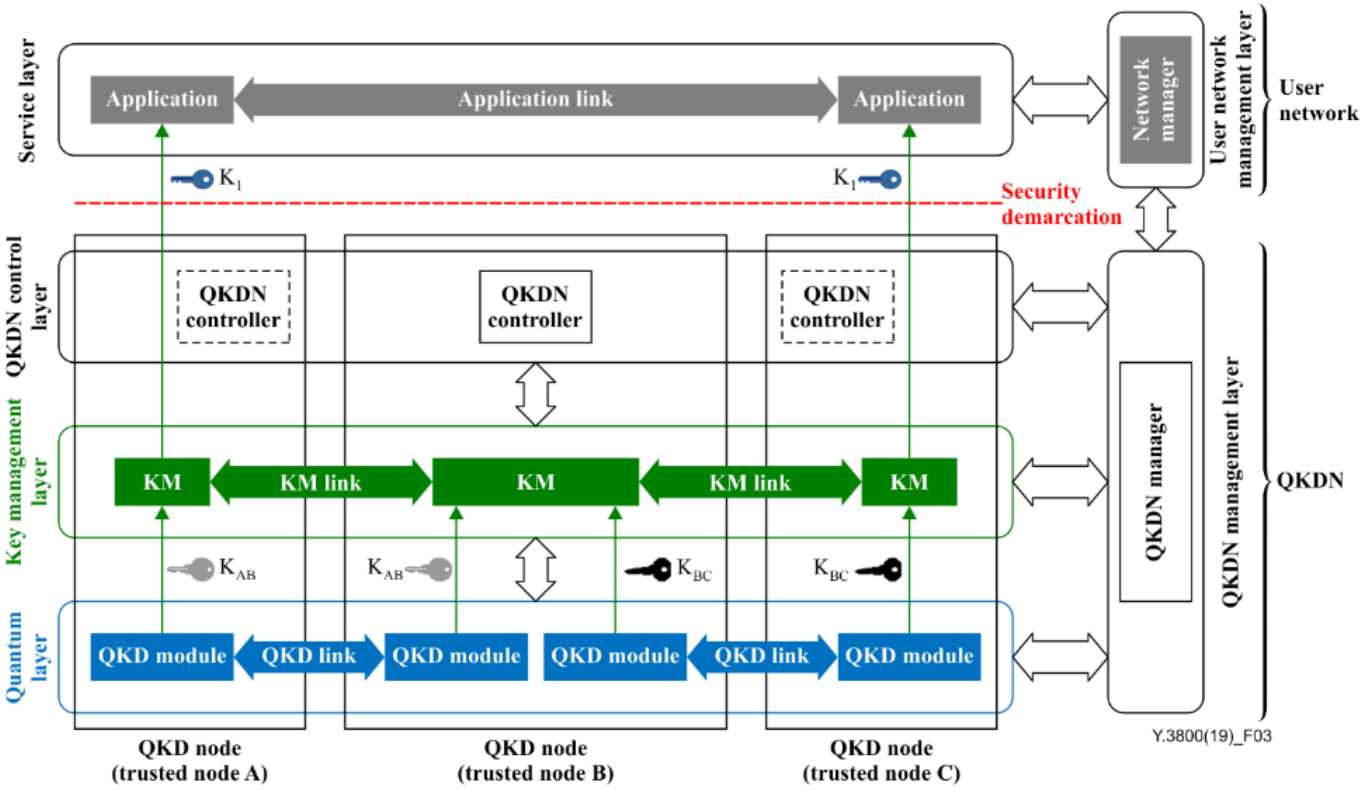


Figura 21. Arquitectura de 5 capas red QKD según recomendación ITU-T [33]

Esta estructura de red dividida en 5 capas recomendada por la ITU-T [33], también es aplicada en las estructuras de la Figura 22, Figura 23 y Figura 24. Se diferencian en matices como que la recomendación también añade un módulo que se comunica con la red QKD, denominado capa de gestión de red del usuario, que se encargan de orquestar los recursos en la red del usuario. Este módulo se omite en ocasiones ya que no influye directamente en el segmento de red QKD [21]. Otra diferencia que se aprecia en estos ejemplos de arquitectura de red, es la delimitación de la estructura de un nodo QKD. Los nodos de confianza, definidos en el apartado 3.5.1, solo incluyen las capas de aplicación y de gestión. No obstante, en la recomendación [33], comprende las capas cuántica, de gestión y de control. También por su parte, la implementación de la Figura 22, donde podemos ver que el nodo agrupa desde los dispositivos QKD hasta la aplicación, dándole una connotación más general al conjunto de componentes.

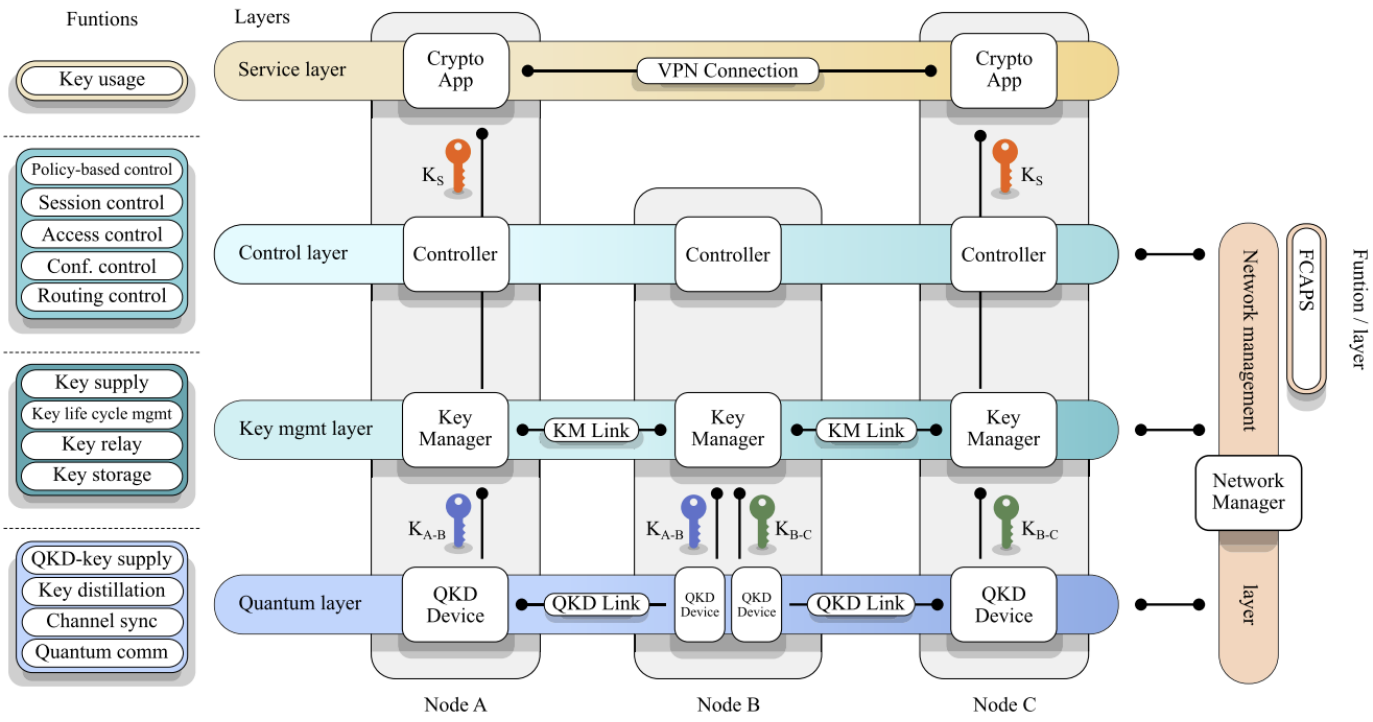


Figura 22. Arquitectura de 5 capas red QKD de estructura y funciones [21]

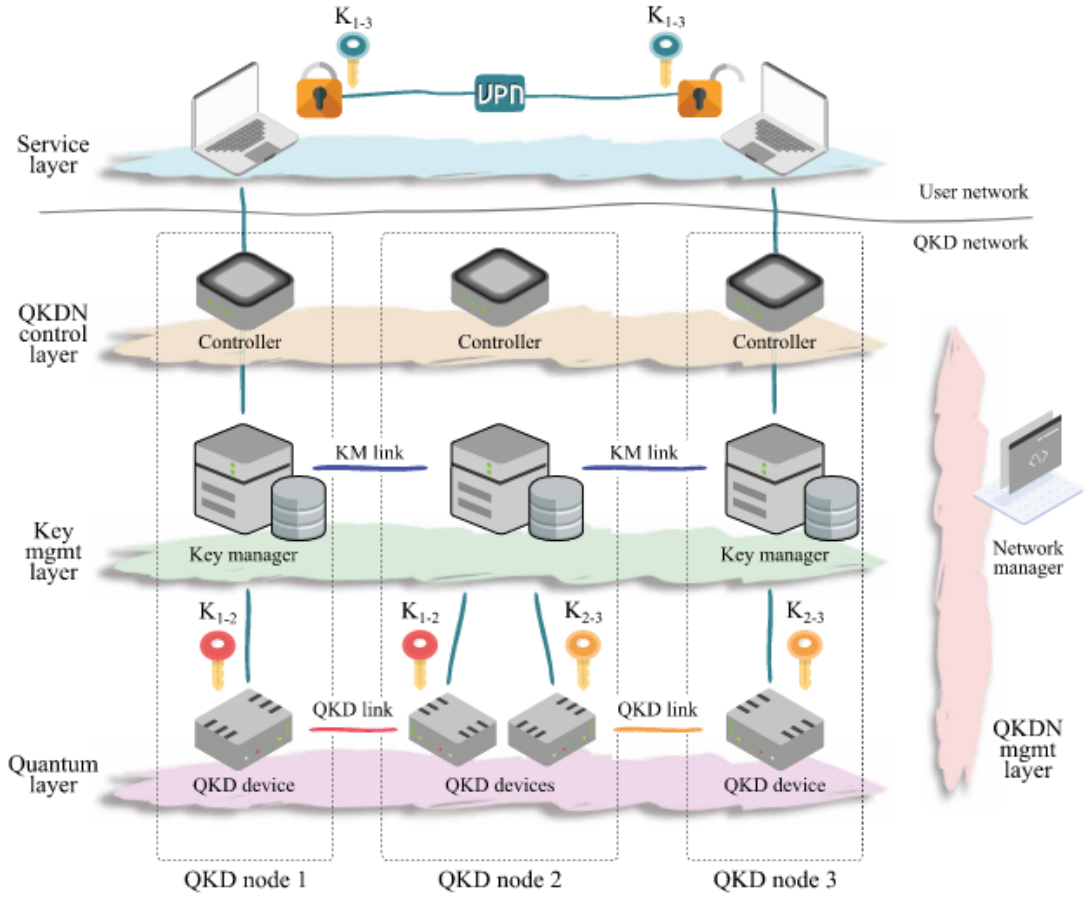


Figura 23. Arquitectura de 5 capas red QKD implementación realista [43]

Finalmente, en la Figura 24 representa de nuevo las capas de arquitectura QKD en pleno proceso de retransmisión de clave a través de los nodos seguros. Podemos ver como las aplicaciones criptográficas piden clave "*key request*". De forma transparente podemos interpretar cómo, en el interior del módulo QKD cada par de módulos comparten un set de claves iguales, que proceden a compartir a través del módulo KM en este caso subdividido en KMA y KSA, siguiendo el modelo de retransmisión expuesto en el apartado 3.5.2 de "*key relay*", para acabar suministrando a las aplicaciones el mismo conjunto de clave segura.

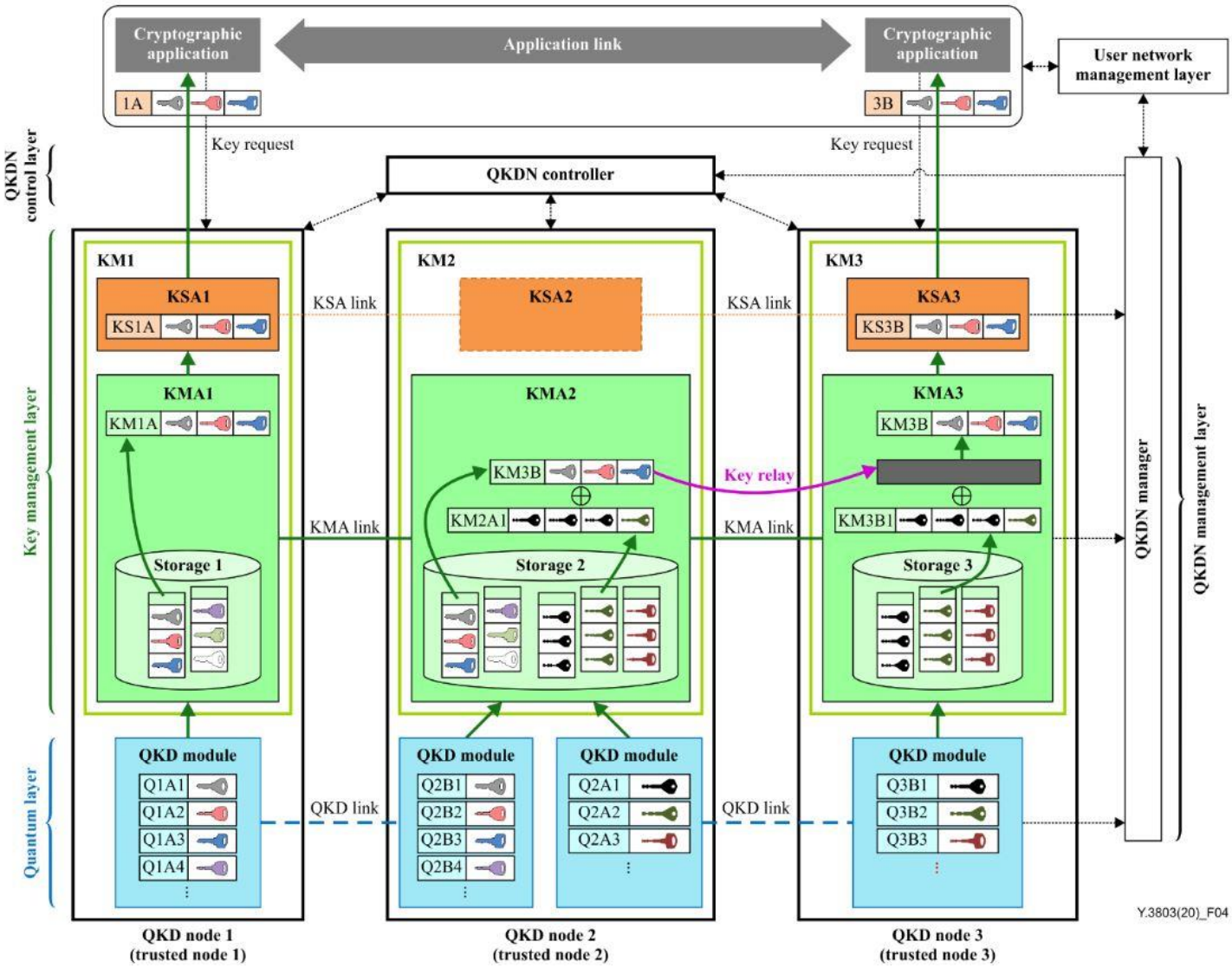


Figura 24. Detalle del proceso de distribución de clave con retransmisión de clave entre TN en red QKD [36]

Capítulo 4

4. Herramienta de simulación QKDNetSim

El desarrollo de las tecnologías seguras para la comunicación cuántica es un proceso complejo y costoso. Por ello, es crucial contar con herramientas como los simuladores para acelerar su avance, permitiendo a los investigadores probar nuevos protocolos, arquitecturas de red o configuraciones, sin la necesidad de construir sistemas físicos.

Tras haber presentado los fundamentos teóricos de las redes de QKD, el siguiente paso consiste en analizar cómo dichos conceptos pueden aplicarse en escenarios prácticos. Para ello se utiliza como referencia central en este estudio, el trabajo de Dervisevic, Voznak y Mehicel [44], que presenta una de las aportaciones más recientes y relevantes en este ámbito. Este artículo introduce un simulador de redes QKD, denominado QKDNetSim, que permite trasladar la teoría de redes QKD a un entorno controlado de experimentación, facilitando la evaluación de aspectos como la gestión de claves, el rendimiento de los enlaces y la integración con sistemas de comunicación clásicos.

En esta línea, el presente capítulo divide la exposición en cuatro partes. En primer lugar, se describe el origen y las características del simulador elegido. En segundo lugar, se describen la arquitectura modelada por el simulador y sus características con el objetivo de ofrecer un contexto de sus adaptaciones con respecto al marco teórico. Seguidamente, se presentan los posibles parámetros de entrada parametrizados por el simulador para distinguir las posibilidades que se encuentran disponibles. Por último, se desarrolla un caso de uso práctico en el que se exponen de manera detallada el proceso de simulación y cómo se presentan los resultados obtenidos a partir de un sistema propuesto.

De este modo, se establece un puente entre la base teórica ya expuesta y la experimentación práctica, proporcionando una visión completa tanto de las capacidades como de las limitaciones del simulador.

4.1. Origen, características y marco de desarrollo

La herramienta de simulación QKDNetSim (*QKD Network Simulation Module*) surge en 2017 [45], como el primer entorno de simulación centrado en redes QKD. Los simuladores relacionados con QKD hasta el momento, no abordan la problemática de

las redes QKD a gran escala, para examinar aspectos como la gestión de clave, la interoperabilidad y el control de red.

El proyecto se enmarca dentro del marco europeo de investigación en seguridad cuántica desarrollado en el ecosistema OpenQKD [46] y financiado por la Comisión Europea con el objetivo de desplegar y evaluar bancos de pruebas cuánticos distribuidos.

Sobre esta base, se construye en NS-3 (*Network Simulator 3*), un entorno ampliamente consolidado para la simulación de redes clásicas, aprovechando sus capacidades de simular y emular un tráfico real. Con las instrucciones adecuadas ofrece la funcionalidad de simular partes específicas de la tecnología QKD. Para comprender mejor el entorno de emulación que ofrece NS-3 y cómo facilita la interacción entre redes simuladas y reales, se puede consultar su página oficial [47].

La versión disponible en este momento se denomina QKDNetSim v2.0 y pueden encontrarse todos los enlaces relacionados con ella en su página oficial [48]. Se implementa como un módulo de simulación integrado en NS-3 extendiendo sus funcionalidades al ámbito de las redes de distribución cuántica de claves (QKD). Desde su diseño inicial [45], se puso especial énfasis en el objetivo de garantizar:

- Precisión en los resultados de simulación.
- Extensibilidad para incorporar nuevos escenarios y protocolos.
- Facilidad de uso para la comunidad investigadora.
- Disponibilidad abierta, favoreciendo la adopción y colaboración.

Al utilizar QKDNetSim, se puede obtener información más profunda sobre el comportamiento de múltiples técnicas de QKD en una variedad de escenarios, facilitando así la investigación y el desarrollo en este campo.

4.2. Arquitectura modelada por el simulador

Las redes de distribución cuántica de claves (QKDN), como hemos visto en el apartado 3.5.4, se pueden dividir en 5 capas. En concreto, el simulador QKDNetSim adopta como referencia este modelo de capas de la ITU-T, adaptado el modelo a sus objetivos prácticos implementando únicamente los componentes esenciales para estudiar el comportamiento del sistema en términos de flujo de claves, gestión de buffers, retransmisión entre nodos y posterior suministro a las aplicaciones. De esta manera, se centra en la interacción entre las capas de gestión y de control de claves, dejando neutrales otros bloques de la arquitectura.

A partir de esta adaptación, ilustrada en la Figura 26, se diferencian en color negro las funciones implementadas, de aquellas que no se incluyen en el simulador, ilustradas en color gris. La relación concreta es la siguiente:

- **Capa cuántica:** Esta capa se abstrae de la estructura teórica real, se detallarán sus características en el siguiente apartado 4.2.1, y solo algunas funciones están implementadas para adaptarse al simulador:
  - **Depuración de clave ("*Key distillation*"):** El simulador representa de manera simplificada el proceso que lleva a cabo un módulo QKD para generar claves listas para usarse como si hubieran pasado por un protocolo QKD.
  - **Suministro de clave QKD ("*QKD-key supply*"):** Representa el envío de material de clave desde esta capa cuántica ficticia, hacia la capa de gestión de claves del KMS como una interfaz de control.

  En cambio, el simulador no implementa ni simula como tal los módulos que se encargarían de:
  - Sincronización del canal cuántico ("*Quantum channel synchronization*"): Procesos para alinear temporalmente la transmisión de fotones entre emisor y receptor.
  - Generar números aleatorios (RNG, "*Random Number Generator*"): Necesario para preparar estados cuánticos verdaderamente aleatorios en el protocolo QKD.
  - Comunicación cuántica ("*Quantum communication*"): Transmisión de los fotones codificados en estados cuánticos polarizados.
  - Multiplexación de canal ("*Channel multiplexing*"): Técnicas para combinar en una misma infraestructura de fibra, varios canales cuánticos.

- **Capa de gestión de claves**: En esta capa se encuentran la mayor parte de funciones implementadas. Representa el núcleo del simulador modelando el KMS:
  - **Suministro de claves (KSA):** Módulo que entrega las claves a las aplicaciones criptográficas, en el caso de implementar ETSI QKD 014 proporciona métodos como "GET_KEY" o "GET_KEY_WITH_KEY_IDS".
  - **Gestión de claves (KMA):** Agrupa varias funciones fundamentales:
    - **Retransmisión de claves ("*Key relay*")**: Permite retransmitir claves a través de nodos intermedios de confianza, posibilitando la comunicación entre extremos que no comparten un enlace directo.
    - **Almacenamiento de claves ("*Key storage*"):** Almacena las claves en búferes específicos denominados:
      1. **Q-buffers:** Buffers de clave segura, cada KMS tiene uno para cada conexión adyacente, donde almacenan las claves recién generadas
      2. **S-buffers:** Buffers de sesión, que se utilizan de tipo:
         a. "LOCAL": Se establecen dos para cada Q-buffer como se puede ver en la Figura 25, con el objetivo de dividir su flujo de claves, uno para procesos salientes "enc" ("*encryption"*) del propio KMS, y otro para procesos entrantes "dec" ("*decryption"*) cuando sea requerido por un KMS par.

b. "RELAY": Acumulan claves dedicadas a un origen y destino específico en un proceso de retransmisión.
c. "STREAM": Para almacenar flujos de claves reservados para cada sesión específicas para las peticiones de clave ETSI QKD 004.

3. **Gestión del ciclo de vida de las claves ("*Key life cycle management*"):** Controla los estados de las claves, si se encuentra inicializada, lista u obsoleta para sincronizar dichos estados con el nodo par.

- **Control y gestión del KM ("*KM control and management*")**: Supervisa en la simulación todas las operaciones de los KSA y KMA, coordinando la provisión y el almacenamiento de claves, así como el reenvío entre nodos.

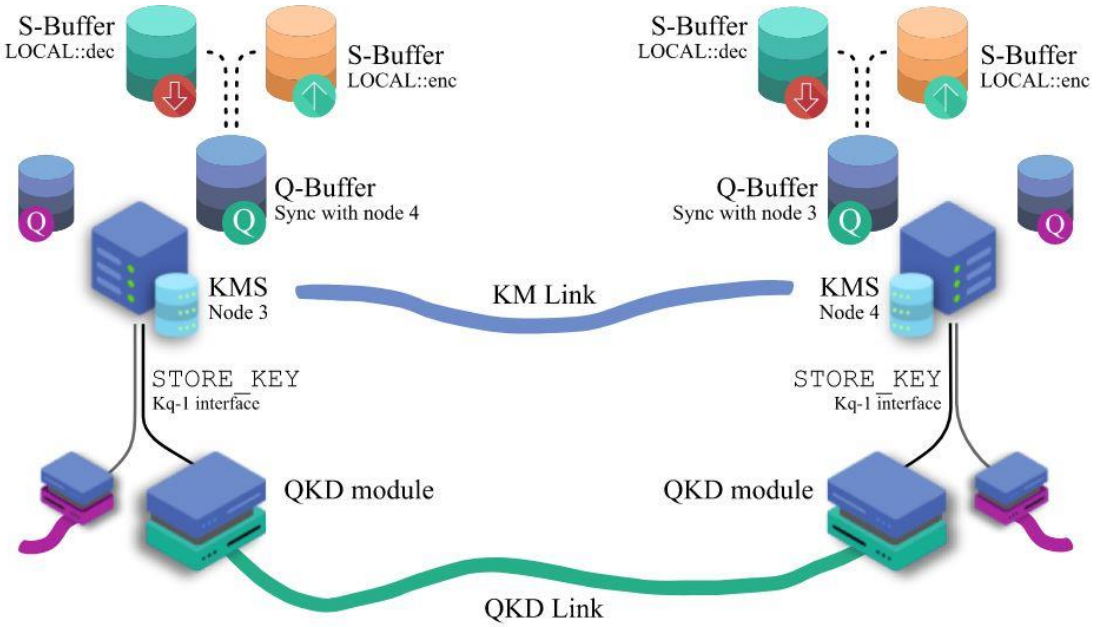


Figura 25. Esquema de almacenamiento de clave en la capa de gestión de clave [44]

- **Capa de control:** En QKDNetSim, la capa de control se implementa de forma parcial:
  - **Control de enrutamiento ("*Routing control*"):** Gestiona la capacidad de decidir qué camino seguirán las claves a través de una red con múltiples nodos. Está implementado de forma básica en el simulador, ya que los controladores rellenan las tablas de enrutamiento utilizando el algoritmo de Dijkstra y la topología de los enlaces de la red QKD. Los pesos de los enlaces se establecen en un valor fijo de 1 y el sistema siempre selecciona la ruta con el menor número de saltos, independientemente del rendimiento o de la disponibilidad de claves en cada enlace.

- **Control de configuración ("*Configuration control*"):** Se encarga de inicializar y mantener los parámetros de cada nodo y enlace de la red QKD.

No son implementados por el simulador los siguientes módulos:

- Control de sesión ("*Session control*"): Administra las sesiones seguras entre aplicaciones, con asignación de identificadores de sesión (KSID). Existe un soporte básico, ya que bajo ETSI QKD 004 genera un KSID al abrir una sesión con "OPEN_CONNECT", sin embargo, no se implementa mayor lógica como escenarios KSID predefinidos.
- Control de acceso ("*Access control*"): Mecanismos que regulan qué aplicaciones se autorizan a usar claves.
- Políticas de control ("*Policy-based control*"): Aplicación de reglas sobre la red, como prioridades de tráfico o calidad de servicio (QoS).
- Control y gestión del controlador de red QKD ("*QKDN control and management*"): Funciones de supervisión y coordinación de toda la capa de control, ausente también en la versión actual del simulador.

- **La capa de servicio:** Donde residen las aplicaciones criptográficas que consumen las claves, QKDNetSim las representa a través de las interfaces normalizadas de ETSI QKD 014 y ETSI QKD 004. Dichas aplicaciones solicitan claves mediante los métodos estándar, y reciben material de clave seguro para cifrar su tráfico correspondiente. Se trata sus características en el siguiente apartado 4.2.1.

- **Capa de gestión de la red QKD:** Este bloque aparece completamente en gris en la Figura 26, abarca las funciones de gestión de fallos, configuración, contabilidad, rendimiento y seguridad ("FCAPS", *Fault, Configuration, Accounting, Performance, Security*) para las capas de control, gestión de claves y cuántica. Adicionalmente, incluye la orquestación de la gestión entre capas ("*Cross-layer management orchestration*"), que coordina la interacción entre estas tres capas, no están implementadas en QKDNetSim ya que exceden el propósito del simulador.

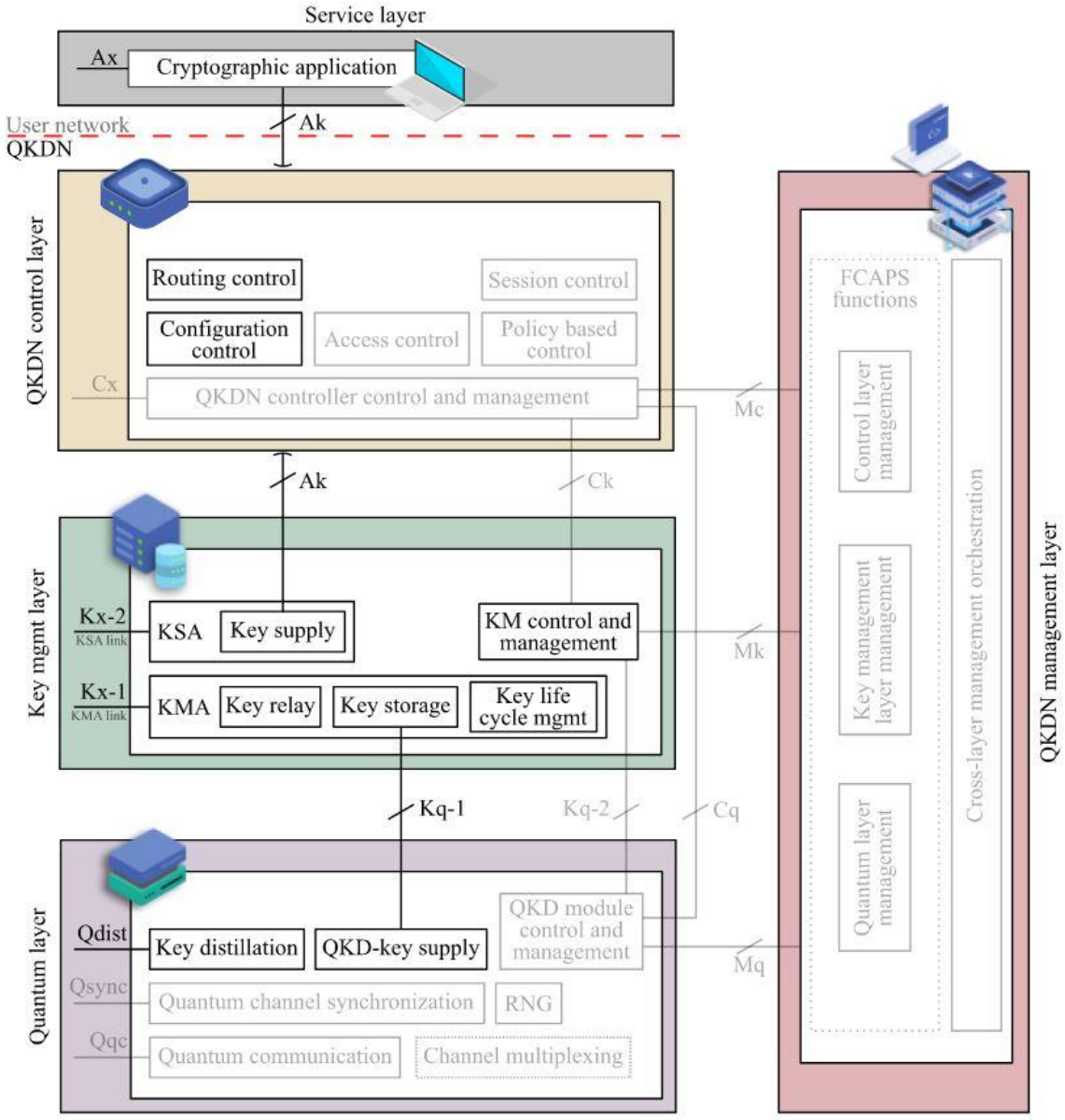


Figura 26. Funciones de la arquitectura implementadas por el simulador QKDNetSim [44]

4.2.1. Características del simulador

El simulador QKDNetSim está diseñado principalmente para emular el comportamiento de una red de distribución cuántica de claves, para ello incorpora un conjunto de aplicaciones personalizadas que se integran en la estructura descrita, para permitir cubrir tanto la simulación de protocolos QKD, como la interacción del sistema con las interfaces normalizadas por ETSI.

- **Aplicación de postprocesado QKD ("QKD Post-Processing Application")**:
  - Cada una de las fases del protocolo BB84 definido en la sección 3.3.1, implica tanto comunicación cuántica para el envío y detección de fotones, como una

comunicación clásica de apoyo para realizar cada fase. Esta aplicación se implementa para simular la actividad de un protocolo QKD, como sería el BB84 o E91, sin realizar cálculos adicionales para evitar carga computacional.

- Funciona de tal manera que para QKDNetSim, estas fases no se implementan de manera física ni algorítmica, sino que se emulan a nivel de tráfico de red. El simulador no genera o transmite fotones, ni procede a medir en distintas bases, por esto, se ignoran los problemas derivados de las comunicaciones en el canal cuántico.

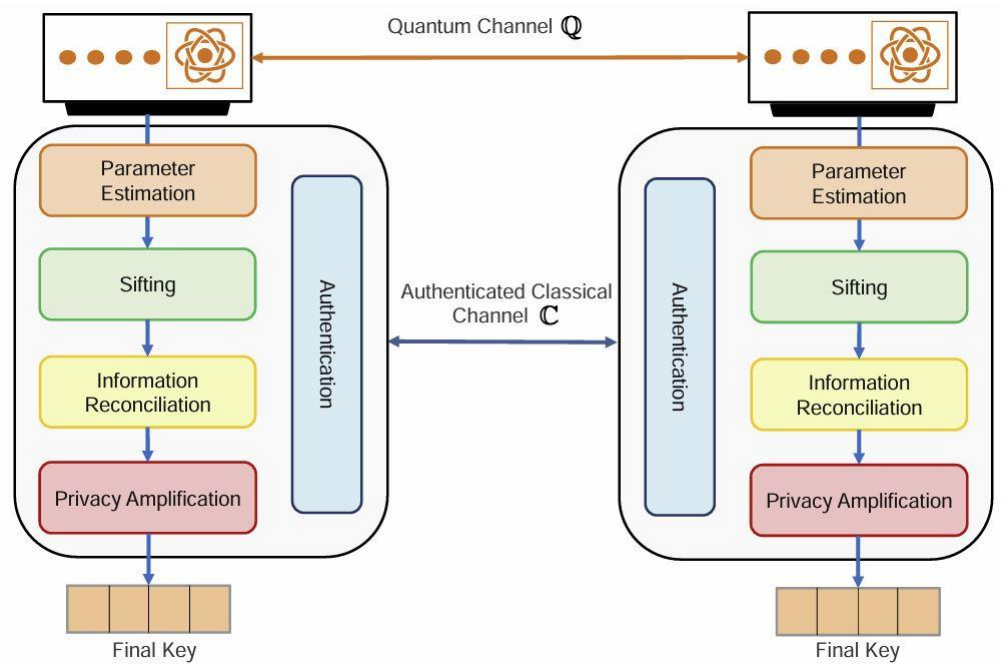


Figura 27. Procesos de protocolo QKD abstraídos en la aplicación de postprocesado [34]

- Se simula por un tiempo determinado un mecanismo mediante valores de contador, en el que los pares de nodos intercambian paquetes que simulan el flujo de información del protocolo QKD. Al final del proceso, las claves generadas con contenido aleatorio se entregan al KMS como si se tratase del módulo QKD visto en teoría.
- Esta aplicación está creada para permitir que se configuren distintos comportamientos del enlace QKD configurando parámetros como la tasa de clave generada, el tamaño de clave generado, junto con la tasa y el tamaño de los paquetes de la fase de postprocesado simulando el protocolo QKD.

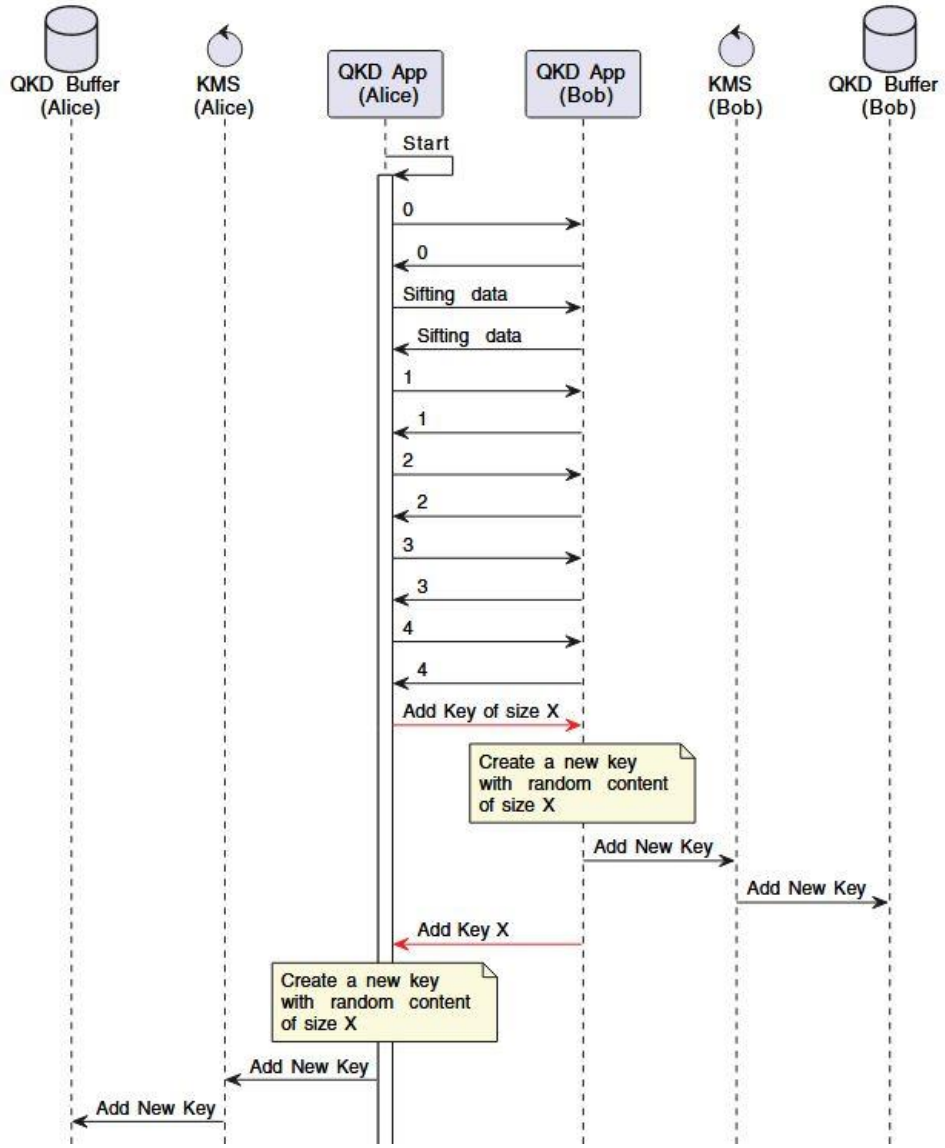


Figura 28. Diagrama de secuencia de la aplicación de postprocesado en QKDNetSim [43]

Por otra parte, el simulador incluye dos aplicaciones de usuario "App014" y "App004" que consumen claves QKD para cifrar datos, de manera que ambas operan según el paradigma maestro-esclavo, donde el maestro envía datos cifrados al esclavo.
Ambas aplican HTTP, en lugar de HTTPS como se introdujo en el apartado 3.4, ya que la seguridad TLS no afecta a los métodos utilizados por las interfaces y se consigue ahorrar tiempo en acciones que no contribuyen al propósito del simulador. También forman las operaciones con el mismo formato URI y encapsulan cada mensaje con un encabezado QKD que indica el algoritmo de cifrado utilizado y los identificadores de clave únicos aplicados, ayudando a las aplicaciones receptoras a procesar los paquetes cifrados y autenticados.

La diferencia fundamental corresponde con la interfaz de KMS utilizada, de tal manera que la ETSI QKD 014 se aplica en la App014 y ETSI QKD 004 en la App004, pero en concreto, se pueden diferenciar los siguientes aspectos:

- **App014:**
  - La aplicación tiene numerosos parámetros configurables por el usuario para admitir simulaciones de diferentes casos de uso.
  - Permite configurar parámetros específicos relacionados únicamente con el funcionamiento de esta aplicación, como el número de claves obtenidas por una única solicitud "GET_KEY" y el tiempo de espera para definir cuánto tiempo espera la aplicación antes de emitir una nueva solicitud si la anterior falla.
  - También se define una estructura de almacenes de claves en la capa de aplicación, donde las claves se clasifican según su uso para determinar cómo las utilizará la aplicación.

- **App004:**
  - Mantiene una cola de claves para cada sesión de flujo de claves establecida, de tal manera que realiza un seguimiento de los flujos de claves, identificados mediante KSID, tanto para el cifrado como para la autenticación.
  - Mediante dos métodos adicionales, establece cómo se comunican los identificadores de claves entre los SAE conectados por el enlace de aplicación, "CONNECT" y "ESTABLISH_QUEUES" para señalar el inicio y el final de la fase de establecimiento de la cola creada.
  - Para iniciar la comunicación, las colas deben acumular primero la cantidad deseada de material de clave mediante las solicitudes "GET_KEY".
  - La clave asociada a la sesión se almacena en un único buffer STREAM, y es la propia aplicación la que decide cómo usarla dentro de la sesión subdividiéndose en claves para cifrado o para autenticación.

4.3. Parámetros simulados

Como hemos visto, en QKDNetSim, las claves seguras son simuladas como secuencias aleatorias generadas por aplicaciones de posprocesamiento de QKD. El conjunto de mecanismos llevados a cabo para realizar la distribución de claves tiene en cuenta múltiples parámetros para realizar los cálculos de las simulaciones.

El simulador abierto al público en su versión actual [46], tiene disponibles una serie de opciones para cada parámetro con el objetivo de permitir distintas configuraciones al usuario. De esta forma, se exponen a continuación los parámetros de entrada configurables para para un enlace QKD y para una aplicación criptográfica.

4.3.1. Parámetros de entrada

Los parámetros que permite configurar el simulador en una configuración de red QKD, son los correspondientes a un enlace QKD entre dos nodos, y los parámetros del enlace de aplicación criptográfica QKD. A continuación, se muestran sus respectivas pantallas de configuración junto con las posibilidades de configuración de los parámetros dispuestos por el simulador.

4.3.1.1. Parámetros configurables del enlace QKD

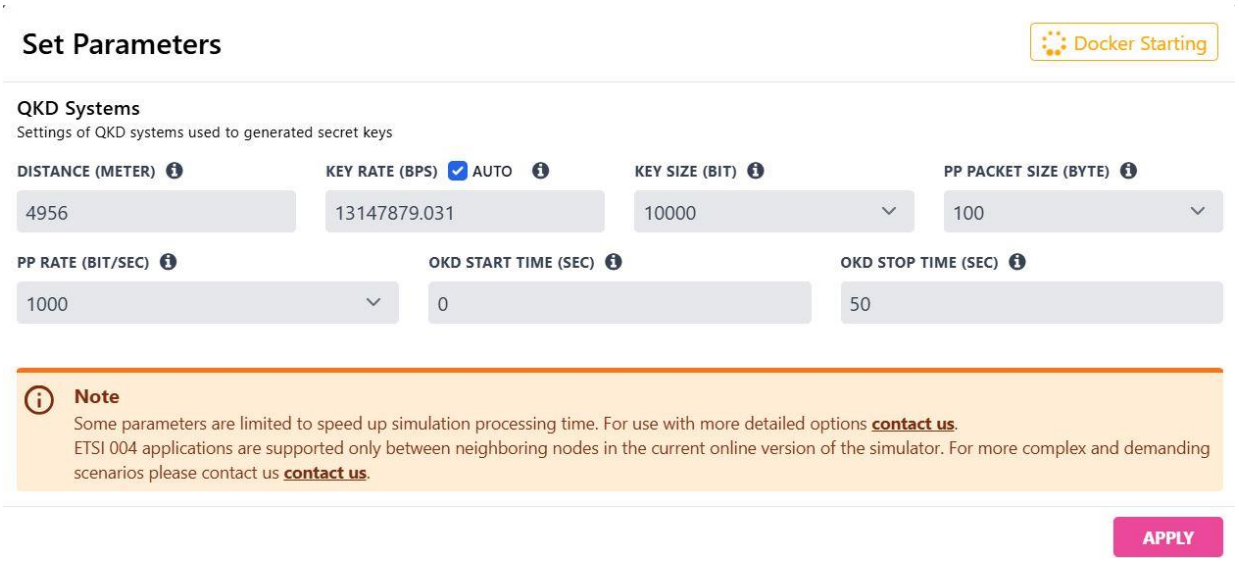


Figura 29. Ventana de configuración de enlace QKD por defecto

- **Distancia del enlace QKD [m]**

  Distancia estimada en metros del camino más corto por carretera entre los dos nodos del enlace a configurar, siguiendo los caminos que detecte gracias a su conexión con Google Maps. Permite editar el valor para ajustar la distancia, pero puede generar confusión en los resultados.

- **Tasa de generación de clave ("*Key rate*") [bps]**

  Se refiere a la cantidad promedio de bits de clave secreta ya procesada que se quiere generar por segundo. Las claves se generen a la tasa más alta admitida y luego se almacenen en un lugar seguro para su posterior uso en procesos criptográficos. El simulador permite dos opciones:

  La primera opción, que aparece por defecto, es en la que se asigna un valor calculado automáticamente por el simulador al situar el enlace entre los nodos, de tal manera que un enlace con una distancia corta entre nodos de en torno a 2 km se le asigna una tasa de ~15,3 Mbps, a 4 km de ~13,8 Mbps y a 10 km ~10 Mbps. Por lo que el sistema automático utiliza tasas de generación de clave generalmente elevadas que disminuyen conforme la distancia aumenta, lo que

tiene sentido teóricamente con el comportamiento teórico que se pretende simular, pero son datos muy elevados en comparación con el estado del arte de redes QKD reales, que para estas distancias se sitúan en la escala de 10^3 y 10^5 bps [19].

También existe una opción manual, donde el rango de valores ofrecido se ajusta mejor a las escalas realistas mencionadas. Se encuentran varias opciones predefinidas al desactivar el botón de cálculo automático, como se ve en la Figura 30, son 5 kbps, 10 kbps, 15 kbps, 20 kbps, 100 kbps y 500 kbps.

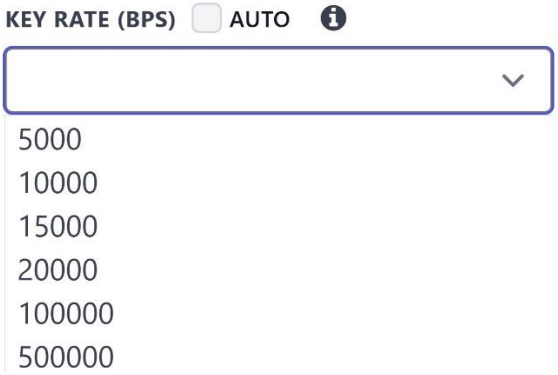


Figura 30. Opciones predeterminadas de configuración de tasa de generación de clave

- **Tamaño clave ("*Key Size*") [bits]**

  Hace referencia a la longitud media de la clave criptográfica que se quiere generar en el enlace seleccionado, fijando el tamaño de los bloques que se van acumulando en el buffer del sistema gestor de claves KMS.

  Los parámetros configurables que se muestran en la Figura 31, muestran que se puede elegir tamaños de clave de 10k, 30k, 50k, y 100k bits. De todas formas, como se analizará más adelante, se plantea la sospecha de que estos valores están expresados en realidad en unidades de bytes y no de bits.

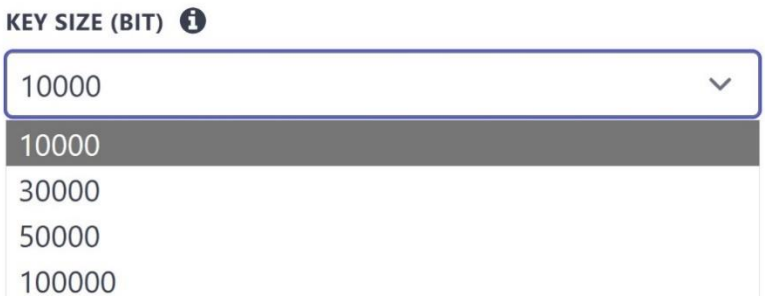


Figura 31. Opciones predeterminadas de configuración de tamaño de clave

- **Tamaño paquete PP (“*PP Packet Size*”) [bytes]**

  Este parámetro hace referencia al tamaño medio que tendrán los paquetes de tráfico intercambiados en el momento del postprocesado simulado de QKD. Los parámetros disponibles son 100, 200 y 300 bytes.

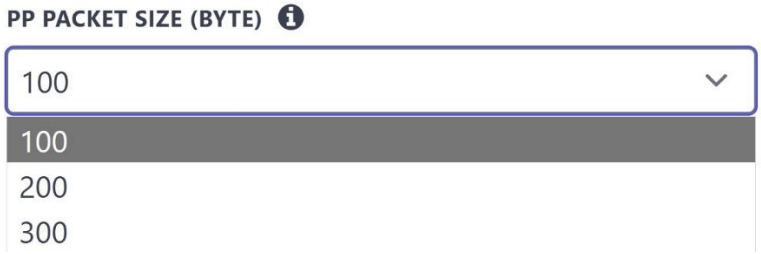


Figura 32. Opciones predeterminadas de configuración de tamaño de paquete del tráfico de postprocesado

- **Tasa de postprocesado (“*PP Rate”)* [bps]**

  La configuración de esta tasa se refiere a la configuración que se quiera establecer para el tráfico promedio de los mensajes de postprocesamiento, simulando cuánto tráfico afectará en el canal clásico y en cuánto tiempo se completará la entrega de claves.

  Las opciones son 1 kbps, 1,5 kbps y 2 kbps, como se ve a continuación.

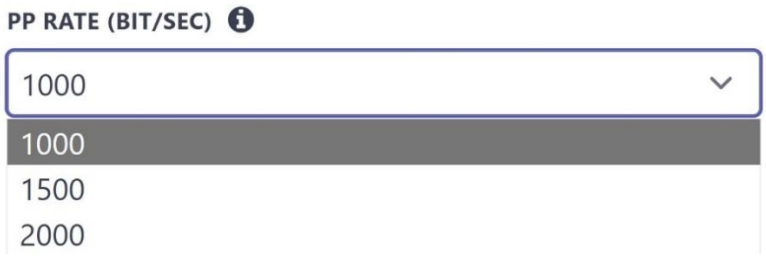


Figura 33. Opciones predeterminadas de configuración de la tasa de postprocesado

- **Inicio (“*QKD Start Time*”) [s] y Fin (“*QKD Stop Time*”) [s]**

  La última opción disponible para configurar en el enlace QKD, se puede analizar conjuntamente como el momento de la simulación en el que los sistemas QKD comienzan y terminan de generar claves, por lo que son los parámetros que definen cuánto tiempo va a estar funcionando la simulación para proporcionar los datos de salida.

  La definición de ambos valores es libre, teniendo en cuenta una ventana máxima de 3600 segundos.

4.3.1.2. Parámetros configurables para una App criptográfica

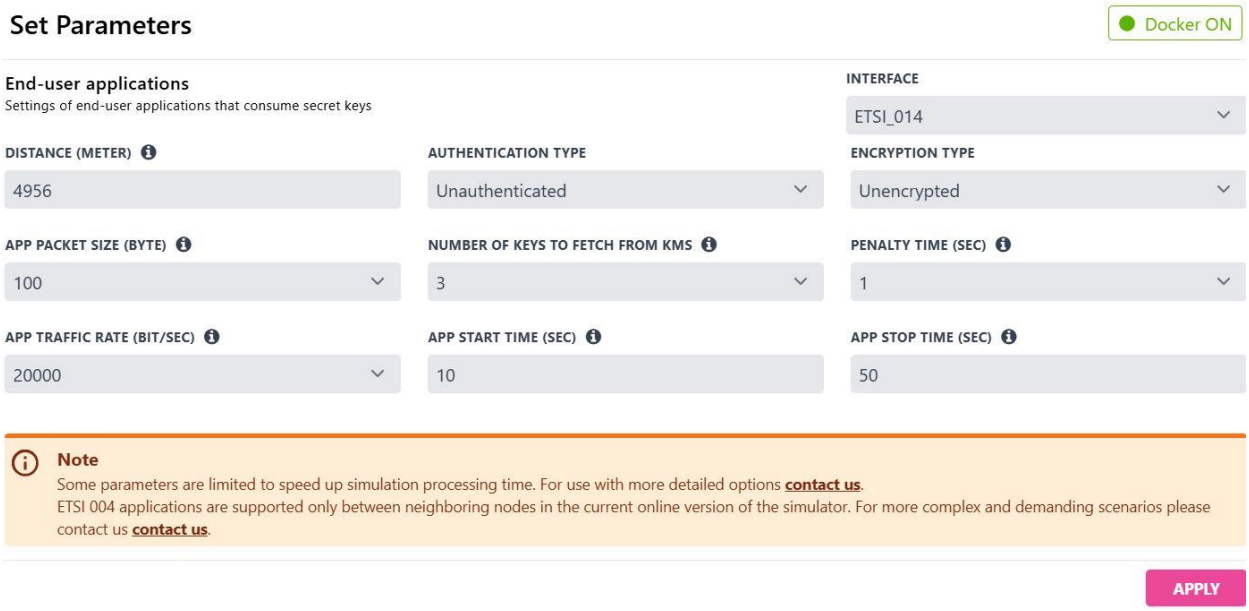


Figura 34. Ventana de configuración de App QKD por defecto

- **Distancia App**

  Muestra la distancia del enlace que une los dos nodos QKD, ya sea un único enlace o la suma de los enlaces, configurados previamente, necesarios para llegar a conectar la ruta de la aplicación que va a consumir claves.

- **Tipo de interfaz ("*Interface*")**

  Esta opción permite variar el modelo de interfaz aplicado para la comunicación entre la aplicación y el KMS correspondiente entre las dos interfaces vistas en el apartado 3.5.3, ETSI-014 y ETSI 004.

  Es importante señalar, que la opción de ETSI-004 solo podrá aplicarse en nodos adyacentes, ya que la especificación actual de QKDNetSim no permite la instalación de la aplicación 004 entre nodos arbitrarios.

Figura 35. Opciones predeterminadas de configuración de tipo de interfaz de aplicación QKD

En base a esto, se tendrá acceso a dos configuraciones diferentes con relación al comportamiento de cada interfaz. ETSI_014 da acceso a modificar los parámetros de número de claves por solicitud y un tiempo de penalización, mientras que ETSI_004 define la capacidad de clave en el buffer de aplicación diferenciado por uso.

- **Número de claves por solicitud (“*Number Of Keys To Fetch from kms*”) [clave] y tiempo de penalización (“*Penalty time*”) [s]**

  La configuración de un número específico de claves por solicitud es una estrategia para la optimización del uso que se hace del KMS, permitiendo establecer que cuando una aplicación necesita clave, pida mediante una única solicitud al KMS 3, 5, 8 o 10 claves a la vez por petición. Corresponde con lo establecido en la recomendación ya que el número de claves solicitadas simula la llamada “GET_KEY”, como se presentó en 3.5.3.2, donde ETSI 014 expone como se puede pedir un lote de claves sucesivas.

  Del mismo modo, se permite configurar un tiempo de espera antes de enviar una nueva solicitud del tipo “GET_KEY” al KMS, después de que se reciba una respuesta indicando que ya no hay suficientes claves almacenadas.

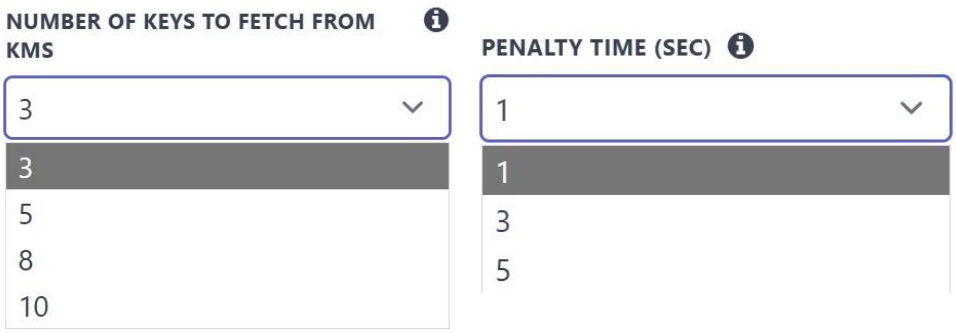


Figura 36. Opciones predeterminadas de interfaz 014

- **Capacidad de clave en buffer de App para cifrado y para autenticación (“*encryption buffer capacity*”) (“*authentication buffer capacity*”) [clave]**

  Permite definir respectivamente cuantas claves se pueden almacenar en el buffer local ligado a la aplicación 004 diferenciando según su uso para cifrado y para autenticación. Este caso permitiría evaluar como el buffer puede gestionar estos distintos parámetros a la vez, cada uno con su propio ritmo.

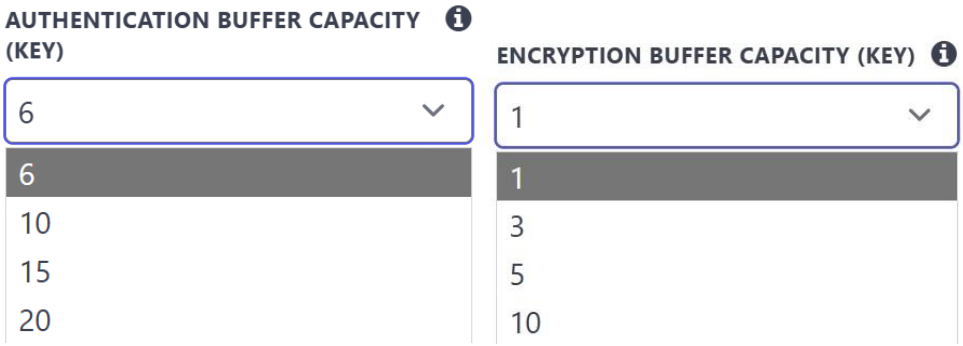


Figura 37. Opciones predeterminadas de interfaz 004

Ya de nuevo con carácter general para la aplicación, se presentan los siguientes parámetros:

- **Tipo de autenticación (“*Authentication type*”)**

  Permite configurar que tipo de autenticación se aplica en la configuración de la aplicación correspondiente a los nodos seleccionados. Las opciones que el simulador permite elegir entre SHA-2 o VMAC. La elección afectaría sobre todo al coste de la autenticación, ya que pese a ser SHA-2 más utilizada, pero VMAC es más eficiente. En este caso, para la autentificación SHA-2 se aplicaría un hash de 256 bits y VMAC una etiqueta de 128 bits más ligera.

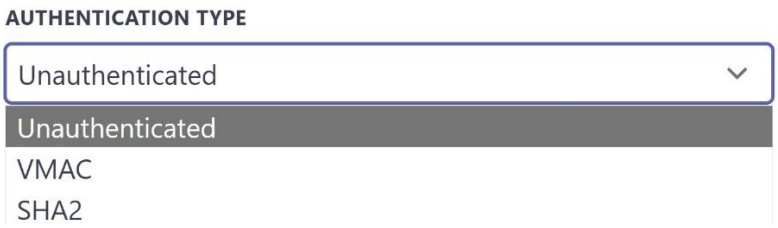


Figura 38. Opciones predeterminadas de configuración de tipo de autenticación de la aplicación

- **Tipo de cifrado (“*Encryption Type*”)**

  El simulador muestra que soporta el establecimiento de un cifrado establecido mediante OTP o AES. En caso de elegir OTP, se correspondería con un consumo de clave igual al consumo de datos, lo que se considera bastante exigente a la hora de consumir clave.

  Si en su lugar se elige AES, se presenta la opción de un nuevo parámetro como “*AES LifeTime*”, que permite estableces cuantos bytes de datos se quieren cifrar con la misma clave, pudiendo elegir entre 10 kB, 20 kB, 100 kB, 200 kB y 300 kB,

permitiendo variar la frecuencia con la que se consume clave QKD del buffer, cada poco tiempo, consumiendo más clave, o, al contrario, consumiendo menos clave.

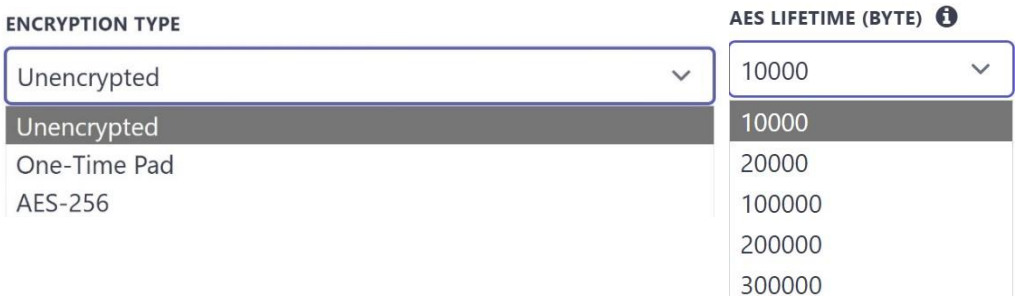


Figura 39. Opciones predeterminadas de configuración del tipo de cifrado de la aplicación

- **Tamaño del paquete de App ("*App Packet Size*") [bytes]**

  Esta configuración permite establecer el tamaño medio del paquete de datos que utilizará la aplicación criptográfica configurada. Pudiendo elegir entre las opciones predeterminadas los valores equivalentes a 100, 300, 500 u 800 bytes.

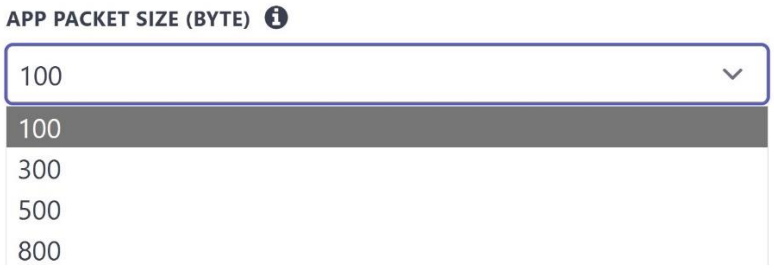


Figura 40. Opciones predeterminadas de configuración del tamaño de paquete de aplicación

- **Tasa de tráfico de App ("*App Traffic Rate*") [bps]**

  Esta tasa, permite configurar el tráfico promedio que tendrá la aplicación que va a consumir claves. Las opciones como se muestran en la siguiente figura, son 20 kbps, 30 kbps, 50 kbps y 100 kbps.

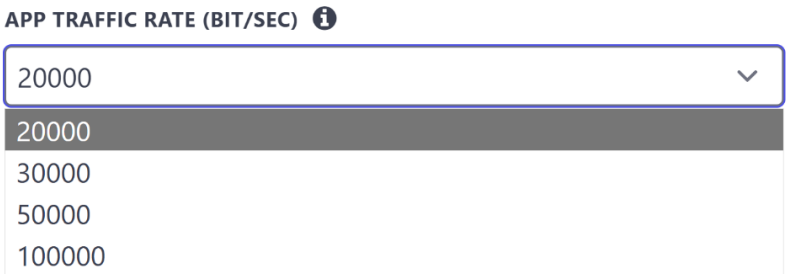


Figura 41. Opciones predeterminadas de configuración de la tasa de tráfico de aplicación

- **Inicio (“APP Start Time”) [s] y Fin (“*APP Stop Time*”) [s]**

  Al igual que con la configuración del enlace QKD, la aplicación criptográfica permite establecer los momentos en los que la aplicación empieza y termina de consumir claves. Es también una definición de valores libre, teniendo en cuenta la ventana de configuración máxima de 3600 segundos y que debería iniciarse unos instantes después del inicio definido en los tiempos de la configuración del enlace, para simular una comunicación lógica con el enlace establecido.

4.4. Caso de uso didáctico para configurar una red de 2 nodos

Se plantea un caso de uso de un sistema QKD desde el entorno web que ofrece QKDNetSim en la URL https://www.open-qkd.eu/. Primero se diseña la red a simular, para posteriormente presentar los pasos a seguir en la configuración y terminar con la exposición de los resultados obtenidos.

4.4.1. Planteamiento

Se quiere simular una red con únicamente dos componentes, que podrían ser Alice y Bob, pero en este caso por simplicidad se les denominará nodos 1 y 2. Estos dos nodos van a estar unidos por un enlace QKD por el que posteriormente se va a configurar un enlace de aplicación QKD. Una vez creada la topología y configurados ambos enlaces, se procede a simular y a obtener los resultados. En el siguiente capítulo se realizarán modificaciones sobre esta misma configuración.

4.4.2. Entorno de configuración y montaje de la red

El entorno de configuración para diseñar las redes QKD, se encuentra disponible como se presenta en la Figura 42, de tal manera que se visualiza el mapa mundial que ofrece Google, permitiendo situar el sistema QKD en cualquier punto que se desee.

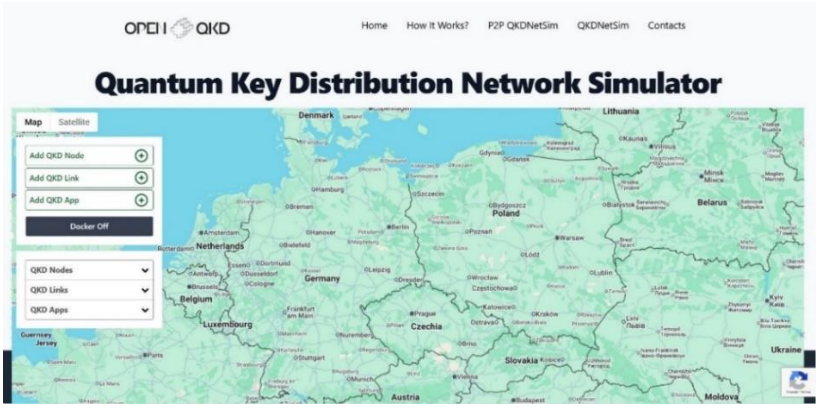


Figura 42. Entorno de configuración del simulador QKDNetSim de OpenQKD

El primer paso consiste en buscar el lugar deseado en el mapa, en este caso se elige una calle del centro de Chicago ya que, pese a ser este un ejemplo de configuración simple, sus calles rectilíneas, largas y cuadradas permiten configuraciones didácticas que ofrecen facilidades visuales y permiten configuraciones más elaboradas, pudiendo visualizar ciertas topologías de forma muy ordenada.

Se selecciona el botón de añadir nodo QKD "*Add QKD Node*" y se selecciona la ubicación de los 2 nodos, como se muestra en la Figura 43, para mayor facilidad y precisión se recomienda ampliar el mapa al lugar exacto. Como se puede ver en la segunda ventana aparecen los nodos que fueron situados, dando también la posibilidad de eliminarlos desde ahí. En los nodos no hay más configuración que su ubicación, la cual habría que tener en cuenta en el caso de querer probar distintas distancias en las simulaciones, ya que la máxima distancia configurable es de 100 km.

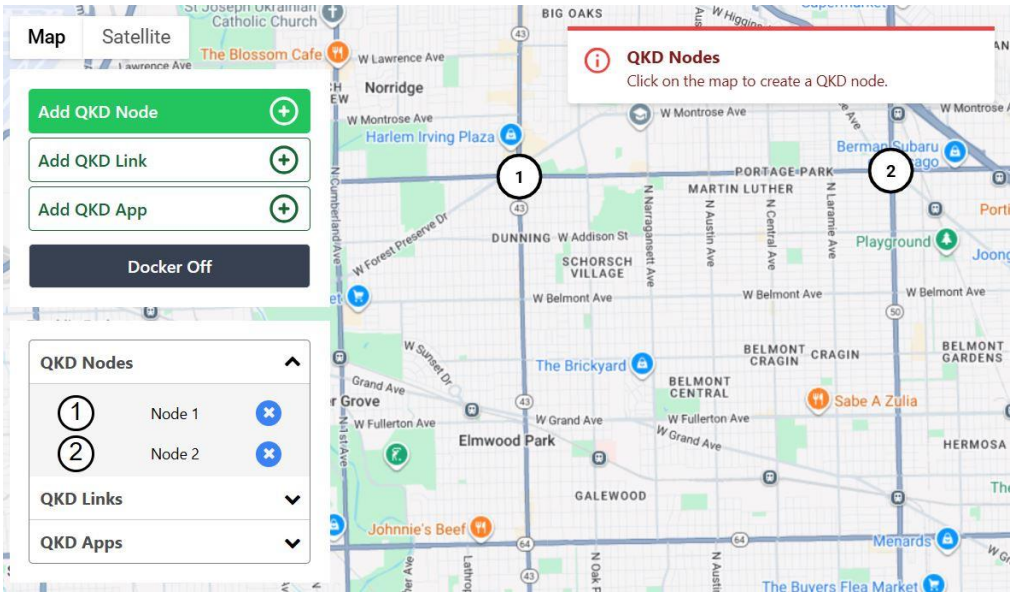


Figura 43. Configuración de 2 nodos

A continuación, se configura el enlace 1-2, seleccionando primero el botón de añadir enlace QKD, "*Add QKD Link*", para luego seleccionar los nodos que se desean unir, en este caso solo el 1 con el 2. Inmediatamente después aparece la pantalla de configuración del enlace de la Figura 29, con los parámetros descritos en el apartado 4.3.1.1, mientras la página empieza a cargar la simulación indicando por pantalla "*Docker Starting*".

Una vez establecidos los parámetros como se desea, se muestra más adelante en la Tabla 7 la configuración elegida para este caso. La visualización una vez aplicada la configuración del enlace sería la de la Figura 44.

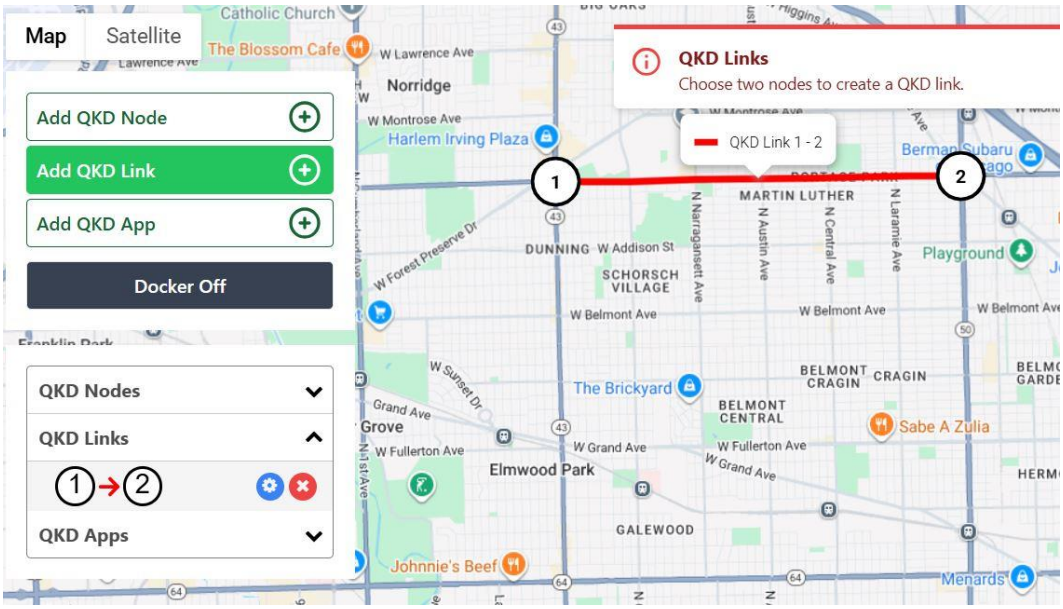


Figura 44. Enlace QKD instalado entre 2 nodos

Posteriormente, solo queda configurar la aplicación entre los nodos 1-2, siguiendo el mismo procedimiento con el botón "*Add QKD App*" y seleccionando los correspondientes nodos, aparece la pantalla de configuración de la aplicación QKD, vista en el apartado 4.3.1.2 con la Figura 34, mientras el estado de conexión de la página debería sin mucho tiempo de espera aparecer como "*Docker ON*", en caso de que aparezca apagado como "*Docker OFF*" presionar sobre el mismo para actualizar.

Se configuran para este caso los parámetros expuestos en la Tabla 8, resultando finalmente la topología de red que se muestra en la Figura 45.

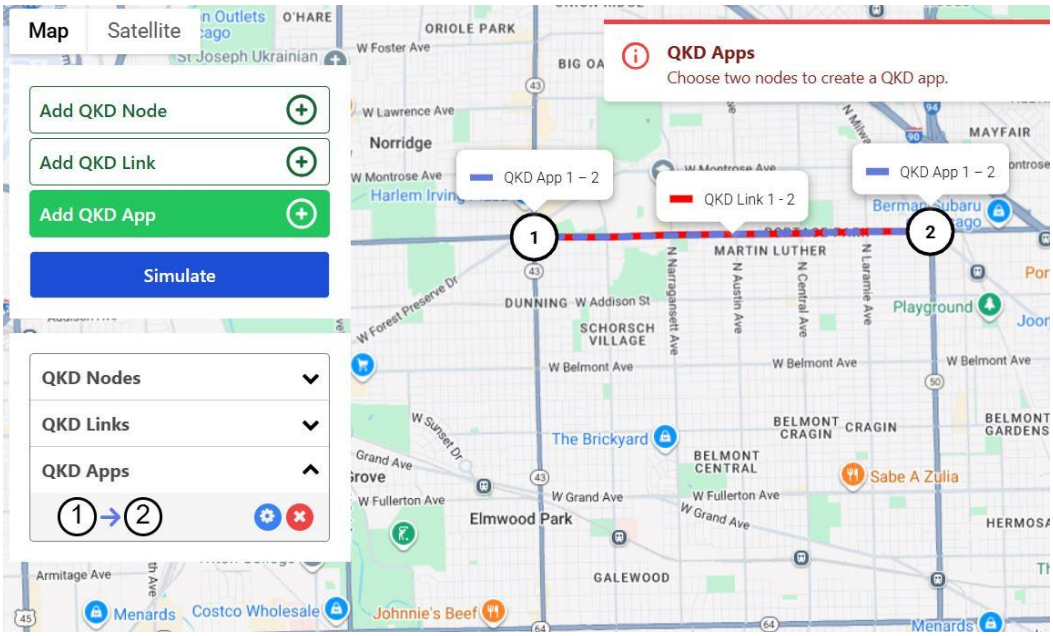


Figura 45. Aplicación QKD entre 2 nodos

Cabe destacar, que la interfaz web limita el número de nodos QKD instalados y las aplicaciones creadas, con el objetivo de ayudar a la obtener una simulación de red QKD rápida. De manera que, teniendo en cuenta que el simulador no permite una topología con nodos desconectados, el número máximo de nodos configurables que se puede alcanzar a crear es una topología de red con 15 nodos, 14 enlaces y un máximo de 13 aplicaciones criptográficas entre ellos.

Con ello, si ya se ha terminado de añadir nodos, enlaces y aplicaciones, como es nuestro caso, el último paso es dar al botón de simular y esperar generalmente unos instantes para que carguen los resultados finales, que aparecerán por pantalla con los parámetros de salida que se muestran más adelante en el texto.

4.4.3. Proceso de configuración de entrada del sistema

Después del montaje de la red, para este ejemplo de uso y como caso de partida para la prueba del apartado 5.2, se han elegido las siguientes configuraciones de los parámetros de entrada, donde la Tabla 7 reúne los parámetros propuestos para el enlace QKD y la Tabla 8, los parámetros de la aplicación criptográfica.

CONFIGURACIÓN PLANTEADA DEL ENLACE QKD						
Enlace QKD	**Tasa de generación de clave [bps]**	**Tamaño clave [bits]**	**Tamaño paquete PP [bytes]**	**Tasa de postprocesamiento [bps]**	**Inicio [s]**	**Fin [s]**
1-2	20.000	10.000	100	1000	0	100

Tabla 7. Configuración planteada del enlace QKDNetSim para 2 nodos en caso de uso

Respecto a los parámetros del enlace, se elige una tasa de generación de clave de 20 kbps como un valor realista en el rango intermedio entre los ofrecidos. Por lo tanto, el rendimiento del enlace QKD entre los nodos 1 y 2 estará configurado para entregar una clave segura a una velocidad constante y garantizada de 20 kbps. Esta será la velocidad a la que el enlace QKD, en principio, puede rellenar el búfer de claves del KMS. Por otro lado, se ha elegido un tamaño de 10 kbits por bloque de clave segura.

Se configuran también los paquetes de postprocesamiento al mínimo posible de 100 bytes con una tasa de postprocesamiento de 1 kbps. Todo ello en un periodo de simulación del enlace enmarcado en 100 segundos.

CONFIGURACIÓN PLANTEADA DE APP CRIPTOGRÁFICA								
Par App.	**Tasa de tráfico de App [bps]**	**Tamaño del paquete de App [bytes]**	**Interfaz**	**Nº claves por solicitud**	**Autenticación**	**Cifrado**	**Inicio [s]**	**Fin [s]**
1-2	20.000	100	014	3	SHA2	OTP	10	50

Tabla 8. Configuración planteada de App criptográfica para 2 nodos en caso de uso

En paralelo, se configura la aplicación para generar el mínimo tráfico con 20 kbps, bajo la interfaz ETSI GS QKD 014, elegida al ser la más desarrollada por el simulador, mediante paquetes de aplicación de 100 bytes, solicitudes de 3 claves por petición al KMS, autenticación mediante SHA-2 como la más generalizada y cifrado con OTP para simular un ejemplo teórico ideal, durante un tiempo de 40 segundos después de esperar 10 segundos iniciales. El tiempo de penalización se fija a 1 segundo.

4.4.4. Proceso de exposición de los resultados

Tras ejecutar la simulación, el simulador genera dos gráficas, una correspondiente al enlace 1-2 y otra correspondiente a la aplicación entre los nodos 1-2.

Los resultados correspondientes al enlace 1-2 se muestran en la Figura 46. En primer lugar, aparece una gráfica con los pares de clave retransmitidos, consumidos y generados. También aparecen unos datos numéricos en formato tabular, que se reproducen en la Tabla 9 para mejorar la legibilidad.

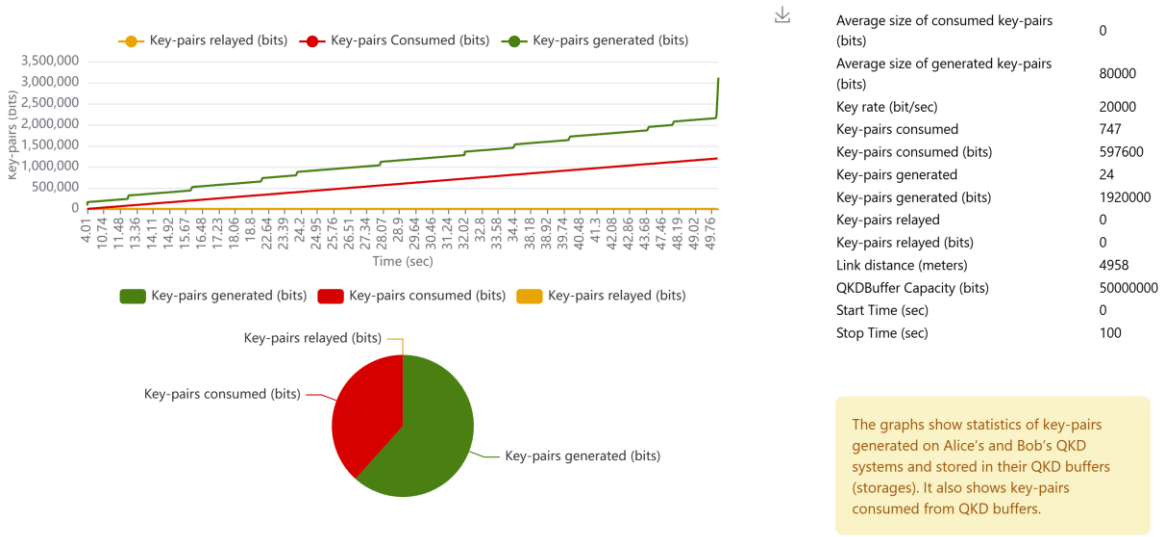


Figura 46. Resultados correspondientes al enlace QKD 1-2

RESULTADOS DEL COMPORTAMIENTO DEL ENLACE	
ENLACE 1-2	
Tamaño promedio de los pares de claves consumidos (bits)	0
Tamaño promedio de los pares de claves generados (bits)	80.000
Pares de claves consumidos	747
Pares de claves consumidos (bits)	597600
Pares de claves generados	24
Pares de claves generados (bits)	1920000
Pares de claves retransmitidos	0
Pares de claves retransmitidos (bits)	0
Distancia del enlace (metros)	4957
Capacidad del búfer QKD (bits)	50.000.000

Tabla 9. Resultados del comportamiento del enlace 1-2 del caso de uso

Los dos últimos parámetros de esta tabla corresponden, primero, a la distancia en metros del enlace configurado que, al haber elegido la tasa de generación de clave a mano entre las opciones disponibles, se podrá comprobar que el parámetro de la distancia entre los nodos pierde sentido y modificando cualquier distancia de los nodos con estos parámetros, dará como resultado los mismos parámetros de salida. Por otro lado, la capacidad del buffer QKD, con un valor de 50.000.000 bits, es probablemente un valor por defecto del simulador para permitir operaciones sin restricciones ya que se mantiene igual por defecto.

Dicho esto, se comenzará el análisis centrando la atención en los resultados sobre la generación de claves. Cuando se hace referencia a los pares de claves, en el contexto de QKD, un "*Key-pairs*" es una unidad fundamental que se refiere al mismo bloque de clave que existe de forma idéntica y segura en ambos extremos del enlace, el 1-2 en este caso. La Figura 47 muestra la evolución en la generación de pares de clave a lo largo del tiempo.

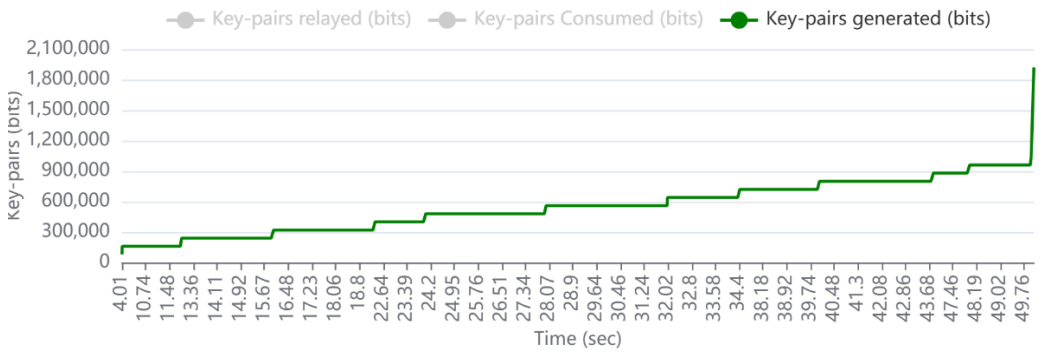


Figura 47. Generación de pares de clave en el enlace QKD 1-2

La configuración elegida establecía que se generaran claves desde t = 0 hasta 100 segundos. Sin embargo, en la figura solo aparecen los resultados hasta t = 50 s (que era el momento en que dejaba de estar activa la aplicación de consumo de claves). Además, la gráfica no refleja los instantes iniciales.

El enlace QKD se había configurado para generar claves de 10 kbits a 20 kbps durante 100 segundos. Por lo tanto, esperaríamos que el número de bits de clave generados en el tiempo de simulación fuera:

$$\text{Nº total de bits de clave generados} =$$

$$= \text{Tasa de generación de clave} \cdot \text{Tiempo de simulación} =$$

$$= 20.000 \text{ bps} \cdot 100 \text{ s} = 2.000.000 \text{ bits de clave}$$

Dado que las claves eran de 10 kbits, esperaríamos generar:

$$\frac{\text{Nº total de bits de clave generados (bits)}}{\text{Tamaño de clave } (\frac{\text{bits}}{\text{par}})} = \frac{2.000.000 \text{ bits}}{10.000 \text{ bits/par}} = 200 \text{ pares de claves}$$

Esto implicaría, en promedio, la generación de una clave cada 0,5 s.

Sin embargo, la Tabla 9 muestra que el tamaño promedio de los pares de claves es de 80 kbits en lugar de 10 kbits. Al realizar pruebas adicionales con otros parámetros, se observa que dicho tamaño es sistemáticamente ocho veces mayor que el tamaño de clave especificado en la configuración. Esto sugiere que es posible que el parámetro en cuestión podría interpretarse internamente en bytes, aunque en la interfaz de usuario aparezca indicado en bits.

Asumiendo esta interpretación, cada clave sería de 10 kbytes y por tanto de 80 kbits. Con ello, el número teórico de pares de claves generados sería:

$$\frac{2.000.000 \text{ bits}}{80.000 \text{ bits/par}} = 25 \text{ pares de claves}$$

Este valor se aproxima al mostrado en la Tabla 9 (24 pares). Considerando los 25 pares teóricos, puesto que se generan claves durante 100 segundos, se produciría un par, en media, cada 4 segundos. Eso es lo que se muestra en la figura escalonada. Aunque en la figura los instantes de generación de nuevas claves parecen no estar equiespaciados, en realidad sí lo están. La representación gráfica es confusa porque

los ticks están equiespaciados pero, por ejemplo, el tiempo entre el primer y el segundo tick es superior a los 6 segundos y entre el segundo y el tercero inferior a 1 segundo. En definitiva, en t = 4,01 s, se genera la primera clave (80 kbits generados), en t = 8,01 s la segunda (y por tanto se tiene un total de 160 kbits de clave generados), en t = 12,01 s la tercera (240 kbits de clave) y así sucesivamente. Ese ligero retraso de aproximadamente 0,01 s, quizás debido al postprocesado, explica por qué se han generado 24 pares en la simulación en lugar de 25: la última clave (el último par) se habría generado en t = 100,01 s, pero la simulación termina en t = 100 s.

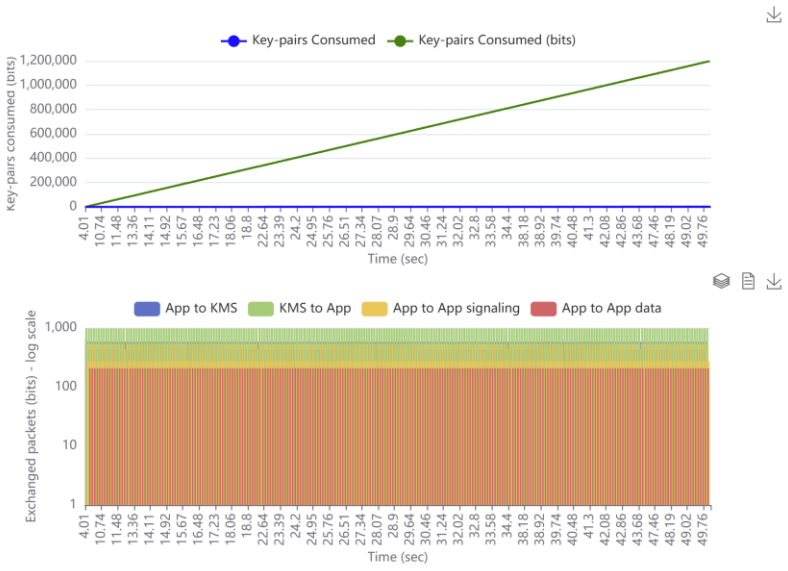
Así pues, el número total de bits de pares de clave generados se corresponde con

$$24 \text{ pares de clave} \times 80 \text{ kbits/par} = 1.920.000 \text{ bits}$$

lo cual coincide con el dato de la Tabla 9 .

Con respecto a los pares de claves retransmitidos, como se observa en la Tabla 9, en este caso simulado es 0, debido a que el enlace une dos nodos que están conectados directamente sin ninguna clave que retransmitir.

Las claves consumidas dependen de las aplicaciones que se hayan configurado. En este caso solo hay una aplicación, así que se analizará directamente. La Figura 48 muestra la evolución de las claves consumidas (en bits) y las estadísticas más relevantes para dicha aplicación, recogidas además, parcialmente, en la Tabla 10.



Key Consumption Statistics

Average size of consumed key-pairs (bits)	800
Key-pairs consumed	747
Key-pairs consumed (bits)	597600

QKDApps Statistics

Authentication	SHA-1
Bytes Received	153264
Bytes Sent	153264
Encryption	OTP
Key/Data utilization (%)	74.41
Missed send packet calls	256
Number of Keys to Fetch From KMS	3
Packet Size (bytes)	100
Packets Received	744
Packets Sent	744
Start Time (sec)	10
Stop Time (sec)	50
Traffic Rate (bit/sec)	20000

QKDApps-KMS Statistics

Bytes Received	488451
Bytes Sent	253122
Packets Received	503
Packets Sent	503

Signaling Statistics

Bytes Received	200694
Bytes Sent	200694
Packets Received	498
Packets Sent	498

Figura 48. Resultados correspondientes a la aplicación QKD 1-2

RESULTADOS DE SALIDA DE SIMULACIÓN DE APLICACIÓN		
APLICACIÓN 1-2		
Estadísticas de consumo de claves	Tamaño medio de los pares de claves consumidos (bits)	800
	Pares de claves consumidos	747
	Pares de claves consumidos (bits)	597600
Estadísticas de QKD Apps	Bytes recibidos	153264
	Bytes enviados	153264
	Porcentaje de utilización de claves/datos (%)	74,41
	Intentos de envío de paquetes perdidos (llamadas)	256
	Paquetes recibidos	744
	Paquetes enviados	744
Estadísticas de comunicación QKDApps-KMS	Bytes recibidos	488451
	Bytes enviados	253122
	Paquetes recibidos	503
	Paquetes enviados	503
Estadísticas de señalización	Bytes recibidos	200694
	Bytes enviados	200694
	Paquetes recibidos	498
	Paquetes enviados	498

Tabla 10. Resultados de la simulación de la aplicación 1-2 del caso de uso

Los pares de clave consumidos representan la utilización real de claves a las que la aplicación dio uso en el tiempo configurado para cifrar los datos. Teniendo de nuevo en cuenta el cálculo teórico para un sistema ideal con datos configurados, se puede calcular la demanda teórica máxima de la aplicación con el siguiente cálculo:

$$\text{Nº total ideal de bits de datos de aplicación en tiempo de simulación} =$$

$$= \text{Tasa de tráfico de App} \cdot \text{Tiempo de simulación del enlace} =$$

$$= 20.000 \text{ bps} \cdot 40 \text{ s} = 800.000 \text{ bits de datos}$$

Podemos comprobar que 800.000 bits representan la demanda teórica de la aplicación, lo que equivale a la cantidad de bits de clave que la aplicación necesita para encriptar los 40 segundos de tráfico de datos. Sin embargo, el dato que proporciona la simulación de pares de clave consumidos en bits, 597.600, representan el consumo real, que es la cantidad de bits que la aplicación realmente pudo usar para cifrar los datos. La diferencia entre el consumo teórico ideal y el real son 202.400 bits, que están ligados a la existencia de "*Missed send packet calls*" (intentos de envío de paquete perdidos), que denotan instancias en las que la aplicación se detuvo por falta de material de clave.

De esta forma, los 747 pares de clave generados se deben al siguiente cálculo:

$$\frac{\text{Pares de claves consumidos (bits)}}{\text{Tamaño del paquete de App (bits)}} = \text{Pares de claves consumidos}$$

$$\frac{597.600}{(100 \cdot 8)\ \text{bits}} = 747\ \text{pares}$$

Respecto a los resultados de salida de la aplicación criptográfica, podemos ver que el simulador los divide en cuatro bloques principales de resultados, donde cada uno de los cuales muestra un aspecto diferente del comportamiento de la aplicación y de su interacción con la red QKD.

- **Estadísticas de consumo de claves**

  Este bloque muestra resume parámetros ya vistos en el enlace como cuántos pares de claves secretas utilizó la aplicación durante la simulación. Cada par de claves representa para la aplicación un conjunto de bits que se reserva para cifrar y descifrar datos de aplicación.

  - Se destacan los dos primeros parámetros, el primero, el tamaño medio de los pares de claves consumidos, que corresponde exactamente al tamaño de los paquetes de datos de la aplicación, en este caso 100 bytes, convertido a bits 800.

  - El segundo parámetro se corresponde al mismo dato que aparece en la tabla de resultados del enlace indicando los pares de clave consumidos, con el mismo dato convertido a bits en el tercer parámetro.

- **Estadísticas de QKD Apps**

  Este bloque refleja el tráfico real de datos de la aplicación criptográfica simulada.

  - Los dos primeros parámetros reflejan la cantidad de tráfico en bytes, que corresponde al mismo para el volumen de datos que es recibido y enviado, por lo que se demuestra que en ambas direcciones es simétrico y será siempre así, por ello solo se destaca en la tabla uno de ellos.

  - Lo mismo ocurre con los dos últimos parámetros de esta sección, que representan los paquetes recibidos y enviados con éxito, donde se puede ver que cada paquete de aplicación transmitido ha aumentado debido posiblemente a introducir cabeceras de aplicación QKD con información

de identificadores y etiquetas del proceso de transmisión. Cada paquete tiene una sobrecarga de 106 bytes, de modo que tienen un total de 206 bytes, por lo que al multiplicar por los 744 paquetes enviados salen los 153.264 bytes mencionados en la Tabla 10. Como se observa, el número de paquetes enviados es 744, que es inferior en 3 unidades al valor de 747 pares de clave. Más adelante se analizará este aspecto.

- El parámetro de salida del "Porcentaje de utilización de claves/datos", hace referencia a cuánta de la clave disponible se utilizó en el proceso de manera correcta para encriptar los datos. Este resultado puede ser inferior al 100%, estando relacionado con el siguiente parámetro, el cual indica los "Intentos de envío de paquetes perdidos". Son llamadas de la aplicación criptográfica que no llegan a utilizar clave porque no consiguen salir como paquete, por ejemplo, si la aplicación pide 3 claves por paquete y en el momento de la solicitud el KMS solo tiene 1, o si el KMS no responde antes de cierto umbral de tiempo, la aplicación debe abortar el envío.

  La Tabla 10 muestra que hay 256 envíos de paquetes perdidos. Tal y como se comentó anteriormente, el nº total ideal de bits de datos de aplicación era 800.000 y el tamaño de un paquete de aplicación eran 100 bytes, luego:

$$\text{Nº de paquetes ideal} = \frac{\text{Nº total ideal de bits de datos de App.}}{\text{Tamaño del paquete de App (bits)}} =$$

$$\frac{800.000 \text{ bits}}{(100 \cdot 8)\text{bits/paq}} = 1000 \text{ paquetes}$$

  Al sumar los 744 paquetes enviados más los 256 intentos de envío de paquetes perdidos se obtiene precisamente este valor de 1000 paquetes. Como se indicó anteriormente, esos 256 "*missed send packet calls*" corresponden, según la documentación, a instancias en las que la aplicación se detuvo por falta de material de clave.

- Por otra parte, ciertos datos de salida de este apartado se omiten en la tabla para no repetir parámetros correspondientes a la configuración como tipo de cifrado y autentificación, donde llama la atención que, pese a haber elegido en la configuración SHA-2, aparece en la salida como configurado SHA-1. Los procesos de autentificación y cifrado según las posibilidades que ofrece QKDNetSim, se simula el tráfico equivalente, es decir que no se calcula matemáticamente un hash, pero sí se reflejan cambios en los parámetros según cada tipo.

Este comportamiento se puede demostrar gracias al parámetro de salida de porcentaje de utilización de clave, que mide cuánta de la clave disponible se logra realmente utilizar para el envío de datos con respecto a lo que se solicita, de tal manera que:

$$\%\text{ de utilización de claves} = \frac{\text{Paq. enviados} \cdot 100}{\text{Nº máx. de paquetes ideal}}$$

$$\frac{100 \cdot 744}{1000} = 74.4 \sim 74.41\%$$

Por lo tanto, se comprueba que el porcentaje de utilización de clave para el envío de datos corresponde con el parámetro de salida de paquetes recibidos/enviados y se justifica que los intentos de envío no llegan a consumir clave.

Hasta este punto, también llama la atención no tener una concordancia entre pares de claves consumidos 747 y paquetes de aplicación enviados/recibidos 744, esto es consecuencia del cumplimiento de las medidas de ETSI QKD 014. Se observa como el KMS asigna clave que puede no llegar a usarse, pero se marcan como consumidas igualmente,

$$\text{Paq. App exitoso} + \text{Paq. App no utilizado} = \text{Pares de clave consumidos}$$

$$744 + 3 = 747$$

Estos tres paquetes, no llegan a contarse como recibidos o enviados pueden ser debido a, por ejemplo, descartes por "*time out*", o por ser rechazados por la capa de aplicación, estos paquetes si llegan a consumir clave, pero no se contabilizan como enviados.

- **Estadísticas de comunicación QKDApps-KMS**

  Esta métrica mide la comunicación entre la aplicación y el gestor de claves KMS.

  - El parámetro de bytes enviados se refiere a los datos que la aplicación envía al KMS, como las peticiones de clave y los bytes recibidos, son los datos que la aplicación recibe del KMS como por ejemplo las claves recuperadas. Este tráfico ocurre en paralelo al tráfico de datos entre los nodos 1 y 2.

- Los paquetes recibidos y enviados son los mismos, pero se puede observar que los bytes recibidos son muchos más que los bytes enviados, esto se debe a que los paquetes enviados al KMS son pequeñas peticiones de clave y los paquetes recibidos del KMS contienen las claves con contenido más pesado, cumpliendo con lo que debe ocurrir.

- **Estadísticas de señalización**

  Corresponden con los mensajes que se envían entre las entidades de aplicación criptográfica para coordinarse por ejemplo antes de enviarse los datos entre ellas.

  - En los parámetros de salida obtenidos, vemos como se indican el número de paquetes recibidos y enviados con su correspondiente equivalencia en bytes, con los mismos valores para cada uno de ellos.

Estos datos demuestran que el sistema abstrae la generación de la clave y se enfoca en cómo la aplicación consume esa clave mostrando un análisis de diversos parámetros de la red. Los resultados demuestran que el simulador QKDNetSim se enfoca en la interacción entre la generación de clave y el consumo por parte de las aplicaciones, de manera que permite explorar cómo la clave generada y el tráfico de señalización necesario impactan en el rendimiento.

Capítulo 5

5. Escenarios de simulación

En este capítulo, se plantean dos escenarios de prueba para analizar los resultados que proporciona la plataforma de simulación QKDNetSim frente a cambios en su configuración. En primer lugar, se define la metodología que se seguirá en el capítulo, a continuación, se presenta un primer caso de estudio que continúa la exposición presentada en el capítulo anterior. Seguidamente, sobre esta primera prueba se realizan dos pequeñas modificaciones para analizar su comportamiento con respecto a la primera prueba. Finalmente se presenta un segundo caso de estudio que incorpora un nodo intermedio a la configuración.

5.1. Planteamiento y parámetros bajo análisis

Para evaluar la relación entre la generación cuántica de clave y el consumo de clave por parte de una aplicación criptográfica, se plantean la topología de red configurada junto con la configuración de los parámetros de entrada del simulador.

La idea central del experimento es variar el tamaño de paquete de la aplicación entre las opciones disponibles 100, 300, 500 y 800 bytes, manteniendo constantes el resto de los parámetros, con el fin de observar cómo esta variación del tráfico de aplicación afecta a la gestión de claves en el KMS por parte del sistema.

5.2. Primera prueba simulada con 2 nodos

A continuación, se muestra la topología de la red, planteada ya en el caso de uso, para dos nodos unidos por el enlace QKD 1-2 y la aplicación criptográfica 1-2.

Tabla 11. Topología de red QKDNetSim con 2 nodos en caso de uso

5.2.1. Configuración del sistema variando el tamaño de aplicación criptográfica

Como se puede observar, se mantienen las mismas dos tablas del caso de uso visto en el anterior capítulo, pero en este caso la Tabla 13, se muestran destacadas las cuatro opciones de tamaño de aplicación, donde se realiza una simulación para cada una de ellas.

CONFIGURACIÓN PLANTEADA DEL ENLACE QKD						
Enlace QKD	**Tasa de generación de clave [bps]**	**Tamaño clave [bits]**	**Tamaño paquete PP [bytes]**	**Tasa de postprocesado [bps]**	**Inicio [s]**	**Fin [s]**
1-2	20.000	10.000	100	1000	0	100

Tabla 12. Configuración del enlace QKDNetSim para 2 nodos en Prueba 1

CONFIGURACIÓN PLANTEADA DE APP CRIPTOGRÁFICA								
Par App.	**Tasa de tráfico de App [bps]**	**Tamaño del paquete de App [bytes]**	**Interfaz**	**Nº claves por solicitud**	**Autenticación**	**Cifrado**	**Inicio [s]**	**Fin [s]**
1-2	20.000	100	014	3	SHA2	OTP	10	50
		300						
		500						
		800						

Tabla 13. Configuración planteada de la App criptográfica para 2 nodos para Prueba 1

5.2.2. Resultados

Se muestran de igual manera que en el caso de uso, las dos tablas de resultados que se obtienen en cada simulación, donde la Tabla 14 muestra las salidas de las cuatro simulaciones para el comportamiento del enlace configurado, y la Tabla 15 muestra de la misma manera las salidas de las cuatro simulaciones para la aplicación criptográfica simulada en cada caso.

RESULTADOS DEL COMPORTAMIENTO DEL ENLACE				
ENLACE 1-2	**Tamaño del paquete de App**			
	100 bytes	**300 bytes**	**500 bytes**	**800 bytes**
Tamaño promedio de los pares de claves consumidos (bits)	0	0	0	0
Tamaño promedio de los pares de claves generados (bits)	80.000	80.000	80.000	80.000
Pares de claves consumidos	747	249	150	96
Pares de claves consumidos (bits)	597600	597600	600000	614400
Pares de claves generados	24	24	24	24
Pares de claves generados (bits)	1920000	1920000	1920000	1920000
Pares de claves retransmitidos	0	0	0	0
Pares de claves retransmitidos (bits)	0	0	0	0
Distancia del enlace (metros)	4957	4957	4957	4957
Capacidad del búfer QKD (bits)	50.000.000	50.000.000	50.000.000	50.000.000

Tabla 14. Resultados del comportamiento del enlace 1-2 de Prueba 1

COMPARATIVA DE RESULTADOS DE SIMULACIONES					
APP 1-2		**Tamaño del paquete de App**			
		100 bytes	**300 bytes**	**500 bytes**	**800 bytes**
Estadísticas de consumo de claves	Tamaño medio de los pares de claves consumidos (bits)	800	2400	4000	6400
	Pares de claves consumidos	747	249	150	96
	Pares de claves consumidos (bits)	597600	597600	600000	614400
Estadísticas de QKD Apps	Bytes recibidos	153264	100688	89688	85164
	Bytes enviados	153264	100688	89688	85164
	Porcentaje de utilización de claves/datos (%)	74,41	74,26	74	75.2
	Intentos de envío de paquetes perdidos (llamadas)	256	86	52	31
	Paquetes recibidos	744	248	148	94
	Paquetes enviados	744	248	148	94
Estadísticas de comunicación QKDApps-KMS	Bytes recibidos	488451	300128	264557	248903
	Bytes enviados	253122	85881	52584	34422
	Paquetes recibidos	503	171	105	69
	Paquetes enviados	503	171	105	69
Estadísticas de señalización	Bytes recibidos	200694	66898	40300	25792
	Bytes enviados	200694	66898	40300	25792
	Paquetes recibidos	498	166	100	64
	Paquetes enviados	498	166	100	64

Tabla 15. Comparativa de resultados de simulación de aplicación 1-2 de Prueba 1

5.2.3. Análisis y conclusiones

Esta simulación cumple con todo lo expuesto en el caso de uso, cabe destacar para cada bloque:

- **Estadísticas de consumo de claves**
  - El tamaño promedio de par de claves consumido escala linealmente con el tamaño del paquete configurado convertido a bits: 800, 2400, 4000 y 6400 bits.
  - Al aumentar el tamaño de los paquetes el número de pares consumidos disminuye desde 747 hasta 96 en el caso del tamaño de paquete más grande. Es esperable debido a que, al ser cada paquete de aplicación más grande, hay menos paquetes totales, por lo que se acaban contabilizando menos solicitudes y se entregarán menos pares.
  - En cuanto al consumo de bits, destaca la poca variación que muestran los casos, ya que se mantiene prácticamente constante en torno a los 600.000 bits, Esto indica que el simulador, al mantener las condiciones del enlace, sigue consumiendo prácticamente el mismo porcentaje de la demanda independientemente del tamaño del paquete configurado para la aplicación.
  - Esto muestra la estabilidad de aplicar OTP y que, en este caso, la aplicación está limitada por la tasa de éxito del canal, no por el tamaño requerido de los paquetes.

- **Estadísticas de QKD Apps**
  - Aumentar el tamaño del paquete se refleja en que se transmiten menos paquetes totales y se ve como los parámetros de bytes recibidos y enviados decrecen según aumenta el tamaño de paquete configurado.
  - Al haber menos paquetes en general, también habrá menos intentos fallidos de envío pasando de 256 a 31. Pese a ello, el porcentaje de utilización de claves se mantiene en torno al 74,5%.
  - Cada paquete QKD de aplicación no solo lleva los datos útiles, sino también carga adicional, como son las cabeceras de aplicación QKD y la autentificación, que tendrán menor peso relativo al aumentar el tamaño de los paquetes. Observando los resultados, esa carga adicional son 106 bytes en todos los casos.

- **Estadísticas de comunicación QKDApps-KMS**
  - Tanto el número de paquetes intercambiados por la aplicación y el KMS como su parámetro de bytes, descienden al aumentar el tamaño del paquete.
  - Esto demuestra que a medida que los paquetes son más grandes, se necesitan menos peticiones de clave, por lo tanto, menos solicitudes al KMS y menos bytes totales intercambiados.

- **Estadísticas de señalización**
  - Los parámetros de señalización disminuyen al aumentar el tamaño del paquete, ya que los eventos que requieren señalización disminuyen, de forma que cuanto mayor es el tamaño del paquete de aplicación, menos paquetes necesitan, y menos señalización se requiere comunicar.

En los resultados de esta prueba se observa que, independientemente del tamaño de los paquetes, el porcentaje de utilización de clave se mantiene prácticamente constante en torno al 75%. Esto significa que, desde la perspectiva de la eficiencia del uso de la clave generada en los enlaces, no existe una gran diferencia entre trabajar con paquetes pequeños o grandes. En este caso la aplicación consigue consumir la misma proporción de clave disponible para transmitir su tráfico independientemente de este parámetro.

Podemos destacar que, al aumentar el tamaño del paquete de aplicación, el sistema necesita menos paquetes totales para enviar la misma cantidad de información. De esta forma, la variación observada en el volumen de bytes recibidos se debe al efecto de sobrecarga asociado a cada paquete, donde los paquetes pequeños generan más cabeceras, lo que hace crecer adicionalmente el tráfico reportado, aunque no aumente la información útil transmitida. Por el contrario, los paquetes grandes reducen esta sobrecarga mostrando un menor tráfico total, sin que ello implique una peor eficiencia en clave. Por tanto, la relación entre la clave consumida y la carga útil transmitida permanece estable en todas las configuraciones.

Se demuestran estas relaciones, en la Tabla 16, donde el tamaño de paquete promedio enviado (206 bytes, para el caso de 100 bytes de aquí en adelante) se calcula con la división de los parámetros de salida de bytes enviados (153264) entre paquetes enviados (744). Se puede observar en la columna de sobrecarga, que este tamaño de paquete que se está produciendo en la simulación es 106 bytes superior al configurado, pudiendo comprobar que esto es así para todos los tamaños de paquete configurados.

Sabiendo esto, se puede calcular para cada tamaño de aplicación cuál es la sobrecarga total, multiplicando la sobrecarga (106 bytes) por los paquetes enviados registrados (744), con ello, restando este número a los bytes enviados que nos da el simulador (153264), se obtiene el dato de bytes útiles transmitidos, donde podemos comprobar en la tabla la poca variación entre los cuatro casos.

Siguiendo este análisis, también podemos comprobar que los bytes útiles calculados (74.400), son ligeramente parecidos al dato obtenido de clave consumida en bytes (74.700), esta pequeña diferencia (300 bytes) corresponde a un número de bytes no utilizado, que es exactamente la diferencia entre los paquetes enviados (744) y pares de clave consumidos (747), es decir, son exactamente los paquetes que se han simulado como no utilizados (3). Con esto se confirma que el consumo de clave es

directamente proporcional (1:1) al volumen de datos de la aplicación cuando se emplea la configuración OTP que se configuró.

Tamaño paq. (bytes)	Tamaño de paq. enviado (bytes)	Sobrecarga (bytes)	Sobrecarga total (bytes)	Bytes útiles	Clave consumida (bits)	Clave consumida (bytes)	Bits no utilizados	Paq. no utilizados
100	206	106	78.864	74.400	597.600	74.700	300	3
300	406	106	26.288	74.400	597.600	74.700	300	1
500	606	106	15.688	74.000	600.000	75.000	500	2
800	906	106	9.964	75.200	614.400	76.800	1600	2

Tabla 16. Relación de cálculos sobre los resultados obtenidos en la Prueba 1

5.3. Variantes de la primera prueba

Con relación a esta primera prueba se van a realizar dos modificaciones sobre el rendimiento del enlace QKD planteado, siendo la variante número 1 una modificación sobre el tamaño de clave, y la variante numero 2 una modificación sobre la tasa de generación de clave. Con el objetivo de identificar conjuntamente variaciones en el comportamiento de los resultados de la simulación de la primera prueba.

5.3.1. Variante 1 modificando tamaño de clave

Como se puede observar en la Tabla 17, se ha destacado el único parámetro de entrada modificado para la configuración del nuevo enlace QKD entre los mismos nodos 1-2. El tamaño de clave va a pasar a ser de 30.000 bits en vez de 10.000, lo equivalente a aumentar su tamaño por tres.

CONFIGURACIÓN PLANTEADA DEL ENLACE QKD						
Enlace QKD	Tasa de generación de clave [bps]	Tamaño clave [bits]	Tamaño paquete PP [bytes]	Tasa de postprocesado [bps]	Inicio [s]	Fin [s]
1-2	20.000	30.000	100	1000	0	100

Tabla 17. Variante 1 de configuración del enlace QKDNetSim para 2 nodos modificando tamaño de clave

5.3.1.1. Resultados de la variante 1

A continuación, se muestran las tablas de resultados correspondientes a esta simulación.

RESULTADOS DEL COMPORTAMIENTO DEL ENLACE				
ENLACE 1-2	**Tamaño del paquete de App**			
	100 bytes	**300 bytes**	**500 bytes**	**800 bytes**
Tamaño promedio de los pares de claves consumidos (bits)	0	0	0	0
Tamaño promedio de los pares de claves generados (bits)	240000	240000	240000	240000
Pares de claves consumidos	474	159	96	54
Pares de claves consumidos (bits)	379200	381600	384000	345600
Pares de claves generados	8	8	8	8
Pares de claves generados (bits)	1920000	1920000	1920000	1920000
Pares de claves retransmitidos	0	0	0	0
Pares de claves retransmitidos (bits)	0	0	0	0
Distancia del enlace (metros)	4957	4957	4957	4957
Capacidad del búfer QKD (bits)	50.000.000	50.000.000	50.000.000	50.000.000

Tabla 18. Resultados del comportamiento del enlace 1-2 de variante 1

COMPARATIVA DE RESULTADOS DE SIMULACIONES					
APP 1-2		**Tamaño del paquete de App**			
		100 bytes	**300 bytes**	**500 bytes**	**800 bytes**
Estadísticas de consumo de claves	Tamaño medio de los pares de claves consumidos (bits)	800	2400	4000	6400
	Pares de claves consumidos	474	159	96	54
	Pares de claves consumidos (bits)	379200	381600	384000	345600
Estadísticas de QKD Apps	Bytes recibidos	97026	64148	56964	46206
	Bytes enviados	97026	64148	56964	46206
	Porcentaje de utilización de claves/datos (%)	47,1	47,31	47	40,8
	Intentos de envío de paquetes perdidos (llamadas)	529	176	106	74
	Paquetes recibidos	471	158	94	51
	Paquetes enviados	471	158	94	51
Estadísticas de comunicación QKDApps-KMS	Bytes recibidos	312123	193825	171690	142804
	Bytes enviados	163114	56904	35715	22451
	Paquetes recibidos	324	114	72	45
	Paquetes enviados	325	114	72	46
Estadísticas de señalización	Bytes recibidos	127348	42718	25792	14508
	Bytes enviados	127348	42718	25792	14508
	Paquetes recibidos	316	106	64	36
	Paquetes enviados	316	106	64	36

Tabla 19. Comparativa de resultados de simulación de aplicación 1-2 de variante 1

5.3.2. Variante 2 modificando tasa de generación de clave

Como se puede observar en la Tabla 20, se ha destacado para este segundo caso el único parámetro de entrada modificado para la configuración del nuevo enlace QKD entre los mismos nodos 1-2. La tasa de generación de clave va a pasar a ser de 100.000 bps en vez de 20.000 bps, lo equivalente a aumentar su tamaño por cinco.

CONFIGURACIÓN PLANTEADA DEL ENLACE QKD						
Enlace QKD	**Tasa de generación de clave [bps]**	**Tamaño clave [bits]**	**Tamaño paquete PP [bytes]**	**Tasa de postprocesado [bps]**	**Inicio [s]**	**Fin [s]**
1-2	100.000	10.000	100	1000	0	100

Tabla 20. Variante 2 de configuración del enlace QKDNetSim para 2 nodos modificando tasa de generación de clave

5.3.2.1. Resultados de la variante 2

A continuación, se muestran las tablas de resultados ofrecidas por QKDNetSim correspondientes a este sistema QKD.

RESULTADOS DEL COMPORTAMIENTO DEL ENLACE				
ENLACE 1-2	**Tamaño del paquete de App**			
	100 bytes	**300 bytes**	**500 bytes**	**800 bytes**
Tamaño promedio de los pares de claves consumidos (bits)	0	0	0	0
Tamaño promedio de los pares de claves generados (bits)	80.000	80.000	80.000	80.000
Pares de claves consumidos	1002	336	201	126
Pares de claves consumidos (bits)	801600	806400	804000	806400
Pares de claves generados	124	124	124	124
Pares de claves generados (bits)	9920000	9920000	9920000	9920000
Pares de claves retransmitidos	0	0	0	0
Pares de claves retransmitidos (bits)	0	0	0	0
Distancia del enlace (metros)	4957	4957	4957	4957
Capacidad del búfer QKD (bits)	50.000.000	50.000.000	50.000.000	50.000.000

Tabla 21. Resultados del comportamiento del enlace 1-2 de variante 2

COMPARATIVA DE RESULTADOS DE SIMULACIONES					
APP 1-2		Tamaño del paquete de App			
		100 bytes	300 bytes	500 bytes	800 bytes
Estadísticas de consumo de claves	Tamaño medio de los pares de claves consumidos (bits)	800	2400	4000	6400
	Pares de claves consumidos	1002	336	201	126
	Pares de claves consumidos (bits)	801600	806400	804000	806400
Estadísticas de QKD Apps	Bytes recibidos	205794	135198	120594	112344
	Bytes enviados	205794	135198	120594	112344
	Porcentaje de utilización de claves/datos (%)	99,9	99,71	99,5	99,2
	Intentos de envío de paquetes perdidos (llamadas)	1	1	1	1
	Paquetes recibidos	999	333	199	124
	Paquetes enviados	999	333	199	124
Estadísticas de comunicación QKDApps-KMS	Bytes recibidos	652631	402407	351944	324189
	Bytes enviados	337082	113418	68013	42788
	Paquetes recibidos	669	225	135	85
	Paquetes enviados	669	225	135	85
Estadísticas de señalización	Bytes recibidos	269204	90272	54002	33852
	Bytes enviados	269204	90272	54002	33852
	Paquetes recibidos	668	224	134	84
	Paquetes enviados	668	224	134	84

Tabla 22. Comparativa de resultados de simulación de aplicación 1-2 de variante 2

5.3.3. Conclusiones

El comportamiento de las pruebas sigue el mismo patrón descrito en los anteriores apartados, pero los cambios introducidos han modificado datos significativos como los siguientes:

- Aumentar el tamaño de clave a bloques a un tamaño tres veces mayor, consigue que el tamaño promedio de par generado también se ha multiplicado por tres, de 80.000 bits a 240.000 bits. Por lo tanto, tiene sentido que, al no haber variado el marco general de la configuración, los pares generados acaben siendo tres veces menos, de 24 a 8, manteniendo en total el mismo valor de bits de clave generado a 1.920.000 porque la tasa de generación y el tiempo de simulación son los mismos solo que se distribuyen en menos bloques. Esto provoca a grandes rasgos, que la aplicación no consiga aprovechar la misma cantidad de clave, al ser los bloques demasiado grandes con relación a la demanda de la aplicación, con respecto a la primera prueba los pares consumidos han bajado en torno al 36% y los bits consumidos ahora rondan los 370.000 en vez de 600.000 bits. Lo mismo ocurre con los

parámetros de salida de la aplicación, donde se destaca que la utilización ha bajado considerablemente y por ello los intentos de envío suben considerablemente.

- En cuanto a la variante que aumenta la tasa de generación a un valor cinco veces el original, vemos como el tamaño promedio de pares generados se mantiene a 80.000 bits, pero en este caso, los pares generados aumentan en esas cinco veces (pasando de 24 a 124), reflejándose igual en los bits de clave generados con ahora 9.920.000 bits. Esto consigue que el enlace ofrezca más clave disponible y la aplicación pueda aprovecharse de ello, se puede ver como los pares consumidos han aumentado en torno a un 34% y los bits consumidos ahora rondan los 800.000 bits. Teniendo en cuenta que los parámetros de demanda de la aplicación no se modificaron, los resultados de la aplicación muestran comportamientos casi ideales con una utilización de más de 99% casi sin intentos fallidos.

En las siguientes figuras, se encuentran diversas gráficas mostrando estos comportamientos de los parámetros destacados.

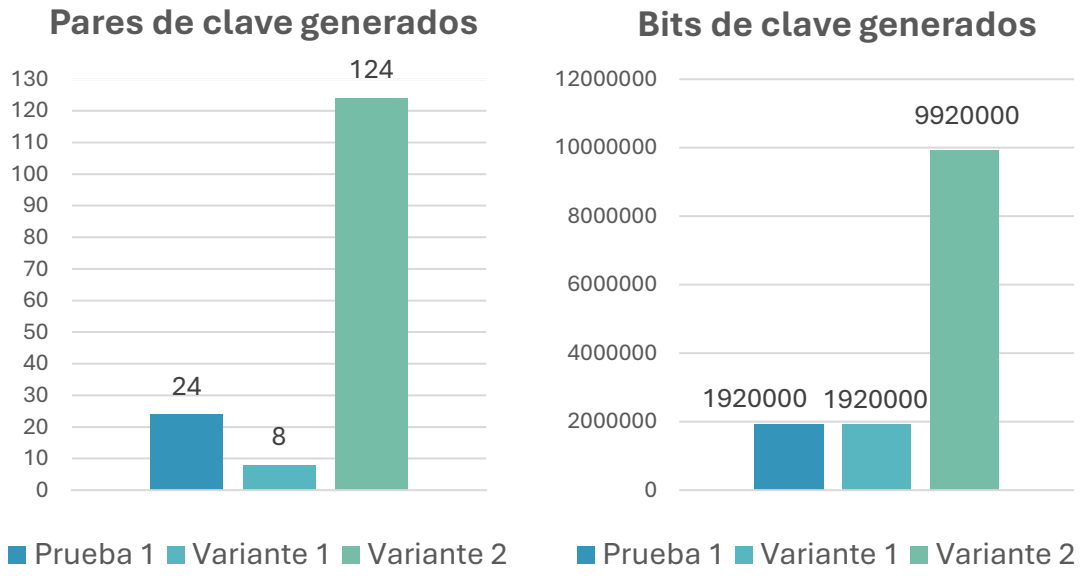


Figura 49. Gráficas comparativas de Prueba 1 y sus variantes para la clave generada

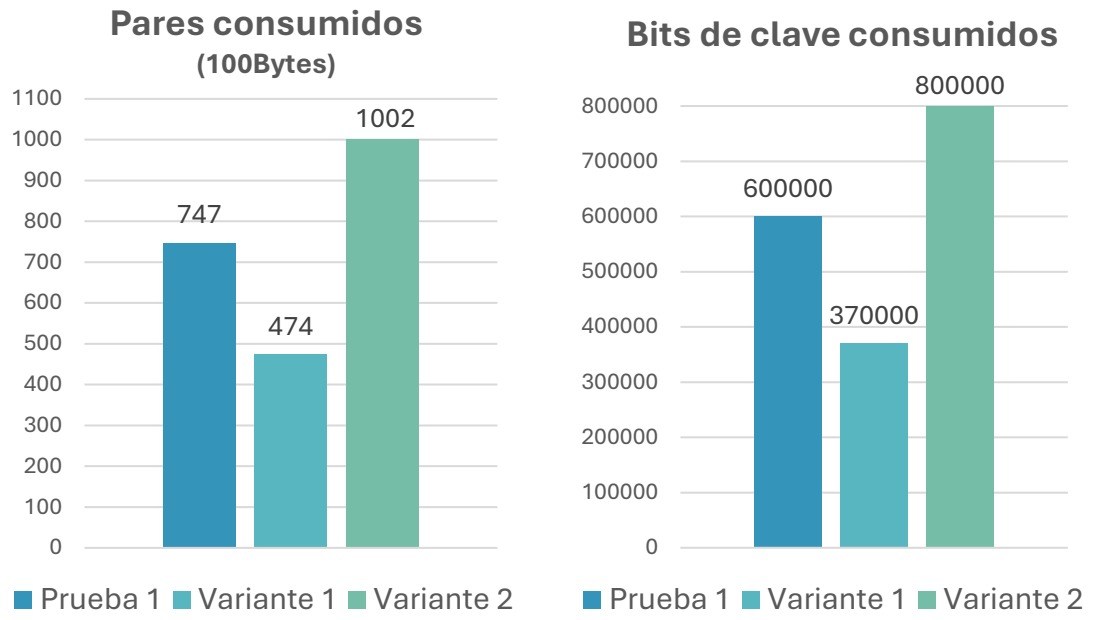


Figura 50. Gráficas comparativas de Prueba 1 y sus variantes para la clave consumida

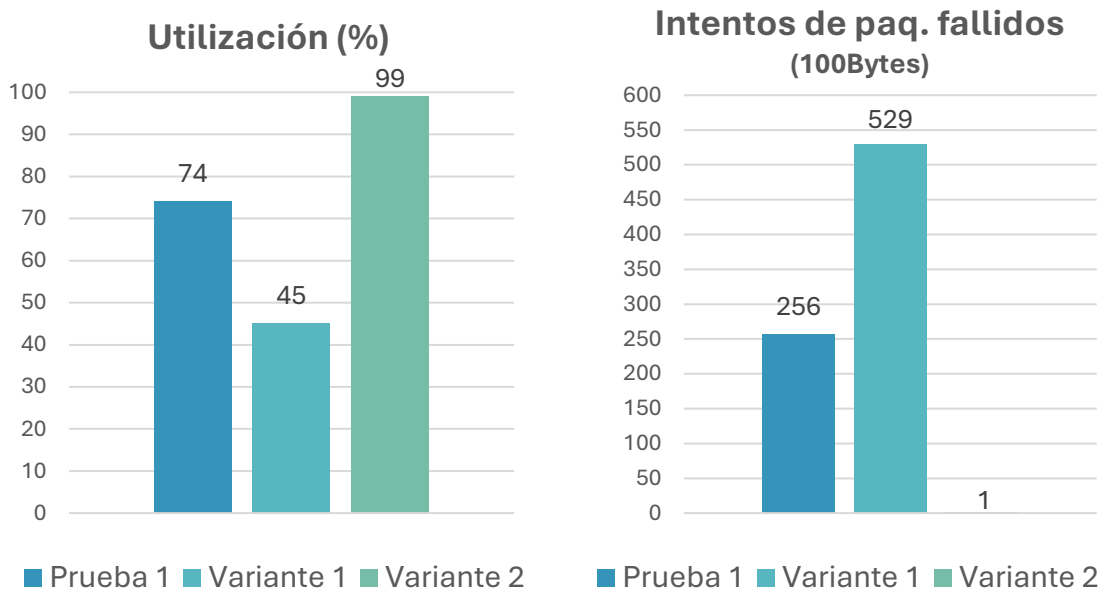


Figura 51. Gráficas comparativas de Prueba 1 y sus variantes para utilización e intentos de paquete fallidos

5.4. Segunda prueba simulada con 3 nodos

A continuación, se muestra la topología de la red para tres nodos unidos por dos enlaces QKD, denominados 1-2 y 2-3, junto con una aplicación criptográfica configurada entre los nodos 1 y 3. En este caso, entran en juego las retransmisiones de clave, donde el nodo intermedio debe realizar un proceso de retransmisión de clave fiable entre los nodos conectados en los extremos de sus dos enlaces.

El simulador ahora marca el enlace de aplicación como si omitiese al nodo 2, pero no es más que una representación lógica del enlace entre las aplicaciones criptográficas, que se comunican independientemente.

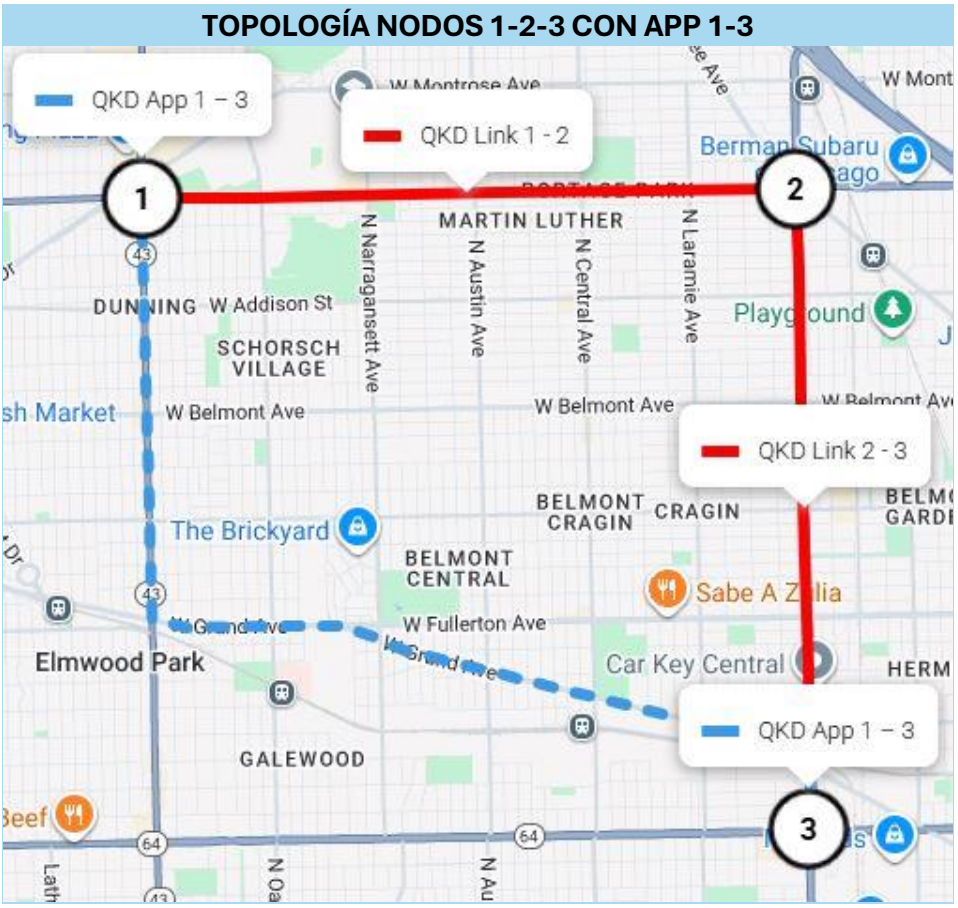


Tabla 23. Topología de red QKDNetSim con 3 nodos para la 2ª Prueba

5.4.1. Configuración del sistema

Se configuran los parámetros de entrada para esta segunda prueba siguiendo el mismo patrón que en la primera prueba.

CONFIGURACIÓN PLANTEADA DE LOS ENLACES QKD						
Enlace QKD	**Tasa de generación de clave [bps]**	**Tamaño clave [bits]**	**Tamaño paquete PP [bytes]**	**Tasa de postprocesado [bps]**	**Inicio [s]**	**Fin [s]**
1-2	20.000	10.000	100	1000	0	100
2-3	20.000	10.000	100	1000	0	100

Tabla 24. Configuración del enlace QKDNetSim para 3 nodos para la 2ª Prueba

CONFIGURACIÓN PLANTEADA DE APP CRIPTOGRÁFICA								
Par App.	**Tasa de tráfico de App [bps]**	**Tamaño del paquete de App [bytes]**	**Interfaz**	**Nº claves por solicitud**	**Autenticación**	**Cifrado**	**Inicio [s]**	**Fin [s]**
1-3	20.000	100	014	3	SHA2	OTP	10	50
		300						
		500						
		800						

Tabla 25. Configuración planteada de la App criptográfica para 2 nodos para 2ª Prueba

5.4.2. Resultados

A continuación, se muestran las tablas de resultados ofrecidas por QKDNetSim correspondientes a este sistema QKD.

RESULTADOS DEL COMPORTAMIENTO DEL ENLACE				
ENLACE 1-2 y 2-3	**Tamaño del paquete de App**			
	100 bytes	**300 bytes**	**500 bytes**	**800 bytes**
Tamaño promedio de los pares de claves consumidos (bits)	0	0	0	0
Tamaño promedio de los pares de claves generados (bits)	80.000	80.000	80.000	80.000
Pares de claves consumidos	0	0	0	0
Pares de claves consumidos (bits)	0	0	0	0
Pares de claves generados	24	24	24	24
Pares de claves generados (bits)	1.920.000	1.920.000	1.920.000	1.920.000
Pares de claves retransmitidos	1520	1520	780	420
Pares de claves retransmitidos (bits)	778240	778240	399360	215040
Distancia del enlace (metros)	4953	4953	4953	4953
Capacidad del búfer QKD (bits)	50.000.000	50.000.000	50.000.000	50.000.000

Tabla 26. Resultados del comportamiento del enlace 1-2 y 2-3

COMPARATIVA DE RESULTADOS DE SIMULACIONES					
APP 1-3		Tamaño del paquete de App			
		100 bytes	300 bytes	500 bytes	800 bytes
Estadísticas de consumo de claves	Tamaño medio de los pares de claves consumidos (bits)	800	2400	4000	6400
	Pares de claves consumidos	927	309	96	30
	Pares de claves consumidos (bits)	741600	741600	384000	192000
Estadísticas de QKD Apps	Bytes recibidos	190344	125048	56964	27180
	Bytes enviados	190344	125048	56964	27180
	Porcentaje de utilización de claves/datos (%)	92,4	92,22	47	24
	Intentos de envío de paquetes perdidos (llamadas)	76	26	106	95
	Paquetes recibidos	924	308	94	30
	Paquetes enviados	924	308	94	30
Estadísticas de comunicación QKDApps-KMS	Bytes recibidos	604265	370559	171490	82488
	Bytes enviados	322312	104768	35715	15672
	Paquetes recibidos	620	208	72	32
	Paquetes enviados	620	208	72	33
Estadísticas de señalización	Bytes recibidos	249054	83018	25792	8060
	Bytes enviados	249054	83018	25792	8060
	Paquetes recibidos	618	206	64	20
	Paquetes enviados	618	206	64	20

Tabla 27. Comparativa de resultados de simulación de aplicación 1-3

5.4.3. Conclusiones

El resultado de ambos enlaces es el mismo y se representa en la Tabla 26, podemos ver que cada enlace sigue generando 24 pares de 80.000 bits generando 1.920.000 bits en total, al igual que en el resultado de la primera prueba por tener la misma configuración de enlace. Sin embargo, en este caso se indica que los pares consumidos en el enlace son 0, lo cual corresponde a lo esperado ya que ahora el consumo ha de indicarse de extremo a extremo y no puede aparecer directamente en el enlace.

Para esta prueba ahora sí aparecen datos en el parámetro de salida de pares de claves retransmitidos, estos corresponden al tráfico interno necesario para coordinar las operaciones entre los buffers y los KMS, de tal manera que, a mayor tamaño de paquete de aplicación, menos pares deben retransmitirse al encapsularse más bits de aplicación en cada operación de retransmisión, demostrando así que el nodo 2 actúa como punto de retransmisión entre los extremos. Llama la atención que el dato de retransmisiones sea tan elevado en comparación con los pares de

claves consumidos que aparecen en la Tabla 27, esto se debe a las operaciones de retransmisión que se producen en el nodo intermedio.

El consumo de clave decae al simular los paquetes de aplicación de mayor tamaño con 500 y 800 bytes, a diferencia del comportamiento visto para dos nodos en la primera prueba. Esto indica que la retransmisión por el nodo intermedio penaliza más a los paquetes grandes que a los pequeños, por lo que para esta configuración los paquetes pequeños logran garantizar más continuidad por el paso intermedio y se ven menos penalizados por él.

La utilización de clave para los dos tamaños de paquetes de 100 y 300 bytes es muy alta entorno al 92%, incluso mejorando el porcentaje de la primera prueba, lo que muestra que, en este caso la retransmisión ayuda a estabilizar el flujo de clave. Sin embargo, la utilización decae notablemente para los paquetes más grandes, llegando a niveles del 47% y 24%, como se puede ver en la gráfica de la Figura 52. Con relación a este comportamiento, los intentos de envío de paquete también son desfavorables para los de mayor tamaño, resaltando que el sistema no consigue que dichos paquetes se retransmitan de forma eficiente.

Respecto a la comunicación QKDApp-KMS y a la señalización, tanto los paquetes como los bytes siguen el mismo comportamiento, son altos para paquetes pequeños y bajos para paquetes grandes, por lo que el nodo de retransmisión sigue penalizando también estas comunicaciones.

Este comportamiento demuestra que los paquetes que requieren de clave para paquetes pequeños tienen un mejor comportamiento, con un alto nivel de utilización de clave, pero entre las dos opciones, elegir la de 300 bytes resultaría en una mejor optimización del tráfico general de los procesos al reducir la carga general de aplicación, de comunicación entre aplicación-KMS y de señalización.

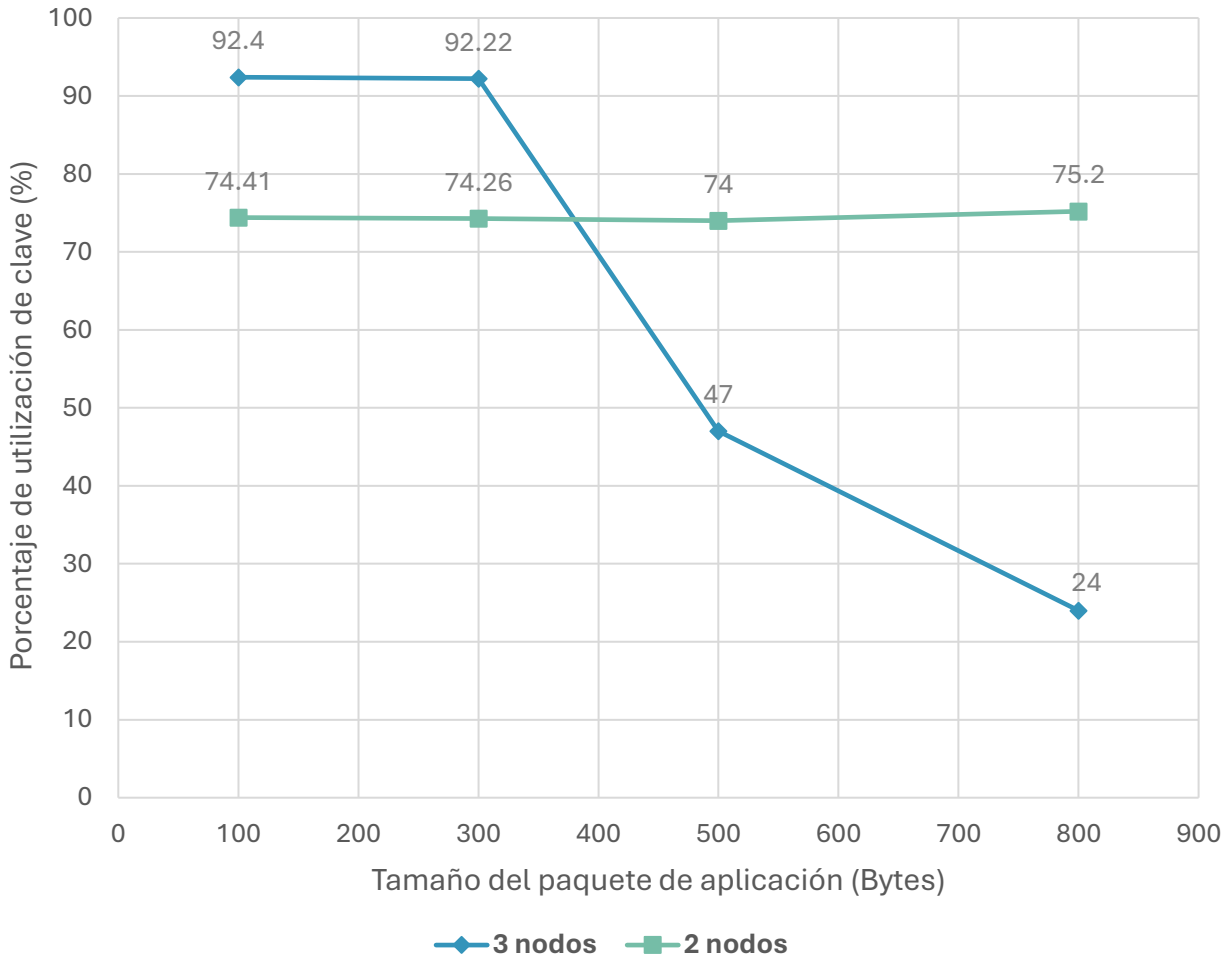


Figura 52. Gráfica comparativa de la eficiencia de utilización de clave para las configuraciones de la 1ª Prueba (2 nodos) y la 2ª Prueba (3 nodos)

5.5. Conclusiones

El propósito general del simulador es el desarrollo de estudios de gestión de clave en redes QKD con la posibilidad de configurar nodos como repetidores de confianza bajo las interfaces adaptadas de ETSI 014 y ETSI 004. El simulador no implementa directamente la propagación cuántica ni la atenuación óptica ya que el canal cuántico se abstrae. Se observa que al fijar la tasa de clave ("*KeyRate*") la simulación ya no depende de la distancia entre nodos, sino de un valor fijo configurado para ese escenario. Esto significa que, aunque aumente la distancia entre nodos en la topología de la configuración, no variarán los resultados simulados.

Si se quiere modelar la caída de tasa por distancia, el simulador indica que se debe de introducirse manualmente ajustando los parámetros de la tasa de clave deseada

simulando una mejor o peor tasa, o bajo la opción donde el simulador calcular su tasa de manera automática en base a parámetros no conocidos.

En este caso el simulador para cumplir con su propósito, asume que se pueden alcanzar las tasas de generación de clave expuestas, mientras que si se quiere otro tipo de análisis más profundo y realista de los parámetros de un sistema QKD, se debe profundizar más en aspectos teóricos como el artículo propuesto por Attema et al. [49], que expone como estos datos están intrínsecamente limitados por parámetros óptimos que requieren un cálculo complejo si se tuvieran en cuenta por ejemplo los canales cuánticos con ruido.

Las pruebas bajo estudio en este capítulo principalmente han mostrado el siguiente comportamiento.

- **1ª Prueba con dos nodos:** Se observa que, al variar el tamaño de los paquetes de aplicación el número de pares de claves consumidos varía, pero el consumo total en bits se mantiene prácticamente constante. Esto demuestra estabilidad en el consumo de clave y confirma que el rendimiento está limitado por la propia configuración del sistema. También se comprueba que el simulador procesa los paquetes y que el consumo que hace con OTP cumple con su comportamiento esperado.

    - **Variante 1 de la prueba 1:** Aumentar el tamaño de clave reduce el aprovechamiento por parte de la aplicación, porque los bloques grandes generan un promedio de pares de claves mayor, que hace bajar el porcentaje de utilización de clave de la aplicación para el uso configurado al tardar en ser atendidas las solicitudes de la aplicación por le buffer.
    - **Variante 2 de la prueba 1:** Aumentar la tasa de generación de clave mejora el rendimiento al generar más claves disponibles rápidamente permitiendo un mejor consumo por parte de la aplicación.

- **2ª Prueba con tres nodos:** El nodo intermedio funciona como retransmisor de claves. Aparece consumo indirecto debido a los pares retransmitidos y se ve que los paquetes grandes sufren más penalización que los pequeños, al requerir de un suministro de clave que no es proporcionado eficientemente por la red simulada, resultando en bajos porcentajes de utilización de clave.

Capítulo 6

6. Conclusión y líneas futuras

6.1. Conclusión

La investigación en distribución cuántica de claves (QKD), suele centrarse en el perfeccionamiento de equipo y en la mejora de las velocidades y distancias alcanzables mediante dispositivos y canales ópticos, con un gran enfoque en aspectos experimentales del hardware. En contraste, este proyecto ha planteado en un primer lugar, un estudio de los fundamentos teóricos que establecen las bases de la cuántica detrás de QKD, para posteriormente introducir cómo se ve aplicada en los elementos necesarios que componen el funcionamiento de una red QKD en su conjunto, analizando aspectos como las implementaciones disponibles, los protocolos existentes (en especial BB84), el proceso de generación y retransmisión de claves, los componentes de la estructura de capas, así como los esfuerzos de estandarización y la organización de la red.

Con este enfoque teórico, se abstraen tanto la complejidad física como las posibles vulnerabilidades del diseño de los equipos, permitiendo centrar la atención en la lógica del sistema. Para ello, se presenta el simulador en línea QKDNetSim, del cual se describen sus principales características antes de someterlo a experimentación práctica. En esta fase, se implementaron diversos casos de configuración predefinidos con el objetivo de analizar los resultados obtenidos y evaluar el comportamiento del sistema.

En conjunto, QKDNetSim implementa de forma efectiva las funciones básicas de un sistema QKD aunque omite la modelización de aspectos como la validación, el control de acceso, la calidad de servicio (QoS) y la gestión integral de la red cuántica. Por ello, se posiciona como un simulador enfocado al análisis del comportamiento de red y la administración de recursos de clave, más que a la emulación detallada de los fenómenos físicos subyacentes a los protocolos QKD. Las simulaciones han permitido identificar tanto las condiciones que favorecen un rendimiento óptimo como las limitaciones que deben abordarse. Los resultados evidencian la sensibilidad de la red frente a los parámetros de configuración, y consolidan el valor del simulador como herramienta académica para explorar y comprender la configuración y el comportamiento de sistemas QKD.

6.2. Líneas futuras

Como líneas futuras a partir del trabajo realizado, se identifican existen múltiples líneas de investigación teórica que permiten profundizar en el estudio de la criptografía cuántica como:

- Uno de los enfoques más relevantes es el análisis del entrelazamiento cuántico como recurso fundamental para la distribución de claves. Especialmente en protocolos como E91 o esquemas basados en teleportación cuántica, que superan limitaciones del modelo BB84. Permiten estudiar la seguridad desde una perspectiva más física, vinculada a la no-localidad y las desigualdades de Bell.
- Estudio que abordan desafíos clave en la seguridad, eficiencia y resistencia frente a ataques como los protocolos MDI-QKD ("*Measurement-Device-Independent*") o DI-QKD ("*Device-Independent*").

En el plano práctico, se plantea como línea futura el contacto directo con los desarrolladores de QKDNetSim con el objetivo de acceder a documentación técnica ampliada, configuraciones avanzadas o funcionalidades no disponibles públicamente. Otro campo de estudio para pruebas en el simulador podría basarse en:

- Estudiar el simulador con la tasa de clave automáticamente calculada, evaluar su comportamiento y a qué se debe su cálculo.
- Un análisis detallado respecto al procedimiento detallado que simula el enlace para la generación, consumo y retransmisión de claves que refuten los resultados que se obtienen de salida en base a las gráficas del simulador.
- Ampliar las simulaciones a redes más complejas con múltiples nodos.
- Configurar redes compuestas por enlaces de distintos rendimientos para ver cómo afecta a las aplicaciones configuradas sobre ellos.
- Configurar sobre una red QKD una red de distintas aplicaciones, de manera que al solaparse entre ellas ver cómo afecta al rendimiento de la simulación si se congestionan los enlaces.

Bibliografía

[1] M. Mehic *et al.*, "Quantum Key Distribution: A Networking Perspective," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–41, Sept. 2021, doi: 10.1145/3402192.

[2] T. G. Wong, *Introduction to classical and quantum computing*. Omaha, Nebraska: Rooted Grove, 2022.

[3] R. Van Meter, *Quantum networking*. in Networks and telecommunications series. London : Hoboken, NJ: ISTE ; Wiley, 2014.

[4] M. Golec, E. S. Hatay, S. S. Gill, Y. Mao, and R. Buyya, "Quantum computing at a glance," in *Quantum Computing*, Elsevier, 2025, pp. 3–18. doi: 10.1016/B978-0-443-29096-1.00010-6.

[5] C. J. Hoofnagle and S. L. Garfinkel, "Law and Policy for the Quantum Age," 2022, doi: 10.1017/9781108883719.

[6] V. Vasani, K. Prateek, R. Amin, S. Maity, and A. D. Dwivedi, "Embracing the quantum frontier: Investigating quantum communication, cryptography, applications and future directions," *J. Ind. Inf. Integr.*, vol. 39, p. 100594, May 2024, doi: 10.1016/j.jii.2024.100594.

[7] Pal, O., Jain, M., Murthy, B.K. and Thakur, V., "Cyber Security and Digital Forensics - 2022 - Ghonge - Quantum and Post-Quantum Cryptography-Criptografía cuántica y postcuántica," pp. 45–58, 2022, [Online]. Available: https://onlinelibrary.wiley.com/doi/chapter-epub/10.1002/9781119795667.ch2

[8] R. Alléaume *et al.*, "Using quantum key distribution for cryptographic purposes: A survey," *Theor. Comput. Sci.*, vol. 560, pp. 62–81, Dec. 2014, doi: 10.1016/j.tcs.2014.09.018.

[9] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, Philadelphia, Pennsylvania, United States: ACM Press, 1996, pp. 212–219. doi: 10.1145/237814.237866.

[10] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949, doi: 10.1002/j.1538-7305.1949.tb00928.x.

[11] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-04101-3.

[12] National Institute of Standards and Technology (US), "Secure hash standard," National Institute of Standards and Technology (U.S.), Washington, D.C., NIST FIPS 180-4, 2015. doi: 10.6028/NIST.FIPS.180-4.

[13] National Institute of Standards and Technology (US), "SHA-3 standard : permutation-based hash and extendable-output functions," National Institute of Standards and Technology (U.S.), Washington, D.C., 2015. doi: 10.6028/NIST.FIPS.202.

[14] W. Dai and T. Krovetz, "VHASH Security," 2007, 2007/338. [Online]. Available: https://eprint.iacr.org/2007/338

[15] M. N. Wegman and J. L. Carter, "New hash functions and their use in authentication and set equality," *J. Comput. Syst. Sci.*, vol. 22, no. 3, pp. 265–279, June 1981, doi: 10.1016/0022-0000(81)90033-7.

[16] T. Krovetz, "Message Authentication on 64-Bit Architectures," in *Selected Areas in Cryptography*, vol. 4356, E. Biham and A. M. Youssef, Eds., in Lecture Notes in Computer Science, vol. 4356. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 327–341. doi: 10.1007/978-3-540-74462-7_23.

[17] S. K. Sehgal and R. Gupta, "Quantum Cryptography and Quantum Key," in *2021 International Conference on Industrial Electronics Research and Applications (ICIERA)*, Dec. 2021, pp. 1–5. doi: 10.1109/ICIERA53202.2021.9726722.

[18] "GS QKD 014 - V1.1.1 - Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API," *ETSI Ind. Specif.*, Feb. 2019, [Online]. Available: http://www.etsi.org/standards-search

[19] Y. Cao, Y. Zhao, Q. Wang, J. Zhang, S. X. Ng, and L. Hanzo, "The Evolution of Quantum Key Distribution Networks: On the Road to the Qinternet," *IEEE Commun. Surv. Tutor.*, vol. 24, no. 2, pp. 839–894, 2022, doi: 10.1109/COMST.2022.3144219.

[20] A. Ekert and A. Kay, "Quantum Communication Experiments with Discrete Variables," in *Quantum Information: From Foundations to Quantum Technology Applications*, Wiley, 2019, pp. 401–435. doi: 10.1002/9783527805785.ch19.

[21] E. Dervisevic *et al.*, "Quantum Key Distribution Networks - Key Management: A Survey," *ACM Comput. Surv.*, vol. 57, no. 10, pp. 1–36, Oct. 2025, doi: 10.1145/3730575.

[22] V. Martin *et al.*, "Quantum technologies in the telecommunications industry," *EPJ Quantum Technol.*, vol. 8, no. 1, pp. 1–31, Dec. 2021, doi: 10.1140/epjqt/s40507-021-00108-9.

[23] S. Pirandola, "End-to-end capacities of a quantum communication network," *Commun. Phys.*, vol. 2, no. 1, p. 51, May 2019, doi: 10.1038/s42005-019-0147-3.

[24] Y.-A. Chen *et al.*, "An integrated space-to-ground quantum communication network over 4,600 kilometres," *Nature*, vol. 589, no. 7841, pp. 214–219, Jan. 2021, doi: 10.1038/s41586-020-03093-8.

[25] M. Mehic, S. Rass, P. Fazio, and M. Voznak, *Quantum Key Distribution Networks: A Quality of Service Perspective*. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-031-06608-5.

[26] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, "Secure quantum key distribution with realistic devices," *Rev. Mod. Phys.*, vol. 92, no. 2, p. 025002, May 2020, doi: 10.1103/RevModPhys.92.025002.

[27] R. Kumar, H. Qin, and R. Alléaume, "Coexistence of continuous variable QKD with intense DWDM classical channels," *New J. Phys.*, vol. 17, no. 4, p. 043027, Apr. 2015, doi: 10.1088/1367-2630/17/4/043027.

[28] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theor. Comput. Sci.*, vol. 560, pp. 7–11, Dec. 2014, doi: 10.1016/j.tcs.2014.05.025.

[29] M. Mehic, O. Maurhart, S. Rass, D. Komosny, F. Rezac, and M. Voznak, "Analysis of the Public Channel of Quantum Key Distribution Link," *IEEE J. Quantum Electron.*, vol. 53, no. 5, pp. 1–8, Oct. 2017, doi: 10.1109/JQE.2017.2740426.

[30] S. K. Singh *et al.*, "Advancements in secure quantum communication and robust key distribution techniques for cybersecurity applications," *Cyber Secur. Appl.*, p. 100089, Mar. 2025, doi: 10.1016/j.csa.2025.100089.

[31] A. K. Ekert, “Quantum cryptography based on Bell’s theorem,” *Phys. Rev. Lett.*, vol. 67, no. 6, pp. 661–663, Aug. 1991, doi: 10.1103/PhysRevLett.67.661.
[32] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, “Experimental quantum cryptography,” *J. Cryptol.*, vol. 5, no. 1, pp. 3–28, Jan. 1992, doi: 10.1007/BF00191318.
[33] “SERIES Y.3800: Overview on networks supporting quantum key distribution.” Int. Telecommun. Union, 2020. [Online]. Available: https://www.itu.int/itu-t/recommendations/rec.aspx?rec=14257&lang=es
[34] Y. Luo, X. Cheng, H.-K. Mao, and Q. Li, “An Overview of Postprocessing in Quantum Key Distribution,” *Mathematics*, vol. 12, no. 14, p. 2243, Jan. 2024, doi: 10.3390/math12142243.
[35] “GS QKD 004 - V2.1.1 - Quantum Key Distribution (QKD); Application Interface,” *ETSI Ind. Specif.*, Aug. 2020, [Online]. Available: http://www.etsi.org/standards-search
[36] “Y.3803 : Quantum key distribution networks – Key management.” [Online]. Available: https://www.itu.int/rec/T-REC-Y.3803-202311-I!Amd1/en
[37] “GR QKD 007 - V1.1.1 - Quantum Key Distribution (QKD); Vocabulary,” *ETSI Ind. Specif.*, Dec. 2018, [Online]. Available: http://www.etsi.org/standards-search
[38] P. Burdiak *et al.*, “Use-Case Denial of Service Attack on Actual Quantum Key Distribution Nodes:,” in *Proceedings of the 9th International Conference on Information Systems Security and Privacy*, Lisbon, Portugal: SCITEPRESS - Science and Technology Publications, 2023, pp. 89–94. doi: 10.5220/0011672000003405.
[39] Y.-L. Tang *et al.*, “Measurement-Device-Independent Quantum Key Distribution over Untrustful Metropolitan Network,” *Phys. Rev. X*, vol. 6, no. 1, p. 011024, Mar. 2016, doi: 10.1103/PhysRevX.6.011024.
[40] M. Lucamarini, Z. L. Yuan, J. F. Dynes, and A. J. Shields, “Overcoming the rate–distance limit of quantum key distribution without quantum repeaters,” *Nature*, vol. 557, no. 7705, pp. 400–403, May 2018, doi: 10.1038/s41586-018-0066-6.
[41] “GS QKD 011 - V1.1.1 - Quantum Key Distribution (QKD); Component characterization: characterizing optical components for QKD systems, 2016-05.” Accessed: Aug. 19, 2025. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/QKD/001_099/011/01.01.01_60/gs_QKD011v010101p.pdf
[42] J. Li *et al.*, “Integration of Quantum Key Distribution Networks and Classical Networks: An Evolution Perspective,” *IEEE Netw.*, vol. 39, no. 3, pp. 180–187, May 2025, doi: 10.1109/MNET.2025.3537691.
[43] M. Mehic, E. Dervisevic, P. Burdiak, V. Lipovac, P. Fazio, and M. Voznak, “Emulation of Quantum Key Distribution Networks,” *IEEE Netw.*, vol. 39, no. 1, pp. 116–123, Jan. 2025, doi: 10.1109/MNET.2024.3398404.
[44] E. Dervisevic, M. Voznak, and M. Mehic, “Large-scale quantum key distribution network simulator,” *J. Opt. Commun. Netw.*, vol. 16, no. 4, p. 449, Apr. 2024, doi: 10.1364/JOCN.503356.
[45] M. Mehic, O. Maurhart, S. Rass, and M. Voznak, “Implementation of quantum key distribution network simulation module in the network simulator NS-3,” *Quantum Inf. Process.*, vol. 16, no. 10, p. 253, Aug. 2017, doi: 10.1007/s11128-017-1702-z.

[46] "OpenQKD - QKDNetSim - Quantum Key Distribution Network Simulator." [Online]. Available: https://www.open-qkd.eu/
[47] "13. Emulation Overview — Model Library." [Online]. Available: https://www.qkdnetsim.info/models/build/html/emulation-overview.html
[48] "Home, QKDNetSim (v2.0)." [Online]. Available: https://www.qkdnetsim.info/
[49] T. Attema, J. Bosman, and N. Neumann, "Optimizing the Decoy-State BB84 QKD Protocol Parameters," *Quantum Inf. Process.*, vol. 20, no. 4, p. 154, Apr. 2021, doi: 10.1007/s11128-021-03078-0.