Criptografía cuántica aplicada

Jesús Martínez Mateo

Abril de 2008

GIICC Grupo de Investigación en Información y Computación Cuántica
http://gcc.ls.fi.upm.es/
Laboratorio de Análisis Numérico, nº 5213.
Teléfono: (+34) 91 336 6938

Facultad de Informática - Universidad Politécnica de Madrid
Campus de Montegancedo s/n
28660 Boadilla del Monte (Madrid)

Proyecto realizado por:

Jesús Martínez Mateo
jmartinez@fi.upm.es

Bajo la dirección de:

Vicente Martín Ayuso
vicente@fi.upm.es

Versión del documento: 1.0
Fecha de la última modificación: 28 de abril de 2008.

Índice general

Lista de algoritmos

Índice de figuras

Índice de tablas

Parte I

Prólogo

Prefacio

Seguramente, en el contenido de este documento encuentre más información de la estrictamente necesaria para el desarrollo del proyecto. Es cierto que podría haber descartado parte de esa información, pero, siempre con miedo a perder la orientación principal, he intentado reunir todo aquello que ha estado relacionado con su desarrollo. Hay varias razones por las que he decidido incluir más información de la necesaria. En primer lugar, este proyecto requiere un esfuerzo especial debido a las exigencias que impone la utilización de una disciplina de la física, por lo general desconocida para un ingeniero, como es la mecánica cuántica. En ese esfuerzo, he intentado hacer una recopilación de toda la información que ha sido de interés para la elaboración de este proyecto. También he intentado poner especial cuidado en el desarrollo de un índice lo suficientemente descriptivo y estructurado como para poder estudiar sólo aquellos aspectos que se consideren necesarios. Por esta misma razón, si se lee este proyecto es su totalidad se podrá encontrar repetidos algunos conceptos. Esa reiteración en la exposición no tiene el mayor interés que la posible lectura autocontenida de algunas partes del proyecto; la lectura global ilustrará una misma idea desde varios puntos de vista, ligándo varios puntos de vista.

Finalmente, deseo añadir mi decisión de hablar en primera persona sólo en este pequeño prólogo, y utilizar el plural en el resto del documento, debido a que el esfuerzo para la realización este proyecto ha sido compartido por varias personas. Entre ellas, he de agradecer sinceramente la ayuda de Vicente Martín y Daniel Lancho.

Capítulo 1

Introducción

El mundo de la tecnología impregna cada día más la sociedad actual, hasta el punto de que siempre que utilizamos el término *información* lo relacionamos directamente con un ordenador o un dispositivo de almacenamiento. Aunque este uso sea válido, podemos ver que la información está asociada a muchos otros campos de la naturaleza. La genética o, como veremos en este proyecto, la física, son algunos de esos campos.

Podemos afirmar, en consecuencia, que la teoría de la información no es una disciplina exclusiva de la ingeniería, siendo también una herramienta fundamental para otras ramas de la ciencia. Pero existe algo todavía más interesante en la implementación física de la información, y es la posibilidad real de extender el átomo básico de información, *bit*, al *qubit*, una nueva magnitud que contiene a la antigua y evoluciona con nuevas reglas dictadas por la física cuántica. Su uso nos conduce a los ordenadores cuánticos y más concretamente a lo que será el tema de este proyecto: los sistemas de criptografía.

1.1. Un poco de historia

Comenzaremos estudiando la historia de la criptografía desde el momento en el que inicia su relación con la mecánica cuántica con el desarrollo del primer protocolo de distribución cuántica de claves (en lo sucesivo referido por sus iniciales inglesas: QKD, Quantum Key Distribution). Concluiremos con una exposición cronológica de los acontecimientos más importantes y sus consecuencias. En este estudio preliminar de la historia de la criptografía cuántica queremos destacar tres hechos. En primer lugar, el nacimiento de la distribución cuántica de claves como consecuencia de una idea brillante, ignorada durante mucho tiempo, y rescatada por dos autores que la adaptan a otro entorno: la distribución de claves. Aunque estos dos personajes pasarán a la historia por la descripción del primer protocolo, al que aportan su nombre, muchas veces es ignorada la importancia del enorme esfuerzo que dedican en sus trabajos posteriores a completar el desarrollo de un sistema que implemente este protocolo. Producirán el primer prototipo de QKD, años antes de que la criptografía convencional se vea en peligro por las consecuencias de la computación cuántica.

Dinero cuántico

Tenemos que remontarnos a finales de los años sesenta y comienzos de los setenta para conocer los inicios de la criptografía cuántica. Es entonces cuando un estudiante de grado de la universidad de Columbia intenta publicar una idea a la que llamó *dinero cuántico*. El estudiante fue *Stephen Wiesner*, y aunque su idea era revolucionaria, nunca fue tomada en serio por la comunidad científica. Permaneció en el anonimato durante algo más de una década, hasta que finalmente fue publicada en el año 1983 [5], ¡13 años después de su concepción!. Antes había conseguido captar el interés de un amigo y antiguo estudiante de su misma universidad, *Charles H. Bennett*, quién la adaptaría para desarrollar el primer protocolo de distribución cuántica de claves.

La idea de Wiesner describe un mecanismo para crear dinero imposible de falsificar. Incorporando un total de 20 trampas de luz en los billetes de un dolar, y codificando en cada una de ellas uno de dos posibles valores con un fotón polarizado. Wiesner pretendía obtener una huella identificativa para cada billete. La seguridad de dicha huella reside en el hecho de que el fotón contenido en cada trampa de luz es polarizado con respecto a una base, y sólo con el conocimiento de esa base se puede recuperar el estado de polarización correcto de cada fotón sin ninguna posiblidad de error. Si no se conoce la base el resultado del proceso de lectura es completamente aleatorio, produciendo con igual probabilidad el resultado correcto o el incorrecto. De esta forma, sólo la entidad que codificara cada billete podría saber, con un margen de error de 1 entre 2^{20} posibilidades, cuál es el número correcto.

El primer protocolo de distribución cuántica de claves

Un año después de la publicación de Wiesner, en 1984, Charles H. Bennett (del centro de investigación Thomas J. Watson de IBM) y Gilles Brassard (de la universidad de Montreal, en Canadá), definen el que será el primer protocolo de criptografía, BB84, basado en los principios de una disciplina relativamente moderna de la física, la mecánica cuántica [6]. Dicho protocolo constituye el primer diseño práctico de un sistema criptográfico cuántico, y por esta razón, suele arrebatar el puesto que debería ocupar en la historia la idea de S. Wiesner.

En él se establecen los dos primeros niveles de trabajo de un sistema de QKD: el intercambio de clave en bruto y la reconciliación de bases, que en un sistema ideal, libre de errores, son suficientes para realizar el intercambio de una clave. En la práctica, su implementación se ve afectada por imprecisiones, ruido, una actuación malintencionada, etc. Por ello, continúan con la definición formal de los siguientes niveles de desarrollo: la corrección de errores y la amplificación de la privacidad; completando así los cuatro primeros niveles de la arquitectura que se utiliza actualmente en la implementación de sistemas de QKD.

1.1.1. Cronología

Es habitual ligar la criptografía cuántica a los ordenadores cuánticos. Esto es así debido al enorme impacto que tuvo el descubrimiento del algoritmo de Shor,

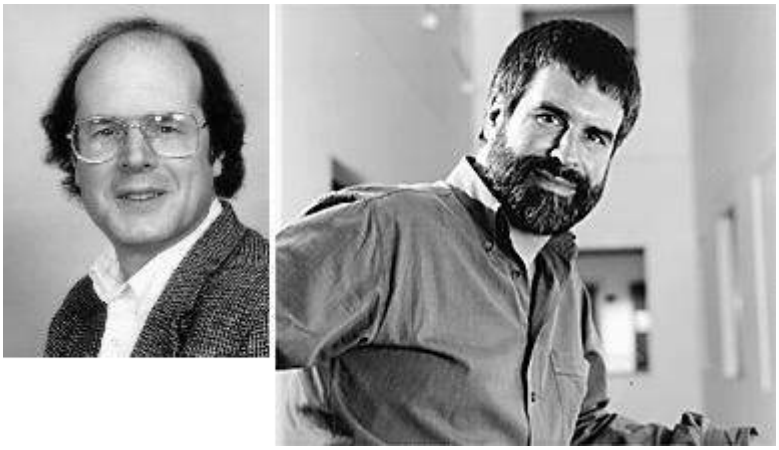


Figura 1.1: Charles H. Bennett (izquierda) y Gilles Brassard (derecha).

que reduce el coste computacional de la factorización de números enteros, eliminando de un golpe la seguridad teórica de métodos criptográficos convencionales como el RSA. Una vez eliminada la competencia convencional se introduce un nuevo mecanismo de distribución de claves, teóricamente perfecto e inmune a cualquier tipo de ataque computacional, que se encargaría de sustituir los procedimientos actuales de cifrado asimétrico. En realidad la historia no ocurre exactamente así: QKD surge a partir de una idea brillante varios años antes de la propuesta del algoritmo de Shor.

La cronología de los hechos más importantes, tanto en criptografía como en computación cuántica, se muestra a continuación:

1970 S. Wiesner propone el "dinero cuántico". Este concepto no es publicado hasta 1983, pero en algún instante desconocido de ese lapso de tiempo, Wiesner comparte su idea con Bennett, que utilizará en la elaboración del primer protocolo de QKD.

1982 R. Feynman idealiza el ordenador cuántico.

1983 S. Wiesner consigue publicar su idea de "dinero cuántico" [5].

1984 C. H. Bennett y G. Brassard proponen el primer protocolo para la QKD, que se conocerá con el nombre de BB84 [6].

1988 Bennett y Brassard utilizan la amplificación de la privacidad en los sistemas de criptografía cuántica, con el objetivo de reducir la información de la clave intercambiada que puede poseer un hipotético espía.

1989 Bennett y J. Smolin construyen, en los laboratorios del centro de investigación T.J. Watson de IBM, el primer prototipo que implementa un protocolo de QKD. Esta prueba de concepto dispara un interés generalizado sobre QKD.

1991 A. Ekert propone un nuevo protocolo para la distribución de claves, basado en pares de partículas entrelazadas (pares EPR) [10].

1992 Aparece la primera descripción del algoritmo de Shor. Sus implicaciones constituyen un riesgo importante para los sistemas de cifrado asimétricos, basados en la criptografía de clave pública.

1994 Brassard y Salvail completan un nuevo mecanismo para la corrección de errores, y proponen una implementación práctica: *Cascade*.

1998 Se construye el primer ordenador cuántico de 2 qubits en la universidad de California.

Como vemos en el cronograma mostrado, Bennett y Brassard completan la arquitectura básica de un sistema de QKD prácticamente antes de que Shor defina el algoritmo que recibe su nombre. Luego debemos ver QKD como un mecanismo de distribución de claves sencillo, que surge a partir de una idea brillante, que resulta físicamente factible y posee una enorme seguridad intrínseca. Sólo posteriormente se presenta como alternativa frente a los riesgos potenciales que introduce la computación cuántica.

1.2. Conceptos previos

Antes de meternos de lleno en la descripción de los protocolos de QKD, debemos repasar algunos de los fundamentos de la mecánica cuántica que constituyen la base para el desarrollo de estos sistemas. Es el momento de abandonar la intuición y aceptar unas nuevas reglas de juego que imperan en el mundo cuántico.

1.2.1. Principios de incertidumbre y teorema de no-clonación

El *principio de incertidumbe de Heisenberg* nos asegura que es imposible determinar, con precisión absoluta y de forma simultánea, el valor de dos magnitudes conjugadas de un sistema elemental. Ejemplos de estos son posición y momento, energía y tiempo, o tiempo y frecuencia. Así, según el principio de incertidumbre, resulta imposible determinar de forma precisa la posición y cantidad de movimiento de una partícula, o la energía y y tiempo, etc.

Otro de los resultados más importantes de la mecánica cuántica aplicado a QKD es el *teorema de no clonación*. Propuesto por Dieks, Wootters y Zurek en 1982, el teorema afirma la imposibilidad de copiar un estado cuántico desconocido de manera exacta, debido a que en el intento de obtener información acerca de este, la misma medición provoca su modificación[1]. Esto hace que sea una ley de la naturaleza la que nos garantice la seguridad de los procolos de QKD.

1.2.2. Superposición y entrelazamiento

La superposición y el entrelazamiento cuántico son dos conceptos difíciles de asimilar de una forma intuitiva. No podemos observar la superposición ni el

[1] Este mismo resultado es una de las principales dificultades para la construcción de un computador cuántico, puesto que el entorno funciona como un sistema que mide los estados, provocando su decoherencia.

entrelazamiento en el mundo macroscópico y, por lo tanto, no podemos atribuir sus efectos a un comportamiento que consideremos lógico de manera natural. Por esta razón, presentaremos estos conceptos desde un punto de vista práctico, aplicado a los sistemas que más tarde implementaremos.

El *principio de superposición* viene a afirmar que un sistema cuántico puede poseer simultáneamente dos o más valores de una cantidad observable, valores que pueden ser incluso contradictorios. Algo que se sale completamente de lo que podemos entender por razonable, pero que es clave para conocer el comportamiento de los sistemas que pretendemos implementar. La ilustración típica de este concepto se hace utilizando el ejemplo del *gato de Schrödinger*, que se presenta como una superposición de los estados, *gato vivo* y *gato muerto*. Evidentemente, en nuestra experiencia o bien el gato está vivo, o está muerto, pero nunca simultáneamente en ambos estados.

En un ejemplo más próximo a los sistemas que se usan en QKD, imaginemos un pulso de luz que contiene un único fotón[2]. Si hacemos pasar dicho pulso por un divisor de haz, obtenemos dos pulsos que seguirán caminos distintos, aunque el contenido del pulso sigue siendo un único fotón. El resultado es un fotón supuestamente dividido, y que se encuentra al mismo tiempo en dos localizaciones distintas. Es lo que conocemos como superposición de estados. Si medimos la posición del fotón lo encontraremos en un único camino (correspondiente a una de las dos salidas del divisor). En ese momento la función de onda ha colapsado y deja de encontrarse en un estado de superposición de los dos caminos. Ahora bien, este último argumento no quiere decir que el fotón había seguido un camino en concreto, puesto que realmente se encontraba en ambos caminos. En cada uno de los dos se encuentra con una cierta probabilidad (con el 50 % en cada camino si el divisor de haz es simétrico), esto es un estado de superposición. La forma de demostrarlo es modificando una propiedad del fotón en cada uno de los caminos, por ejemplo la fase, y provocar la interferencia con una intersección de los dos trayectos. En ese momento, podemos comprobar que el resultado va a depender de los cambios de fase aplicados en cada uno de los trayectos, luego realmente el fotón ha pasado por los dos caminos al mismo tiempo.

Otra característica de los sistemas cuánticos es el *entrelazamiento*[3], propiedad bajo la cual dos partes del sistema se encuentran ligadas de tal forma que ciertas modificaciones sobre una de ellas afectarán a la otra. Esto será así aún cuando las dos partes del sistema estén alejadas en el espacio y completamente descontectadas. El entrelazamiento vuelve a ser otro concepto que se sale del sentido común del mundo macroscópico, y que podemos entender mejor con un ejemplo presentado desde su formalización matemática (la no separabilidad). En el capítulo siguiente veremos como en la descripción de estados cuánticos se usa un espacio de Hilbert. En estos espacios los vectores base se suelen escribir en la notación de Dirac, por ejemplo, en un espacio de dimensión dos la base se escribiría como $|0\rangle$ y $|1\rangle$. Si tenemos dos qubits en el estado $|0\rangle$ lo escribimos

[2]Es el caso que se nos presentará en la implementación de un sistema de QKD.

[3]El entrelazamiento es una propiedad de las partículas cuánticas que fue vaticinada por Einstein, Podolsky y Rosen, allá por los años 30, en lo que conocemos como la *paradoja EPR*, como un intento por demostrar la incompletitud de la mecánica cuántica.

como $|00\rangle$. Este estado es separable, en el sentido en que representa el producto tensorial, $|0\rangle \otimes |0\rangle$, lo cual quiere decir que podemos medir ambos qubits por separado sin que la medida de uno afecte al otro. En este espacio existen también estados que no se pueden expresar como el producto directo de estados que afecten a una sóla partícula. En estos estados la medición del estado de un qubit afecta al otro. No importa lo separados que esten espacialmente. Así el estado $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ no se puede expresar como factores que afecten por separado a ambos qubits: si medimos el primer qubit y le encontramos en el estado $|0\rangle$ (lo que el estado arriba mencionado ocurrirá el 50 % de las veces), el otro qubit está también en el estado $|0\rangle$ (ver la demostración del teorema de no-clonación en el Apéndice, sección A.2). Estos estados se denominan entrelazados. En un mundo macroscópico en el cual tuviésemos "monedas cuánticas" esto significaría que si yo creo el estado entrelazado arriba mencionado, siendo $|0\rangle$ cara y $|1\rangle$ cruz. Sin mirarlas (medir su estado) dos personas se llevan cada uno una de las monedas en su bolsillo. Si uno de ellos mira su moneda y obtiene "cara", sabrá que indefectiblemente la moneda que se llevó la otra persona también estará en "cara". Lo mismo ocurre si el resultado obtenido es de cruz. Este comportamiento "mágico" es de gran interés para la criptografía cuántica, puesto que nos permitirán implementar un sistema muy especial para la distribución de claves.

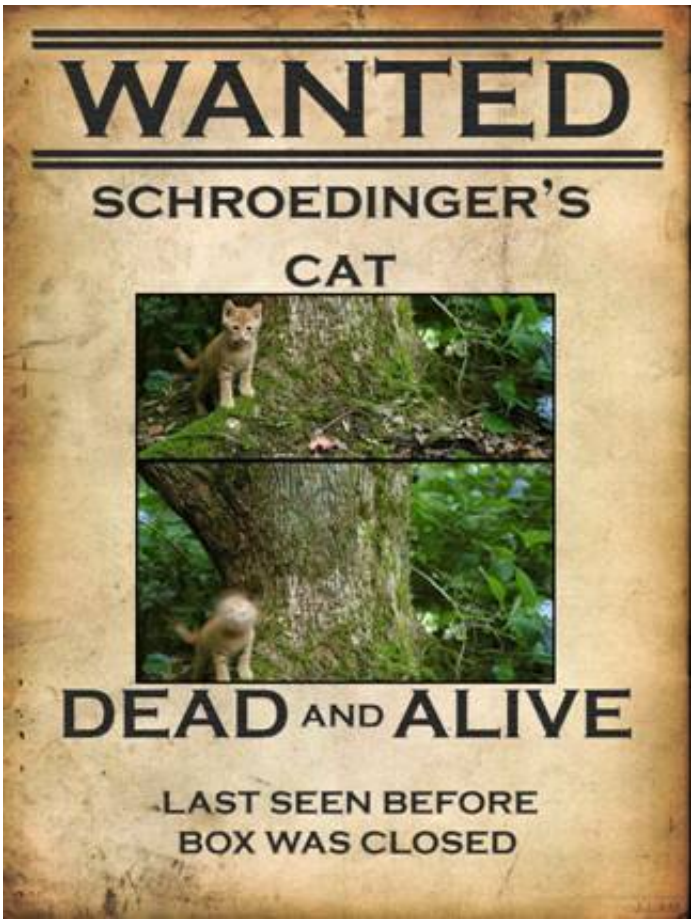


Figura 1.2: El gato de Schrödinger.

Parte II

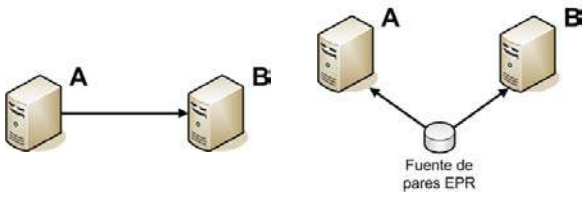
Sistema

Capítulo 2

Protocolos

Desde la propuesta, en 1984, del primer protocolo de distribución cuántica de claves hasta hoy, han aparecido diversas alternativas para el intercambio seguro de claves basadas en los principios físicos de la mecánica cuántica. Estas propuestas han seguido dos caminos distintos desde que Ekert inventa, en 1991, el primer protocolo basado en pares entrelazados. De acuerdo con esta división encontramos, por un lado, los propuesta de los protocolos basados en la transmisión de un único qubit; grupo en el que se encuentran los protocolos BB84, B92 y SARG04 entre otros. Y por otro, los protocolos basados en pares entrelazados, también conocidos como pares EPR.

La implementación de ambos grupos de protocolos es radicamente distinta. Un protocolo como el BB84, que utiliza 4 estados no ortogonales, requiere de una fuente emisora y otra receptora de partículas. Mientras que un protocolo basado en pares EPR necesita dos unidades receptoras conectadas a una misma fuente de emisión. La siguiente figura, 2.1, muestra las diferentes configuraciones de ambos tipos de sistemas.



(a) Protocolo basado en estados no ortogonales

(b) Protocolo basado en pares EPR

Figura 2.1: Estructura a nivel físico de dos protocolos QKD distintos.

En el presente capítulo hablaremos algo más de los protocolos basados en pares entrelazados, pero en el resto del documento no volveremos a trabajar con ellos, ya que los basados en pares EPR están menos maduros.

2.1. Consideraciones previas

De aquí en adelante trabajaremos con una arquitectura básica para el estudio de una comunicación en la que sólo intervienen dos interlocutores, A y B, que personificaremos con los nombres anglosajones de *Alice* y *Bob* respectivamente. Ambos extremos, Alice y Bob, estarán conectados a través de dos canales de comunicación, uno cuántico o privado, y otro público o convencional (autenticado). A partir de este momento se asumirá que el canal convencional está autenticado, sin esta característica los protocolos de QKD pierden su seguridad.

Ocasionalmente, también introduciremos la actuación de un supuesto espía, al que denominaremos *Eve*. El objetivo principal de este nuevo agente es obtener la máxima información posible acerca de la clave intercambiada entre Alice y Bob[1], para lo cual, dispondrá de acceso a los canales cuántico y convencional bajo los siguientes supuestos:

1. El acceso al canal cuántico será total, teniendo la capacidad de hacer cualquier cosa que no esté prohibida por las reglas de la mecánica cuántica, y por lo tanto le fuerzan a:

   a) No poder copiar o duplicar la información transmitida a través del canal.

   b) La lectura de cualquier estado provocará la modificación del mismo.

2. El espía o atacante podrá leer toda la información transmitida a través del canal convencional, pero nunca podrá modificarla, ya que este se encuentra autenticado.

La figura 2.2 muestra un escenario completo donde aparecen todos los personajes que acabamos de describir. En ésta aparecen Alice, Bob y el espía, Eve.

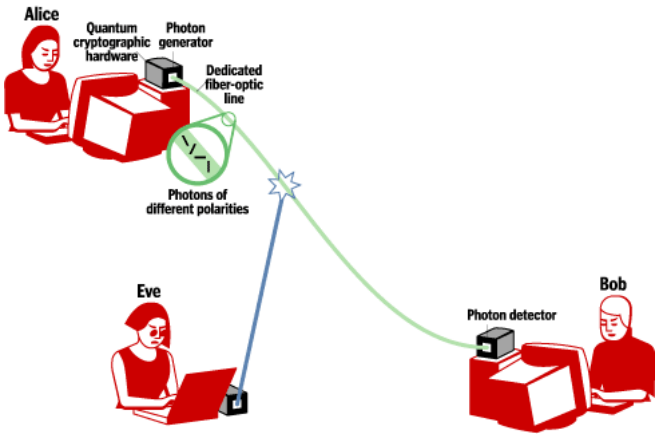


Figura 2.2: Escenario de un sistema QKD.

[1]El objetivo final de nuestro espía es el acceso a la información intercambiada entre los dos interlocutores, pero nos centraremos exclusivamente en el ataque al proceso de intercambio de claves realizado entre Alice y Bob, puesto que nuestro único interés es estudiar la seguridad de los sistemas de distribución cuántica de claves.

2.2. Protocolos basados en conjuntos de estados no ortogonales

2.2.1. BB84

El protocolo BB84 es, como comentamos en el capítulo de introducción, el primer protocolo de distribución cuántica de claves. Fue propuesto por Bennett y Brassard en la *International Conference on Computers, Systems and Signal* celebrada en Los Álamos, California, durante el año 1984 [6]. En la propuesta original la propiedad en la que se codifica la información que transporta el fotón es la polarización. Esta propiedad describe en que plano vibra el campo electromagnético en la dirección de propagación del haz. Se mide por tanto en grados. Si hacemos pasar un fotón con una polarización α a través de un filtro polarizador con una orientación β, el fotón pasará cambiando su polarización a β con probabilidad $\cos^2(\alpha - \beta)$, o será absorbido con la probabilidad complementaria, $\sin^2(\alpha - \beta)$. Si las diferencia $\alpha - \beta$ es exactamente 90º el fotón nunca pasa, y si es 0º pasa siempre sin ver afectada su polarización. Si la diferencia es de 45º, la mitad de las veces pasa adquiriendo una polarización β, y la otra mitad de veces es absorvido. Habitualmente se usan dos bases de polarización, una la horizontal-vertical (B_+), y otra oblicua o diagonal ($B_\times$). En la base B_+ tomamos como bit lógico 1 al qubit con estado de polarización vertical, y como 0 al horizontal. Una elección semejante se hace en el caso de la base oblicua, con el 1 a 45º y el 0 a 135º. Si preparamos un fotón en horizontal y lo medimos en la base diagonal, la mitad de las veces obtenemos 1 y la otra mitad 0. Lo mismo ocurre con un fotón preparado como 1 en la base $B_\times$ que es medido en la base B_+, la mitad de las veces se obtiene el resultado correcto y la otra mitad el incorrecto. La única manera de obtener con certeza cual era el estado original es conocer en qué base fue preparado y hacer la medición en la misma.

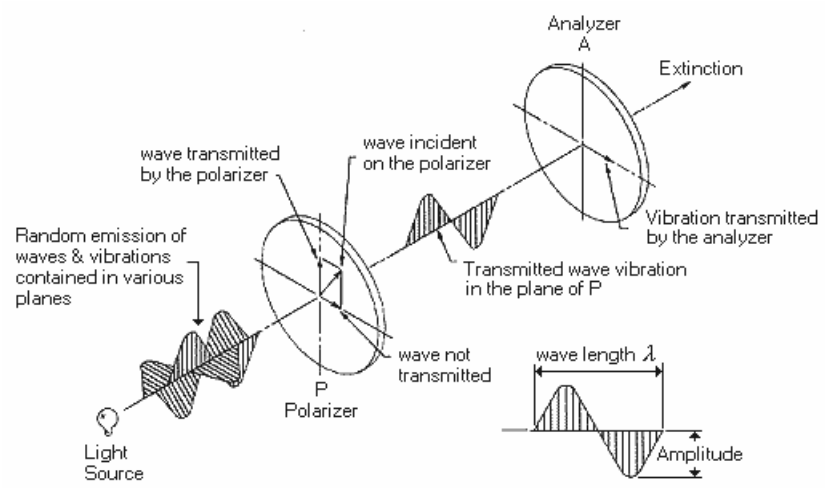


Figura 2.3: Polarización de un pulso de luz. En esta figura mostramos un ejemplo de como se comporta un polarizador, dejando pasar tan sólo las ondas del pulso que se transmiten en la dirección del plano de polarización.

Antes de comenzar una descripción del protocolo BB84, queremos hacer notar que este protocolo puede ser dividido en dos partes distinguibles por el canal de comunicación utilizado, y por lo tanto nos encontraremos con una parte del protocolo que podemos considerar cuántica y otra parte clásica o convencional, que estudiaremos de forma separada.

Intercambio de la clave a través del canal cuántico

La primera parte del protocolo BB84 discurre por medio de un canal de comunicación cuántico, a través del cual, dos interlocutores, Alice y Bob, intercambian un conjunto de qubits, en este caso implementados en fotones, codificados conforme a cuatro estados de polarización. Estos cuatro estados se agrupan formando dos bases con estados de polarización ortogonales. De esta forma, un estado de polarización queda perfectamente determinado al pasar por un polarizador correspondiente a su base, mientras que el resultado será totalmente aleatorio si pasa por un polarizador de otra base.

Base	Valor	Estado	Significado
B_+	$\leftrightarrow$	$\|\psi_{+0}\rangle$	Preparar el qubit (fotón) en la base B_+ (polarización horizontal-vertical) con el valor 0.
B_+	$\updownarrow$	$\|\psi_{+1}\rangle$	Preparar el qubit en la base B_+ con el valor 1.
$B_\times$	$\searrow$	$\|\psi_{\times 0}\rangle$	Preparar el qubit en la base $B_\times$ (polarización diagonal) con el valor 0.
$B_\times$	$\swarrow$	$\|\psi_{\times 1}\rangle$	Preparar el qubit en la base $B_\times$ con el valor 1.

Tabla 2.1: Nomenclatura de los estados en función de la base y el valor asociado.

En la tabla 2.1 definimos la agrupación de los cuatro estados utilizados por el protocolo. Se usan dos bases B_+ y $B_\times$, que se corresponden con los estados de polarización rectilíneos y oblicuos, respectivamente. También hemos asignado valores binarios a los estados de polarización: 0 para la polarizaciones horizontal y oblicua derecha, $\leftrightarrow$ y $\searrow$, y 1 para la polarizaciones vertical y oblicua izquierda, $\updownarrow$ y $\swarrow$. Donde los estados de las bases rectilínea y oblicua están relacionados de la forma:

$$|\psi_{\times 0}\rangle = \cos\frac{\pi}{4}|\psi_{+0}\rangle + \sin\frac{\pi}{4}|\psi_{+1}\rangle$$

$$|\psi_{\times 1}\rangle = \cos\frac{\pi}{4}|\psi_{+0}\rangle - \sin\frac{\pi}{4}|\psi_{+1}\rangle$$

El objetivo de esta primera fase del protocolo es que Alice prepare una secuencia aleatoria de n estados, obtenidos a partir de dos secuencias, también aletorias, una de bases y otra de valores para la clave.

Una vez que Alice tiene listo cada estado, envía el mismo a través del canal cuántico, y Bob trata de interpretar dicho estado eligiendo al azar una de las dos bases posibles. Puesto que las secuencias de bases elegidas por Alice y Bob

son aleatorias, la probabilidad de que coincidan es del 50 %, por lo que tan sólo la mitad de los estados codificados por Alice habrán sido interpretados correctamente por Bob. En el resto de estados, aunque la base utilizada no ha sido la adecuada, la probabilidad de acierto será también del 50 %. Así, la información total que Bob dispone de la clave inicialmente generada en Alice es del 75 %.

Reconciliación de bases por canal público

Una vez que Bob ha recibido a través del canal cuántico la secuencia completa de estados generados por Alice, ambos interlocutores vuelven a ponerse en contacto, ahora por medio de un canal público autenticado, para intercambiar las bases utilizadas en la codificación y decodificación de la secuencia. Aquellos valores para los que se ha utilizado una base distinta en el proceso de intercambio son desechados. En el caso ideal, sin espía ni ruido introducido por la línea, el resultado de este proceso de *reconciliación* será una clave idéntica en ambos extremos de la comunicación. Si hubiese un espía (o con ruido en la línea) esta clave no sería idéntica, por lo que sigue un paso de búsqueda recursiva de errores. El número de errores nos permite definir la tasa de errores cuánticos (QBER, Quantum Bit Error Rate), magnitud que resultará clave para decidir si se puede transmitir una clave con "seguridad absoluta", en realidad, con un número acotado de bits conocidos por un hipotético espía. Esta cota la podemos hacer tan baja como queramos siguiendo un nuevo proceso de discusión en el canal público conocido como amplificación de la privacidad.

Descripción del algoritmo

A continuación, en el algoritmo 1, mostramos una descripción más precisa de los pasos a seguir para la implementación del protocolo. En casa paso resumimos los procedimientos que Alice y/o Bob deben completar según la descripción que acabamos de realizar. En el algoritmo no contemplamos el momento en el que Alice y Bob se ponen de acuerdo para decidir el tamaño, n, de la secuencia de valores a intercambiar. Ese instante debe ser anterior a la ejecución del protocolo.

También hemos incorporado una tabla con la simulación de este algoritmo para el intercambio de una clave de 10 bits, siguiendo paso a paso los procedimientos descritos en el algoritmo 1. Notesé que no hay errores más allá de los introducidos por las diferentes elecciones de base entre Alice y Bob. Esto es un canal cuántico sin ruido ni presencia de un espía.

La seguridad del protocolo

La seguridad del protocolo reside en que un posible espía, Eve, no tiene información alguna durante la transmisión, acerca de la base que ha de utilizar para interpretar la información transmitida por Alice, de tal forma que dicho espía sólo podrá acertar con la base correcta en el 50 % de los casos, y cuando la base elegida no sea la adecuada, podrá obtener cualquier valor, lo que supone un 25 % de error en la información recibida. Ahora bien, al intentar leer la información transmitida desde Alice, es espía modifica la misma, por lo que no

Algoritmo 1 Protocolo BB84. Basado en 4 estados cuánticos.

1. Alice genera una secuencia de valores aleatorios que corresponderá con la clave que desea intercambiar con Bob.
2. Alice genera otra secuencia aleatoria, ahora con las bases que utilizará para la codificación de la clave generada en el paso anterior.
3. Alice codifica cada valor de la clave con la base correspondiente, según la tabla (2.1). Y envía la secuencia de qubits a Bob.
4. Bob genera una secuencia aleatoria con las bases que utilizará para decodificar la secuencia de estados recibidos de Alice.
5. Bob mide cada estado recibido en la base correspondiente a la secuencia generada.
6. Bob envía a Alice la secuencia de bases utilizada a través de un canal público autenticado.
7. Alice compara la secuencia de bases que ha utilizado para codificación de la clave con la secuencia proporcionada por Bob en el paso anterior, quedánsose sólo con aquellas mediciones para las que han coincidido ambas bases.
8. Alice y Bob comparten ahora una secuencia de valores formada por aquellos en los que las posiciones donde las bases de preparación y medición han coincidido.
9. Después de los puntos anteriores existe un postproceso cuyo objetivo es estimar la presencia de un espía, corregir los errores, y amplificar la privacidad. Estos pasos siempre se ejecutan en un proceso de QKD, aunque formalmente no se consideren parte del protocolo BB84.

tiene más remedio que utilizar una estrategia intercepta-reenvía con la información recibida, lo que supondrá que Bob reciba la información con una tasa de error del 25 % que le indicará la presencia del espía. El espía no tiene muchas más opciones, puesto que, el teorema de no-clonación le impide duplicar la información transmitida por Alice para intentar interpretarla después de que Alice y Bob hayan intercambiado las bases utilizadas. Por esta misma razón, Eva no tiene más remedio que interpretar la información utilizando tan sólo una de las bases.

En la parte final de este proyecto veremos las distintas estrategias de ataque que puede realizar un posible espía para optimizar la probabilidad de acierto en la interpretación de la información transmitida por Alice.

2.2.2. B92

En 1992, Charles H. Bennett propone una modificación del protocolo BB84 que utiliza tan sólo dos estados para la codificación de cada valor de la clave a

Intercambio cuántico										
1. Valores aleatorios	0	1	1	0	1	0	0	0	1	1
2. Bases aleatorias (Alice)	+	+	×	+	×	×	+	×	+	×
3. Codificación	↔	↕	↘	↔	↘	↙	↔	↙	↕	↘
4. Bases aleatorias (Bob)	×	+	+	+	+	×	+	+	×	×
5. Decodificación	x	1	x	0	x	0	0	x	x	1
Reconciliación de bases										
6. Bases utilizadas	×	+	+	+	+	×	+	+	×	×
7. Coincidencias	0	1	0	1	0	1	1	0	0	1
8. Clave reconciliada		1		0		0	0			1
9. Información revelada		1					0			
9'. Confirmación		Ok					Ok			
Salida										
8'. Clave final				0		0				1

Tabla 2.2: Simulación del protocolo BB84 para el intercambio de una clave de 10 bits, en ausencia de ruido y espía, según los pasos descritos en el algoritmo 1.

intercambiar[2] [13]. Es nuevo protocolo no posee grandes ventajas sobre su predecesor, el protocolo BB84, por lo que su interés no va más allá del académico. A pesar de esto, hemos querido presentarlo en este capítulo dado que, en su modo de trabajo podemos encontrar cierta similitud con el protocolo que describiremos a continuación, el SARG04. Esa similitud viene dada por el hecho de que el protocolo no publica las bases utilizadas, sino parte del resultado obtenido.

Descripción del algoritmo

Al igual que ocurre con el protocolo BB84, en ausencia de espías y de ruido en la línea de comunicación, una vez completado el protocolo, la coincidencia en cada valor de la clave intercambiada es del 100 %. Pero en este caso, la longitud final de la clave después del proceso de reconciliación será del 25 % sobre el tamaño de la clave en bruto (la secuencia de bits inicialmente intercambiada).

2.2.3. Estados trampa. (*Decoy states*)

Tenemos que adelantarnos al estudio de los componentes físicos que utilizaremos en la construcción de un sistema QKD, para comprender la propuesta de estos dos últimos protocolos, estados trampa y SARG04. Y es que ambos protocolos surgen a partir de la necesidad de incrementar la seguridad de un sistema QKD frente a un tipo especial de ataques, los ataques por *división del número de fotones*, PNS[3]. Este tipo de ataque lo estudiaremos en otro capítulo del presente documento, por lo que nos quedaremos con la única información que nos interesa conocer en este momento, y es que los ataques PNS se basan en la imprecisión

[2] Recordemos que el protocolo original, BB84, utiliza cuatro posibles estados para la codificación de cada valor de clave a intercambiar entre Alice y Bob.

[3] Photon Number Splitting.

Algoritmo 2 Protocolo B92. Basado en 2 estados cuánticos no ortogonales.

1. Alice y Bob se ponen de acuerdo para decidir el tamaño, n, de la secuencia de bits a intercambiar.
2. Alice genera una secuencia de bits aleatorios, $a = \{0,1\}^n$.
3. Bob genera una secuencia aleatoria de bases: $b = \{B_+, B_\times\}^n$.
4. Alice codifica cada bit de la secuencia con su estado correspondiente, según muestra la tabla ...
   a) Si $a_i = 0$, entonces utiliza el estado $|\psi_{+0}\rangle$.
   b) Si $a_i = 1$, entonces utiliza el estado $|\psi_{\times 1}\rangle$.
5. Bob interpreta cada estado recibido desde Alice, a través del canal cuántico, utilizando la base correspondiente según la secuencia generada en el paso 2.
   a) Si la base elegida es B_+:
      1) Si el valor obtenido es $|\psi_{+0}\rangle$, descarta el valor.
      2) Si el valor obtenido es $|\psi_{+1}\rangle$, registra un 1.
   b) Si la base elegida es $B_\times$:
      1) Si el valor obtenido es $|\psi_{\times 0}\rangle$, descarta el valor.
      2) Si el valor obtenido es $|\psi_{\times 1}\rangle$, registra un 1.
6. Bob envía a Alice aquellas posiciones en las que ha registrado un 1.
7. Alice utilizará como clave los valores de la secuencia de bits generada en el paso 1, que corresponden con las posiciones publicadas por Bob en el paso anterior.

de las fuentes de emisión de fotones individuales. En la actualidad no existen fuentes de fotones individuales bajo demanda, de modo que éstas se construyen atenuando pulsos de un láser. Esa atenuación no es lo suficientemente precisa, por lo que el pulso atenuado podrá contener 0, 1, 2 o más fotones. Cuando el pulso atenuado contiene más de un fotón, la seguridad de nuestro sistema se ve afectada por la posibilidad de que un espía acceda a la información transportada por los fotones sobrantes. En el caso límite la información obtenida por el espía es completa y no deja traza de su intromisión.

El riesgo al que nos enfrentamos en un sistema QKD debido a un ataque por PNS nos lleva a reducir el número medio de fotones esperados a la salida de la fuente, de tal forma que los pulsos con más de un fotón no compromentan la seguridad del sistema. Esta reducción del número promedio de fotones emitidos es crítica cuando las pérdidas en el canal de comunicación son altas, lo que limita considerablemente la distancia que podemos alcanzar con un sistema

QKD[4]. Esta limitación en la distancia que nos imponen los ataques PNS se puede eludir mediante dos estrategias: por un lado con la utilización de repetidores cuánticos, o también con las comunicaciones a través de satélite. Pero ambas tecnologías están muy lejos de ser factibles en la actualidad, por lo que tenemos que buscar otras estrategias para incrementar la distancia en un sistema QKD sin aumentar el riesgo por un ataque PNS. Una de esas alternativas es la proporcionada por los estados trampa, o *decoy state*s. En realidad los estados trampa no son un protocolo propiamente dicho, sino más bien un mecanismo para incrementar la distancia, o la seguridad según se mire, de un protocolo de distribución cuántica de claves. Por lo tanto, un estado trampa debe formar parte de un protocolo QKD como puede ser el BB84. W.-Y. Hwang es el autor que propone la utilización de este nuevo mecanismo [34].

Descripción del protocolo

Para la implementación de los estados trampa Alice utiliza dos fuentes de fotones: una fuente normal, S, y otra fuente de "trampas", S'. En la primera de las fuentes, S, emitimos con un número promedio de fotones $\mu < 1$, mientras que con la fuente de estados trampa, S', emitimos con un número promedio de fotones superior $\mu' \geq 1$. Con esta configuración, la fuente trampa emitirá más pulsos multifotón[5] que fotones individuales. Asumimos que los pulsos trampa son codificados de forma aleatoria, al igual que el resto de pulsos, para que no puedan ser distinguidos. Además, asumimos que los detectores de fotones son incapaces de apreciar el número de fotones contenidos en un pulso. Bajo estas condiciones, Alice comienza a enviar pulsos a Bob utilizando de forma aleatoria cualquiera de las dos fuentes, y no publica cuales han sido los pulsos trampa hasta que Bob comunica que ha terminado de recibir todos los pulsos. Una vez que se ha completado el envío y la recepción de pulsos, Alice y Bob estiman cual ha sido el rendimiento de cada una de las fuentes, S y S', en función de las detecciones y el número promedio de pulsos, μ y μ', a la salida de cada una de las fuentes. Si el rendimiento de la fuente de estados trampa ha sido muy superior al rendimiento de la fuente original, el protocolo ha detectado una ventaja sobre los pulsos multifotón que infiere la actuación de un ataque por PNS, en cuyo caso, Alice y Bob descartan todos los pulsos intercambiados.

El algoritmo 3 define paso a paso una implementación práctica del mecanismo de estados trampa. En la práctica, Alice no utiliza dos fuentes de fotones distintas, sino una única fuente donde se conmuta la configuración del número promedio de fotones esperado a la salida de la fuente.

2.2.4. SARG04 (versión prepara-y-mide)

Valerio Scarani, Antonio Acín, Grégoire Ribordy y Nicolas Gisin, proponen en 2004 una alternativa al protocolo BB84 [42] que pretende incrementar la seguridad de un sistema QKD frente a ataques por división del número de fotones, PNS. El nuevo protocolo utiliza cuatro estados, agrupados en dos bases

[4]Las pérdidas en un canal de comunicación aumentan con la distancia.
[5]Pulsos con más de un fotón.

Algoritmo 3 Estados trampa.

1. Seleccionamos los valores promedio que utilizaremos para envío de pulsos normales, $\mu < 1$, y pulsos trampa, $\mu' \geq 1$.
2. Alice comienza el envío de fotones a Bob seleccionando de forma aleatoria el uso de un valor u otro para el número promedio de fotones a la salida de la fuente.
3. Cuando Bob comunica que ha completado la recepción de los pulsos, Alice publica los índices de los pulsos donde se ha enviado un estado trampa.
4. Con el conocimiento del número promedio de fotones utilizado para los pulsos normales y pulsos trampa, μ y μ', y sus detecciones asociadas, Bob calcula cual ha sido la eficiencia de cada uno de los tipos de pulsos, normal o trampa.
5. Si la eficiencia de los pulsos trampa es muy superior a la eficiencia de los pulsos normales, Alice y Bob abortan el protocol. En caso contrario, continúan con la reconciliación de una clave siguiendo el protocolo utilizado (por ejemplo, BB84).

no ortogonales, con un funcionamiento a nivel cuántico idéntico al del protocolo BB84. La diferencia entre ambos protocolos radica entonces en el procedimiento de reconciliación de bases realizado a través del canal público. En lugar de intercambiar las bases utilizadas, Alice anuncia por el canal público uno de los cuatro pares de estados no ortogonales:

$$\mathcal{F}_{x,y} = \{|\psi_{+x}\rangle, |\psi_{\times y}\rangle\}$$

En esa pareja de estados se encuentra el enviado por Alice, hecho que Bob interpreta en un sentido opuesto al imaginable. Veámoslo con un ejemplo. Imaginemos que Alice envía a Bob el estado $|\psi_{+1}\rangle$ a través del canal cuántico, para después, anunciar por el canal público la pareja de estados $\mathcal{F}_{1,1}$[6]. Si Bob ha utilizado la base B_+ (ver tabla 2.1), el estado obtenido debe ser el mismo que envió Alice, $|\psi_{+1}\rangle$, por lo que el resultado no es concluyente. De igual forma, si Bob obtiene el estado $|\psi_{\times 1}\rangle$, no puede llegar a ninguna conclusión que le asegure que la base utilizada y, en consecuencia, el estado obtenido son correctos. Si por el contrario Bob mide utilizando la base incorrecta y obtiene un estado distinto al proporcionado por Alice (lo que ocurre en el 25 % de los casos), es decir, si Bob obtiene el estado $|\psi_{\times 0}\rangle$, sabe a ciencia cierta que la base utilizada no fue la correcta, luego el estado proporcionado por Alice para la base no utilizada, es el correcto.

Este protocolo es especialmente interesante para evitar uno de los ataques más importantes de los sistemas QKD: el ataque por división del número de fotones, PNS. En este tipo de ataques el espía obtiene información de la clave intercambiada a partir de la medida de fotones "sobrantes", provenientes de

[6] $\mathcal{F}_{1,1} = \{|\psi_{+1}\rangle, |\psi_{\times 1}\rangle\}$.

pulsos intensos cuya atenuación dejó más de un fotón. Para protocolos como el BB84, donde el atacante tiene conocimiento de las bases utilizadas por Alice y Bob[7], la información obtenida por el espía es determinista en aquellas medidas donde la base utilizada coincide con la publicada por Alice y Bob. Esto no ocurre con el protocolo SARG04 ya que, el atacante no tiene forma de comprobar si las bases utilizadas para medir los pulsos atacados coinciden o no con las utilizadas por Alice y Bob.

Algoritmo 4 Protocolo SARG04.

1. Alice genera dos secuencias aleatorias de un mismo tamaño, n, una secuencia de bases y otra de valores. A partir de los elementos que ocupan una posición concreta, i, en ambas secuencias, Alice obtiene el estado asociado y lo envía a Bob a través del canal cuántico.

2. Bob genera una secuencia de bases del mismo tamaño que las generadas en Alice, n, e interpreta cada estado recibido con la base correspondiente.

3. Alice anuncia por el canal público una pareja de estados, cada uno de ellos correspondiente a una de las bases utilizadas, y donde debe aparecer el estado enviado.

4. Bob interpreta el resultado obtenido de la siguiente forma:

   a) Si para la base utilizada por Bob en el proceso de medida, el estado obtenido es el mismo que Alice ha publicado, el resultado no es concluyente, por lo que Bob debe descartar ese valor.

   b) Si al contrario, Bob mide un estado distinto al que Alice publicó para la base utilizada. Bob tiene la certeza de que ha utilizado la base incorrecta, y por lo tanto, el estado anunciado por Alice para la otra base es el correcto.

   Con la interpretación de los resultados, Bob envía a Alice las posiciones donde tiene la certeza de conocer el estado intercambiado.

5. Alice y Bob comparten una clave idéntica en los extremos de la comunicación, de tamaño aproximado a $\frac{1}{4}$ de la secuencia de estados enviados.

Descripción del algoritmo

Para poder presentar una descripción funcional del protocolo, volvemos a realizar la misma asignación de valores del BB84, donde:

Base	Estados B_+	Estados $B_\times$	Valor asignado
B_+	$\|\psi_{+0}\rangle$	$\|\psi_{\times 0}\rangle$	0
$B_\times$	$\|\psi_{+1}\rangle$	$\|\psi_{\times 1}\rangle$	1

[7]Asumimos que esto es cierto debido a que la información con las bases utilizadas en la ejecución del protocolo BB84 se intercambia por un canal público.

2.3. Protocolos basados en estados entrelazados

2.3.1. E91

En 1991, Ekert propone un nuevo protocolo para la distribución cuántica de claves [10], que destaca por su fundamento en un principio de la mecánica cuántica distinto al utilizado por el resto de protocolos, el entrelazamiento. Es el primer método que propone el uso de pares entrelazados, también conocidos como pares EPR, para conseguir el intercambio de información segura, o una clave, entre dos interlocutores, Alice y Bob. El sistema propuesto por este protocolo es el siguiente. Una única fuente de pares entrelazados envía, de forma simultánea, pares de fotones EPR hacia dos interlocutores, Alice y Bob, a través de un canal de comunicación cuántico que une cada extremo con la fuente de pares EPR. Cuando cada interlocutor recibe su parte del par entrelazado, interpreta el estado recibido utilizando una base elegida aleatoriamente de entre dos $\{B_+, B_\times\}$, con lo que cada extremo completará dos secuencias con el valor detectado y la base utilizada en cada momento. Cuando el intercambio de estados acaba, ambos interlocutores intercambian por medio de un canal público autenticado las bases utilizadas durante el proceso de medida, y así desechan los valores obtenidos de una medición con bases distintas. El resultado debe ser una clave idéntica entre ambos extremos de la comunicación.

Descripción del algoritmo

Algoritmo 5 Protocolo E91. Basado en pares entrelazados.

1. Alice genera una secuencia de bases perfectamente aleatoria.
2. Bob genera otra secuencia de bases perfectamente aleatoria.
3. La fuente comienza la emisión de pares entrelazados hacía ambos extremos de la comunicación.
4. Alice interpreta cada estado recibido utilizando la base B_+ o $B_\times$ según la secuencia generada en el paso 1.
5. Bob realiza la misma interpretación que Alice de los estados recibidos, con respecto a la secuencia de bases generada en el paso 2.
6. Los extremos de la comunicación, Alice y Bob, repiten los pasos 4 y 5 hasta que la fuente deja de emitir pares EPR. Entonces ambos interlocutores intercambian la secuencia de bases utilizadas en cada medida.
7. Uno de los dos interlocutores, Alice o Bob, responde al otro con una secuencia de valores indicando en qué base realizó cada medida.

El algoritmo 5 describe paso a paso como funciona el protocolo. En esta descripción damos por hecho que Alice y Bob conocen el momento exacto en el que la fuente de pares EPR comienza la emisión de los mismos, así como el

número de pares entrelazados que va a enviar.

Conclusiones

En el capítulo de introducción definimos el concepto de entrelazamiento, y ahora, acabamos de estudiar el primer protocolo de QKD basado en esa propiedad de las partículas elementales. En el resto del documento, no volveremos a trabajar con este tipo de mecanismos, para centrarnos de lleno en el estudio de los protocolos basados en estados no ortogonales. Por muy prometedores que sean, en la práctica no existen sistemas de QKD basados en pares EPR más allá de equipos de laboratorio.

Como dato adicional comentar el hecho de que existe una versión del protocolo SARG04 preparada para trabajar con estados entrelazados [51], pero que no hemos presentado en este proyecto ya que no vamos a trabajar con este tipo de tecnología.

Capítulo 3

Implementación a nivel físico

La primera demostración práctica de un protocolo de distribución cuántica de claves tiene lugar en el año 1989[1]. El experimento fue realizado por Charles H. Bennett y John A. Smolin en el centro de investigación T.J. Watson de IBM[2], con la colaboración externa de Gilles Brassard y dos miembros de la universidad de Montreal, François Bessette y Louis Salvail. Las figuras 3.1 y 3.2 muestran varias fotografías de este primer prototipo[3], en las que podemos distinguir los tres componentes básicos del sistema: Alice, Bob y el canal cuántico que los comunica [11].

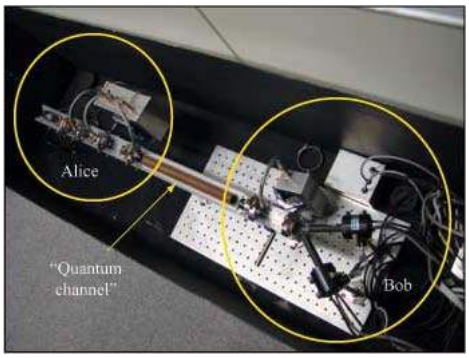


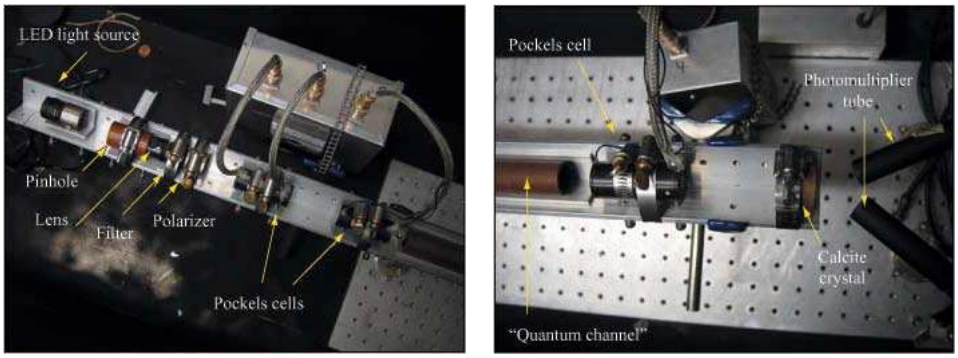
Figura 3.1: Primer prototipo de distribución cuántica de claves.

En esta primera implementación física de un sistema QKD se utilizó el fotón como soporte y la polarización como medida para la codificación de la información. En concreto, el experimento diferenciaba dos estados de polarización lineal: horizontal y vertical, formando una de las bases del sistema, y otros dos estados de polarización circular: izquierda y derecha, para componer la otra base.

[1] El protocolo implementado fue el definido por Bennett y Brassard en 1984, que era hasta la fecha la única propuesta disponible. Recordemos que tenemos que esperar hasta el año 1991 para encontrar la definición de un nuevo protocolo, el definido por Ekert y basado en la utilización de pares entrelazados.

[2] IBM T.J. Watson Research Center `http://www.watson.ibm.com/`

[3] Las imágenes mostradas en este documento, del primer prototipo de distribución cuántica de claves, han sido extraídas de la página web del centro de investigación T.J. Watson de IBM: The early days of experimental quantum cryptography `http://www.research.ibm.com/journal/rd/481/smolin.html`.



(a) Alice (b) Bob

Figura 3.2: Equipos utilizados para implementar los extremos del sistema.

Como comenta Smolin en su descripción del experimento, el objetivo final no era una demostración práctica de la mecánica cuántica, ni trataba de probar el correcto funcionamiento de los protocolos definidos. Ellos confiaban en que la teoría de la mecánica cuántica era correcta y no tenían razones para creer que el experimento no funcionara según lo esperado. En consecuencia, no se trataba realmente de un experimento científico, sino más bien de un proceso de ingeniería para la construcción de un primer prototipo. El resultado no pudo ser mejor, puesto que consiguió captar el interés de la comunidad científica, siendo el primero de muchos otros prototipos y demostrando la validez práctica de las teorías desarrolladas.

3.1. Componentes del medio físico

3.1.1. El fotón

Por sus características, una partícula de masa cero capaz de viajar a la velocidad de la luz, el **fotón** parece la opción más apropiada para implementar un sistema de distribución cuántica de claves[4]. Los fotones son qubits realmente buenos para esto: hay maneras relativamente fáciles de manipular el grado de libertad en el que se codifica la información, ya sea polarización, fase, etc. Pero no podemos fundamentar el diseño de un sistema en las propiedades de una partícula, necesitamos el apoyo adicional de un conjunto de herramientas que permitan construir dicho sistema. Los instrumentos que necesitamos en nuestro caso son:

- Una fuente de emisión de fotones individuales.
- Detectores para esos fotones individuales.

[4]Podemos estudiar implementaciones que utilicen otras partículas elementales. Por ejemplo, el *electrón* como medio de soporte y su espín como medida de información. Pero el tamaño y características de esta partícula hacen que este tipo de sistemas se vean altamente influenciados por el entorno. Esto supone una pérdida considerable de rendimiento que conlleva a que su utilización sea prácticamente inviable para largas distancias.

- Un canal de comunicación que permita el transporte del fotón manteniendo sus propiedades, es decir, aislándolo del mundo exterior. Esta última herramienta no resulta tan extraña, y sabemos de un medio que cumple gran parte de los requisitos esperados: la fibra óptica.

Pero además, no nos basta con disponer de un conjunto de herramientas, necesitamos que esas herramientas trabajen de forma conjunta, y lo hagan en las mejores condiciones posibles con el objetivo de obtener el mayor rendimiento de nuestro sistema. A continuación, veremos cómo ese objetivo se ve limitado por el conflicto de las condiciones de trabajo impuestas por cada componente. Luego, no sólo tendremos que luchar contra la imprecisión de un tecnología aún en desarrollo, sino que además, una configuración idónea del canal de comunicación supondrá una pérdida de eficiencia en otros componentes, como por ejemplo, los detectores.

3.1.2. Emisor de fotones

Hoy día no podemos encontrar fuentes capaces de emitir un sólo fotón cada vez que se le pide. En su lugar utilizamos pulsos láser atenuados, asumiendo el inconveniente de la baja precisión que nos aporta esta estrategia. Veremos cuales son las propiedades de un pulso láser atenuado y qué instrumentos podemos utilizar para obtenerlo.

Pulso láser atenuado

Como su nombre indica, un pulso láser atenuado no es más que un haz de luz cuya energía ha sido absorbida hasta dejar un pulso que contiene un número reducido de fotones. Esa absorción de energía o atenuación del pulso original se realiza con un dispositivo denominado atenuador. Para la implementación de un protocolo QKD necesitamos obtener un único fotón a la salida del atenuador, pero la imprecisión de este tipo de dispositivos nos impide alcanzar ese objetivo. Esto provoca que el pulso resultante contenga un número indeterminado de fotones, del que tan sólo conocemos su distribución. Sabemos que el número de fotones, n, contenido en un pulso láser atenuado sigue una distribución de Poisson de media μ (ver ecuación A.3), media que determina el número promedio de pulsos que contienen 0, 1 o más fotones.

$$P(n,\mu) = \frac{\mu^n}{n!}e^{-\mu} \tag{3.1}$$

En apartados posteriores estudiaremos la forma de elegir ese valor medio, μ, de manera que proporcionemos la menor cantidad de información posible a un hipotético espía, al mismo tiempo que maximizamos la probabilidad de obtener un fotón en el extremo final de la comunicación.

A modo de ejemplo, podemos comprobar como con una media de 0.5, $\mu = 0{,}5$, la probabilidad de obtener un fotón es de aproximadamente el 30 %[5], mientras que la probabilidad de no obtener fotón alguno es muy superior, por encima

[5] $P(1;0{,}5) = 0{,}3$.

del 60 %, quedando los casos en los que obtenemos dos o más fotones, que se aproximan al 10 %. Si queremos reducir el número de pulsos con más de un fotón, aumentará aún más la probabilidad de no obtener fotón alguno a la salida del atenuador, lo que añadido a las pérdidas de la línea de comunicación utilizada y la ineficiencia de los detectores provocará una reducción considerable del rendimiento final del sistema.

Atenuador óptico variable. VOA

El componente físico que utilizamos para obtener un pulso láser atenuado es el atenuador, o atenuador óptico (figura 3.3), instrumento óptico-mecánico que reduce la intensidad de un pulso láser mediante la presión o torsión de un segmento de fibra. Es un componente habitual en un laboratorio de óptica, aunque impreciso como acabamos de ver.

El control que requerimos de la atenuación es dinámico, en función de la distancia que separa a los extremos de la comunicación y del nivel de seguridad deseado (la seguridad del sistema se ve afectada directamente por el número promedio de fotones a la salida del atenuador). Por esta razón, utilizamos un atenuador óptico variable, VOA[6], que podemos ajustar para obtener el grado de atenuación deseado. Este ajuste es dinámico y se realiza por medio de pulsos eléctricos, lo que convierte finalmente al atenuador en un instrumento óptico-electromecánico.

Figura 3.3: Atenuador óptico variable.

En una implementación básica el atenuador sólo se configura una vez, al iniciar el sistema, por lo que no se exige una respuesta inmediata a ese proceso de configuración. En otras implementaciones más avanzadas, como por ejemplo aquellas donde se utilizan estados trampa (*decoy states*), el tiempo de respuesta del atenuador debe ser lo suficientemente rápido como para permitir valores de atenuación distintos en pulsos consecutivos[7].

3.1.3. Detector de fotones

El otro componente destacado en un sistema QKD es el detector de fotones. Afortunadamente, la criptografía cuántica no es la única aplicación práctica donde se requiere la utilización de detectores de fotones individuales. Otros

[6]Variable Optical Attenuator.

[7]Más adelante veremos como nuestro sistema emite pulsos de forma continuada en intervalos de 200 ns, tiempo que debe ser mejorado por la actuación de un atenuador que responda en un sistema con estados trampa. Esto dificulta el diseño de este tipo de componentes, puesto que, estamos hablando de una respuesta casi inmediata para un componente en parte mecánico.

instrumentos como el microscopio óptico utilizan este tipo de dispositivos con el objetivo de reconocer señales de luz muy débil. Por lo tanto, el interés sobre esta tecnología no es exclusivo de la criptografía, y podemos esperar un mayor desarrollo en un futuro inmediato.

No es un objetivo de este proyecto el completar una descripción técnica de los distintos tipos de detectores, pero sí mostramos a modo de referencia el nombre de las dos estrategias de detección más utilizadas en la actualidad, como son:

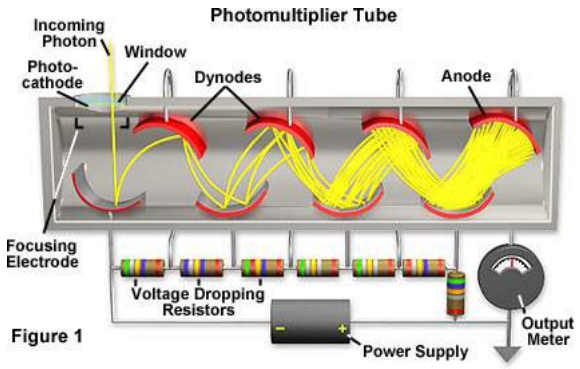


Figura 3.4: Tubo fotomultiplicador.

- **Fotomultiplicadores**[8] (figura 3.4). No se utilizan especialmente en sistemas QKD, aunque sí en otro tipo de utensilios de óptica, como el ya comentado microscopio óptico.
- **Fotodiodos de avalancha**, o APD[9] (figura 3.5).

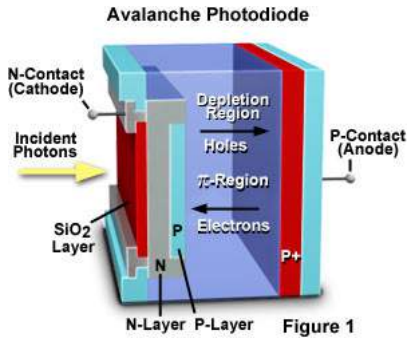


Figura 3.5: Fotodiodo de avalancha (APD).

Destacamos el hecho de que ambos tipos de detectores comparten una limitación, y es que no son capaces de identificar el número de fotones detectados[10].

[8] Photomultipliers.

[9] Avalanche photo-diodes.

[10] Al parecer, se está desarrollando en la actualidad un nuevo tipo de detectores, VLPC (*Visible Light Photon Counter*), capaz de distinguir con gran precisión el número de fotones contenidos en un pulso.

Esto implica que no pueden distinguir los pulsos con un único fotón de los pulsos que contienen más de un fotón, o lo que es equivalente, este tipo de detectores no podrá registrar la llegada de varios pulsos con una pequeña separación temporal[11].

Aunque tampoco las estudiaremos con detalle, sí queremos comentar que las causas de esta limitación son dos:

- Tiempo muerto (*dead time*). Es el tiempo durante el cual los dispositivos quedan inutilizados tras una detección. El bloqueo puede ser implementado de forma manual (por el usuario) o automática (por el sistema de detección). La razón por la que necesitamos interrumpir las detecciones se debe a la inestabilidad de los fotodetectores, causada por el estado de excitación en el que permanecen durante el lapso de tiempo que sigue a una detección. La excitación del detector se debe a la presencia de cargas que quedan atrapadas en el fotodetector después de producirse una detección, cargas que se irán disipando de forma exponencial durante el tiempo muerto (o tiempo de bloqueo). La activación del detector en ese intervalo de tiempo aumenta el riesgo de captar una detección falsa, lo que equivale a un incremento de la probabilidad de obtener un *afterpulse*, con la consiguiente reducción del rendimiento final del sistema[12].

- La multiplicación del ruido (*multiplication noise properties*).

Observación: La primera consecuencia del tiempo muerto es una pérdida de rendimiento causada un aumento forzado de la inactividad de los detectores. Cuanto mayor es el tiempo muerto, menor es la probabilidad final de obtener una detección por pulso generado. Por otro lado, conforme se incrementa el tiempo muerto aumenta también la fiabilidad de la detección obtenida, es decir, se reduce la probabilidad de obtener un *afterpulse*. Este efecto contradictorio del tiempo muerto obliga a encontrar un parámetro de compromiso que maximice el rendimiento final del sistema.

Rendimiento

Estudiando el rendimiento de los detectores de fotones descubrimos la primera traba de nuestro sistema, y es que los intervalos óptimos para la detección se encuentran fuera del rango de longitudes de onda deseable. Entendiendo por rango deseable aquel que coincide con el rango de transmisión óptimo en el canal de comunicación utilizado. Esta limitación la podemos observar en los sistemas que utilizan la tercera ventana de la fibra, la de mayor transmisión, como medio de transporte. En estos sistemas se utiliza un tipo de detector concreto,

[11] La imágenes que mostramos de ambos detectores han sido extraídas de la página web *Molecular Expressions*™, *Exploring the World of Optics and Microscopy*. Florida State University `http://micro.magnet.fsu.edu/`.

[12] En los equipos que hemos utilizado para nuestro proyecto el tiempo de bloqueo es de $10\mu s$, lo que se traduce en un impacto considerable sobre el funcionamiento del sistema (teniendo en cuenta que la separación temporal con la que se emiten dos pulsos es de tan sólo 200 ns).

En los capítulos finales de este proyecto veremos algunas de las vulnerabilidades que presenta nuestro sistema debido a este tiempo de bloqueo.

como es el InGaAs APD, que obtiene el mejor rendimiento alrededor de los 1550 nm, pero cuyo rendimiento es bastante inferior al de otros dispositivos como el fotodetector de Silicio, Si APD. La siguiente figura 3.6[13] muestra un ejemplo de lo comentado. En la gráfica se observa cómo el mejor rendimiento, menor ruido, es el obtenido por el fotodetector de Silicio, Si APD, en el intervalo de longitudes de onda que va desde los 800 nm hasta los 900 nm. Por otro lado, los otros dos fotodetectores, Ge APD e InGaAs APD, tienen un rendimiento óptimo similar, lo que hace que se diferencien exclusivamente por el rango de longitudes de onda donde se alcanza ese comportamiento óptimo. Mientras que el Ge APD posee un comportamiento óptimo en un intervalo más amplio que va desde los 1000 nm hasta los 1400 nm, el InGaAs APD obtiene ese mismo rendimiento en el entorno de los 1500 nm, intervalo que coincide con el rango de menor pérdidas de la fibra óptica (como veremos en la gráfica que mostramos a continuación, figura 3.7). En consecuencia, nos encontramos con dos parejas que aparecerán siempre yustapuestas en un sistema QKD: son las formadas por los fotodetectores Si APD y el aire, o InGaAs APD y la fibra óptica como canal de comunicación.

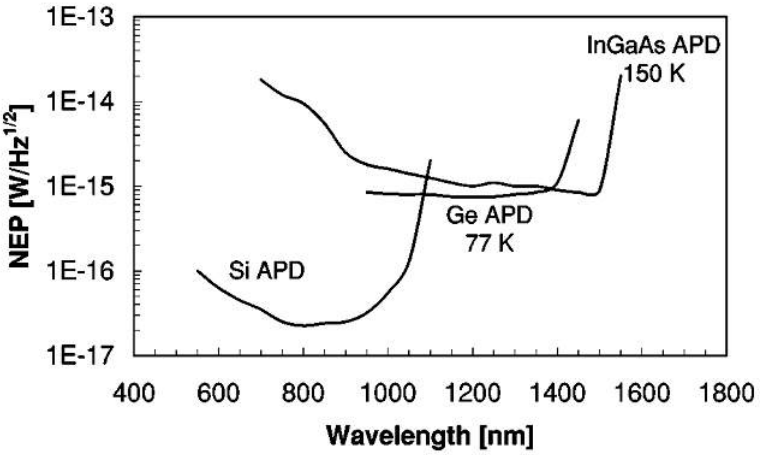


Figura 3.6: Ruido (NEP, *Noise Equivalent Power*) de los fotodetectores de avalancha, APD, en función de la longitud de onda.

3.1.4. Canal de comunicación

Fibra óptica

La fibra óptica es el canal de comunicación más utilizado en la actualidad. Por sus características, permite el transporte de información a través de largas distancias. Pero debemos tener en cuenta un detalle a la hora de estudiar este medio de transporte, y es que vamos a trabajar con pulsos que contienen un único fotón, lo que limitará considerablemente el rendimiento del sistema en función de la distancia utilizada, ya que el fotón es finalmente absorvido siempre.

[13]Imagen extraída del artículo "*Quantum cryptography*". Nicolas Gisin, Grégoire Ribordy, Wolfgang Tittel, Hugo Zbinden. Reviews of modern physics, vol. 74 (2002).

A pesar de que existen dos tipos de fibra: multimodo y monomodo; sólo trabajaremos con uno de ellos, los tipos de fibra monomodo, debido a que este tipo de fibras mantienen mejor las condiciones de los pulsos transmitidos, evitando la dispersión y obteniendo una atenuación menor en función de la distancia.

En la gráfica que aparece a continuación[14], figura 3.7, se pueden observar las tres bandas de transmisión utilizadas actualmente en los sistemas telecomunicaciones a través de fibra óptica[15], como son: la primera ventana situada en el entorno de los $850nm$, la segunda ventana en el entorno de los $1310nm$, y la tercera ventana en el entorno de los $1550nm$. La razón por la que se utilizan estos rangos de transmisión se debe a la baja atenuación que sufre un pulso en dichas amplitudes. Para la primera ventana la pérdida de potencia aproximada o teórica es de $2,5dB/Km$, para la segunda ventana la atenuación se reduce hasta los $0,4dB/Km$, mientras que para la tercera ventana la atenuación es la menor de los tres casos y fluctúa en torno a los $0,2dB/Km$ y $0,25dB/Km$. A partir de esta gráfica podemos también comprobar como la franja donde la fibra óptica sufre menos atenuación se sitúa entre los 1500 y los 1600 nm, que no encaja dentro de entorno de los 800 nm donde los detectores vistos anteriormente trabajaban con una mayor eficiencia (ver figura 3.6).

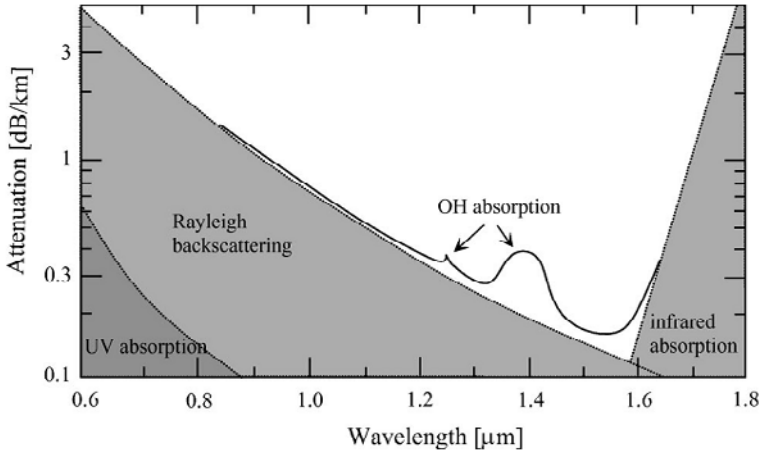


Figura 3.7: Curva de atenuación (pérdida de potencia) de un haz de luz a través de la fibra óptica.

Otro inconveniente de la fibra óptica es que no mantiene la polarización de los fotones enviados, por lo que debemos buscar otro mecanismo para codificar la información. Más adelante, veremos una alternativa basada en la fase de los estados transportados, y estudiaremos cuando resulta conveniente utilizar uno u otro mecanismo.

[14]Gráfica extraída de [30], "*Quantum Cryptography*" por N. Gisin et al.

[15]En la gráfica, el eje de las ordenadas muestra los valores de la atenuación en decibelios por kilómetro, mientras que el eje de las abscidas muestra la evolución de la atenuación en función de la longitud de onda del pulso.

Aire libre

En la actualidad, también se han implementado sistemas de transmisión de información a través de aire libre utilizando fotones. La ventaja de este tipo de sistemas es que permiten utilizar la polarización como transporte de información, debido a que este medio de transporte mantiene mejor las condiciones del fotón en cuanto a polarización.

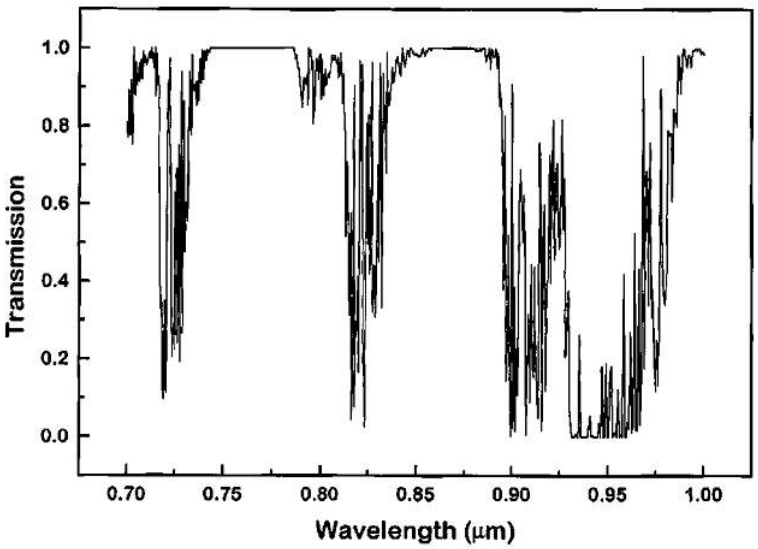


Figura 3.8: Eficiencia en la transmisión de un pulso a través del aire en función de la longitud de onda del pulso.

La figura 3.8 nos muestra otra de las ventajas de este tipo de implementaciones[16], como es el hecho de que la transmisión a través del aire tiene una ventana de rendimiento óptimo en el entorno de los 770 nm. La localización de dicha ventana permite utilizar este canal de comunicación en el rango de mayor eficiencia de los detectores (que vimos en la gráfica anterior, figura 3.6), ofrecido por el fotodetector de Silicio, Si APD.

3.2. Codificación

En el estudio de los componentes del medio físico nos hemos dejado un elemento por analizar, la codificación de la información. Conocemos el soporte o medio de transporte (los fotones), el canal de comunicación (la fibra óptica), y la forma de emitir y detectar esos fotones de forma individual. Nos queda por estudiar la manera en la que esas partículas elementales, los fotones, van a transportar la información que deseamos intercambiar. Buscamos una característica del fotón que pueda ser utilizada como almacén en el transporte, es decir, que permanezca invariante durante la comunicación y donde podamos almacenar la información deseada, en un proceso que denominaremos codificación.

[16]Gráfica extraída de [30], "*Quantum Cryptography*" por N. Gisin et al.

Hasta el momento sólo hemos hablado de la polarización como mecanismo para la codificación de la información a intercambiar, pero en este apartado veremos otra alternativa, la fase, cuyo comportamiento es más estable en el entorno de comunicación elegido (la fibra óptica). Discutiremos las ventajas e inconvenientes de ambos mecanismos de codificación, en distintos entornos y bajo diferentes condiciones de funcionamiento. Pero sobre todo, analizaremos las implicaciones que tiene cada mecanismo, fase o polarización, sobre la estructura del sistema físico que implemente una solución para la QKD.

3.2.1. Fase vs. polarización

Desde un punto de vista teórico-didáctico la *polarización*[17] es seguramente la mejor herramienta para explicar el comportamiento de los qubits codificados en fotones, pero desde el punto de vista práctico su utilización no es siempre apropiada, ya que muchos canales de transmisión modifican los estados de polarización, notororiamente en la fibra óptica usada normalmente en telecomunicaciones. No ocurre lo mismo con el aire, que es sí es un canal adecuado para transmitir qubits codificados en estados de polarización de fotones. Bien es cierto que existe un tipo de fibra óptica capaz mantener la polarización, PMF[18], pero se obtiene a partir de un proceso de fabricación más costoso[19], y no se utiliza habitualmente en la construcción de redes de fibra. En su lugar encontramos la fibra óptica monomodo para la que debemos tener en cuenta los siguientes factores:

- La polarización puede variar en los saltos entre distintos segmentos de fibra.
- Torsiones, o tensiones en la fibra pueden modificar la polarización de los fotones. Es lo que se conoce como birrefringencia, un efecto que provoca la transformación de la polarización lineal en polarización elíptica.

Por estas dos razones, la polarización no es la mejor opción para implementar un sistema QKD para fibra, y recurrimos a otros mecanismos. Una de las alternativas más interesantes es la utilización de la *fase*.

A pesar de lo comentado, existen entornos donde sí es interesante el uso de la polarización, como pueden ser los sistemas de comunicación aérea (donde la polarización no se ve afectada como en el caso de la fibra óptica), o prototipos donde la distancia del canal de comunicación es razonablemente pequeña[20].

3.2.2. Codificación con polarización

Ya vimos en el capítulo anterior cómo podemos utilizar la polarización para implementar un protocolo de distribución cuántica de claves. Y en el caso que

[17] Al utilizar de forma simplificada el término polarización nos referimos en todo momento a la polarización electromagnética.

[18] Polarization-Maintaining Fiber

[19] Para que un segmento de fibra sea capaz de mantener la polarización, éste debe ser "deformado" durante el proceso de fabricación.

[20] ¿Qué es para nosotros "razonablemente pequeño"? Por ejemplo, una red de acceso local.

nos ocupa, su implementación, no aparecen grandes diferencias con respecto a los procedimientos descritos en el capítulo de protocolos. Tan sólo necesitaremos un instrumento, el polarizador, para modificar el estado del fotón que sale desde Alice en dirección a Bob, y a su llegada, de nuevo otro polarizador para actualizar el estado del fotón entrante.

3.2.3. Codificación en fase

La cosa cambia cuando decidimos utilizar la fase como mecanismo de codificación. Y la razón principal se traduce en la implementación de un sistema completamente distinto, puesto que no utilizamos un instrumento específico para medir la fase aplicada por cada extremo de la comunicación, sino que provocamos un proceso de interferencia entre pulsos donde se han aplicado dos cambios de fase de forma independiente, con el objetivo de comprobar si esos cambios de fase producen una interferencia constructiva o destructiva, según si hemos aplicado cambios de fase complementarios o no.

La interferencia viene descrita por la ecuación:

$$I_0 = I \cdot \cos^2\left(\frac{\phi_A - \phi_B + k\Delta L}{2}\right)$$

Donde I_0 es la intensidad en el detector "0", I es la intensidad de la fuente, k es el número de onda[21], y ΔL es la diferencia entre el camino corto y el camino largo.

En la siguiente tabla, 3.1, mostramos todas las combinaciones de resultados que se pueden obtener de la interferencia de un sistema con cuatro estados de fase: 0, $\frac{\pi}{2}$, π y $\frac{3\pi}{2}$.

Alice		Bob		
Valor/bit	ϕ_A	ϕ_B	$\phi_A - \phi_B$	Valor/bit
0	0	0	0	0
0	0	$\frac{\pi}{2}$	$\frac{3\pi}{2}$	?
1	π	0	π	1
1	π	$\frac{\pi}{2}$	$\frac{\pi}{2}$	?
0	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$	?
0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	0	0
1	$\frac{3\pi}{2}$	0	$\frac{3\pi}{2}$	?
1	$\frac{3\pi}{2}$	$\frac{\pi}{2}$	π	1

Tabla 3.1: Resultados de un proceso de interferencia en función de la fase para un sistema de cuatro estados.

[21] El número de onda es inversamente proporcional a la longitud de onda, y su relación viene dada por: $k = \frac{2\pi}{\lambda}$.

El interferómetro de Mach-Zehnder

De todas las configuraciones de interferómetros disponibles, la de Mach-Zehnder[22] es la que mejor se adapta a nuestras necesidades. Con un sistema formado por dos detectores podemos un proceso de interferencia constructiva de otro destructivo. La figura 3.9 muestra un diagrama con la estructura del interferómetro de Mach-Zehnder. En él se pueden apreciar lo siguientes componentes: una fuente de luz, Q, dos divisores del haz 50/50[23], S1 y S4, dos espejos, M1 y M2, y dos detectores, D1 y D2.

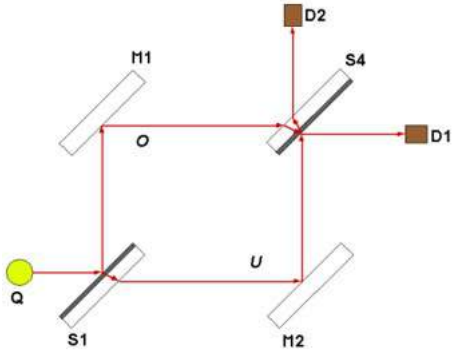


Figura 3.9: Componentes del interferómetro de Mach-Zehnder.

El funcionamiento del interferómetro es el siguiente. La fuente de luz, Q, emite un pulso en la dirección del primer divisor del haz, S1. El ángulo de incidencia del pulso sobre el primer divisor del haz debe ser de 45º, lo que provoca que el pulso se divida al 50 % en dos haces ortogonales, un haz continuará en la dirección del pulso original y el otro haz seguirá una dirección perpendicular. Los dos pulsos se reflejan un ángulo de 45º en los espejos M1 y M2 para volver a encontrarse en el segundo divisor del haz, S4. En éste los pulsos se solaparán para provocar el proceso de interferencia. Dependiendo del resultado activarán uno u otro detector con una probabilidad definida en función de la interferencia sufrida.

La figura 3.10 muestra un ejemplo de cada uno de los procesos involucrados en el interferómetro de Mach-Zehnder. En la primera división de la figura, (a), aparece el comportamiento de un divisor del haz. En la sección (b) se muestra el funcionamiento de un espejo. En (c) encontramos el interferómetro de Mach-Zehnder al completo. Y finalmente, en (d) tenemos un ejemplo de interferencia destructiva.

3.3. Estrategias de conexión

3.3.1. La idea original

Partiendo de la idea original propuesta por el interferómetro de Mach-Zehnder, podemos construir un circuito como el de la figura 3.11, cuyo funcionamiento

[22]El nombre proviene de los físicos Ernst Mach y Ludwig Zehnder.
[23]Divisor de un pulso al 50 %.

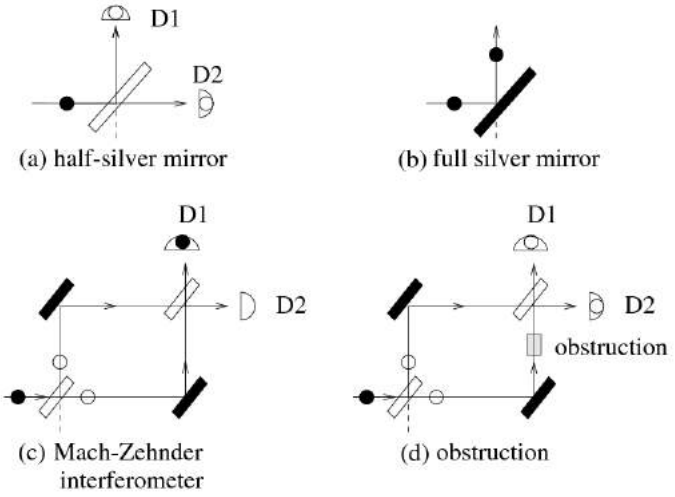


Figura 3.10: Funcionamiento del interferómetro de Mach-Zehnder.

es el descrito a continuación[24].

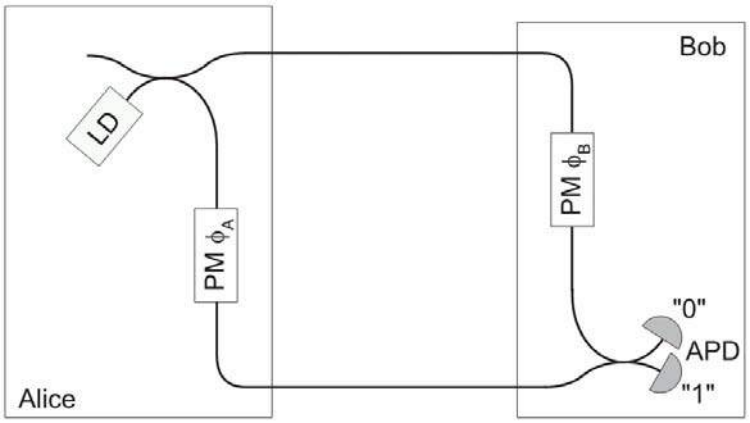


Figura 3.11: Implementación original de un protocolo QKD a partir del interferómetro de Mach-Zehnder. Los componentes que aparecen en el diagrama son: un diodo láser, LD, dos moduladores de fase, PM, y dos fotodetectores de avalancha, APD.

Una única fuente de fotones, LD, emite desde un extremo del circuito. El haz de fotones emitido es dividido a la salida de la fuente y se desplaza a través de dos caminos de idéntica longitud. En uno de los caminos Alice aplica, de forma aleatoria, un cambio de fase, PM_{ϕ_A}. En el otro camino Bob realiza el mismo proceso, cambiando la fase en PM_{ϕ_B}. Al final del circuito, ambos caminos vuelven a unirse por medio de otro divisor del haz. El resultado activará uno de los dos detectores, "0" o "1", en función de las fases aplicadas por Alice y Bob:

- Si Alice y Bob han aplicado el mismo cambio de fase, se producirá una interferencia constructiva en la unión de los caminos, lo que provocará la

[24]Imagen extraída de [30], "*Quantum Cryptography*" por N. Gisin et al.

activación de uno de los detectores.

- Si por el contrario, Alice y Bob han aplicado cambios de fase distintos, la interferencia resultante será destructiva, activando el otro detector.

Este primer diagrama es funcionalmente correcto, válido para implementar un sistema QKD, pero tiene dos grandes inconvenientes:

1. En el sistema que estamos construyendo la transmisión de fotones ha de ser exclusiva, y por lo tanto se exige que ambas líneas de comunicación sean dedicadas. Esto supone un coste de implementación elevado cuando la longitud que separa los extremos del sistema es considerable, no pudiendo rentabilizar ese coste mediante el uso compartido de las líneas por otros sistemas de comunicación.

2. Por otro lado, el proceso de interferencia que se produce al final de circuito, requiere que la longitud de los caminos sea idéntica. Algo realmente difícil de conseguir cuando trabajamos con fibra óptica y largas distancias, ya que un simple cambio en las condiciones del medio provoca la dilatación de una fibra.

3.3.2. Sistemas de dirección única (one-way)

En la práctica el uso de dos líneas de comunicación distintas para la construcción de un sistema QKD que recorra largas distancias es una solución impracticable, por los inconvenientes comentados en el apartado anterior: el coste y la dificultad de obtener dos caminos de idéntica longitud. Una primera alternativa propone una variación del interferómetro de Mach-Zehnder como muestra el diagrama ilustrado en la siguiente figura[25], 3.12.

En este nuevo esquema se utiliza una estrategia similar a la modulación en el tiempo. El pulso emitido se divide antes de salir de Alice, pasando por dos caminos de distinta longitud, y en uno de los caminos Alice cambia la fase del pulso que atraviesa ese trayecto. Al unirse de nuevo los caminos obtenemos dos pulsos[26], uno de ellos con la fase codificada por Alice y el otro sin modificar, que están separados una distancia equivalente a la diferencia entre los caminos del interferómetro de Alice[27]. Esa distribución de ambos pulsos de forma secuencial (en el tiempo), nos permite utilizar un único canal de comunicación, siempre y cuando Bob sea capaz de invertir el proceso realizado por Alice. A la llegada en Bob, se les hace pasar de nuevo a través de dos caminos de idéntica longitud a los de Alice, ofreciendo cuatro alternativas para el recorrido total de un pulso a través del sistema. En la tabla 3.2 vemos esa distribución en función de los trayectos elegidos, lo que nos da una idea del comportamiento que tendrá el

[25]Imagen extraída de [30], "*Quantum Cryptography*" por N. Gisin et al.

[26]Realmente lo que estamos enviando no son dos pulsos, sino la superposición de un fotón con dos estados de fase y localización distintos.

[27]No es preciso hablar de dos interferómetros distintos, en Alice y Bob, ya que sólo se produce un proceso de interferencia, dentro de Bob, con aquellos pulsos que han elegido caminos distintos en cada uno de los extremos. Mantenemos esa terminología debido a la similitud que presentan los dos caminos con un cambio de fase frente al interferómetro de Mach-Zehnder.

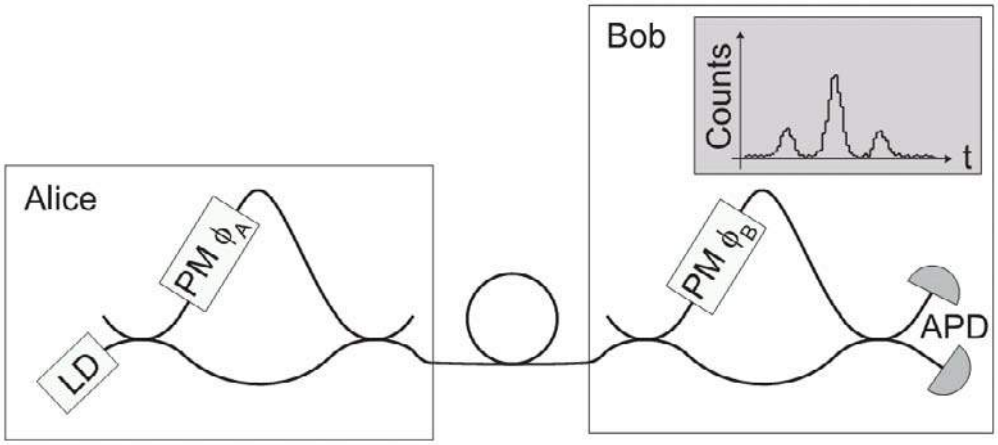


Figura 3.12: Modificación del interferómetro de Mach-Zehnder para la implementación de un protocolo QKD con caminos de dirección única a través de 2 interferómetros, usando un único tramo de fibra intermedia común para los caminos posibles.

sistema al final mismo. El resultado serán tres pulsos, que llegarán en tiempos distintos y con diferente intensidad en función de los siguientes parámetros:

- El tiempo empleado por un pulso en recorrer la distancia que separa a Alice y Bob, T, que dependerá de la longitud del canal de comunicación.
- El tiempo de recorrido del camino más corto del interferómetro, t_1.
- El tiempo de recorrido del camino largo, t_2.

De los tres pulsos, sólo nos interesa el intermedio (el más intenso), que se obtiene como la interferencia de las dos alternativas complementarias de la tabla 3.2. Para distinguir ese pulso intermedio debemos conocer con precisión los tiempos de recorrido de cada trayecto, y utilizar en cada interferómetro caminos de longitud lo suficientemente distinta como para que no se solapen los pulsos desechables (1 y 3).

Interferómetro en Alice	Interferómetro en Bob	Tiempo de llegada
Camino corto	Camino corto	$t_1 + T + t_1$
Camino corto	Camino largo	$t_1 + T + t_2$
Camino largo	Camino corto	$t_2 + T + t_1$
Camino largo	Camino largo	$t_2 + T + t_2$

Tabla 3.2: Recorrido de un pulso a través de los interferómetros de Alice y Bob.

Ahora bien, con este nuevo diagrama reducimos el coste de implementación utilizando una única línea dedicada, pero sólo hemos solucionado parcialmente el problema de la sincronización. Los caminos de cada interferómetro son más cortos, lo que nos permite ajustar mejor el tamaño de cada segmento para que la distancia recorrida por dos pulsos que siguen caminos alternativos sea idéntica.

Pero las condiciones en las que se encuentra cada interferómetro pueden ser distintas debido a la separación entre ambos, condiciones como la temperatura que dilatan o contraen la longitud de la fibra, y que van a provocar pequeñas diferencias entre los caminos. En la práctica esto implica mantener un pequeño gap de aire en los interferómetros para mantener las diferencias de longitud entre los lados cortos y entre los lados largos de los interferómetros en el rango de los milímetros.

3.3.3. Sistemas de doble dirección o Plug and Play (two-ways)

Aún podemos modificar el esquema anterior, con el objetivo de solucionar el problema de la sincronización entre los extremos. Esta nueva estrategia es una variación de la anterior cuya base sigue siendo la idea original del interferómetro de Mach-Zehnder. La siguiente figura, 3.13, muestra el diagrama de esta nueva estrategia[28].

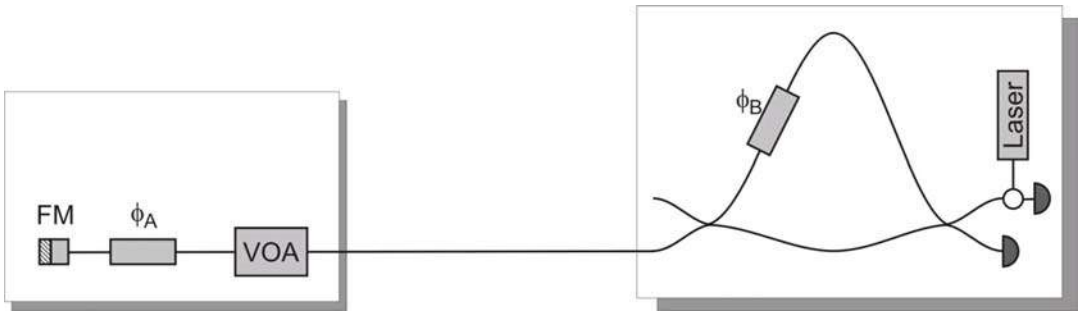


Figura 3.13: Implementación con caminos de doble dirección y un único interferómetro.

La idea consiste en seguir utilizando la modulación en el tiempo, es decir, cada fotón se divide y sus mitades pasan por caminos de distinta longitud para distanciarlos secuencialmente, y así poder utilizar un único segmento de fibra entre Alice y Bob. Pero ahora los pulsos van a ser de ida y vuelta, es decir, la fuente emisora de fotones se sitúa en Bob y los pulsos transmitidos recorren dos veces la distancia que separa a los extremos, Alice y Bob. Bajo esa idea, sólo necesitamos configurar un interferómetro en Bob, con caminos distintos, y esperar a que cada pulso recorra un camino diferente en cada uno de los trayectos de ida y vuelta. Pero veamos como funciona. Bob emite un pulso que se divide a través de dos caminos de longitud distinta, consiguiendo una modulación en el tiempo, es decir, los pulsos se dirigen hacia Alice con una separación que depende de la diferencia de longitud de los caminos del interferómetro. Cuando los dos pulsos llegan a Alice, ésta modifica la fase de uno de ellos, y aplica la atenuación necesaria como para dejar un pulso que contiene un único fotón a la salida. A la vuelta, en Bob, los pulsos vuelven a encontrarse con el interferómetro por lo que sufren una nueva división que marcará el comportamiento final del sistema. En la figura 3.14, mostramos los caminos recorridos por los pulsos que pasan a

[28]Imagen extraída de "*Quantum Cryptography Training Course*" por Grégoire Ribordy.

través de caminos distintos en los trayectos de ida y vuelta, coincidiendo en la unión final del interferómetro para producir la interferencia deseada[29].

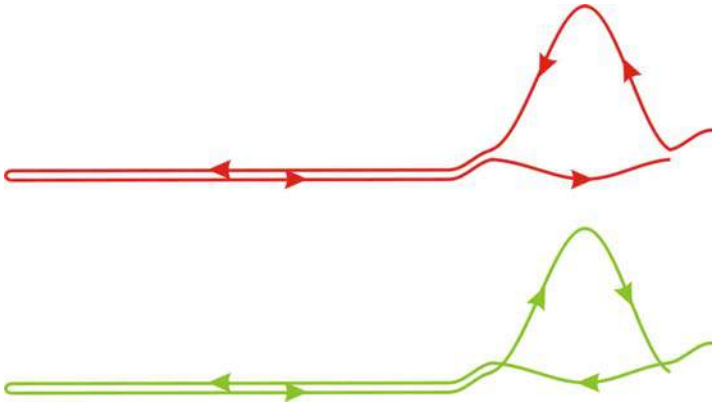


Figura 3.14: Recorrido de los pulsos que provocan la interferencia en Bob.

En función de los trayectos elegidos por cada pulso en cada una de las pasadas por el interferómetro, obtenemos cuatro combinaciones como muestra la tabla 3.3. De esas cuatro combinaciones sólo nos interesan dos, aquellas en las que los pulsos eligen un camino distinto en los trayectos de ida y vuelta. Sólo en ese caso los pulsos interfieren al regresar a Bob, produciendo el efecto deseado. Este resultado es similar al obtenido en la configuración anterior (para el sistema de una única dirección), con la única diferencia de que ahora hay que tener en cuenta dos veces la distancia que separa a Alice y Bob en el cálculo del tiempo de recorrido.

Trayecto de ida	Trayecto de vuelta	Tiempo de llegada
Camino corto	Camino corto	$t_1 + 2T + t_1$
Camino corto	Camino largo	$t_1 + 2T + t_2$
Camino largo	Camino corto	$t_2 + 2T + t_1$
Camino largo	Camino largo	$t_2 + 2T + t_2$

Tabla 3.3: Caminos del interferómetro elegidos por un pulso durante los trayectos de ida y vuelta.

Ventajas e inconvenientes

La justificación de esta estrategia como una alternativa más interesante que los sistemas de dirección única se fundamenta en tres hechos:

1. En primer lugar, el sistema posee una desventaja aparente sobre las estrategias de una dirección única, y es que cada pulso ha de recorrer un trayecto de ida y vuelta, es decir, el doble de distancia que separa a los extremos de la comunicación, con la consiguiente pérdida de eficiencia por el incremento del ruido y la atenuación (doble trayecto). Pero esa desventaja

[29] Imagen extraída de "*Quantum Cryptography Training Course*" por Grégoire Ribordy.

no ocurre en un escenario real porque Bob no envía un pulso atenuado, sino un pulso intenso, que Alice atenúa a su salida para el recorrido del último trayecto.

2. El beneficio que aporta esta estrategia y que la convierte en un sistema potencialmente interesante se obtiene en la **sincronización**. Los pulsos que interfieren han transitado un mismo circuito, utilizando direcciones opuestas, lo que asegura que la distancia recorrida por ambos pulsos ha sido idéntica, siempre que cualquier variación se produzca en tiempos mayores de lo que tarda un fotón en recorrer el sistema, lo que es cierto en la práctica. Pero además, ese tránsito ha sido realizado de forma conjunta, puesto que ambos pulsos han estado separados una distancia equivalente a la diferencia entre los caminos corto y largo del interferómetro, distancia que en tiempo es prácticamente inapreciable. Esa magnitud de tiempo inapreciable la extendemos también al tiempo utilizado por ambos pulsos para completar los trayectos de ida y vuelta, dada la velocidad de la luz en la fibra, y el lapso necesario para que la fibra se contraiga o dilate por una modificación de las condiciones externas.

3. El esquema de doble dirección es más económico puesto que elimina las sincronizaciones y simplifica el diseño de los interferómetros en Alice y Bob. Por el contrario, es menos óptimo porque está obligado a no mezclar los pulsos intensos de ida con los débiles de llegada, por lo que tiene un "ciclo de larga": no puede emitir en contínuo, lo tiene que hacer en base a trenes de pulsos (hasta que un tren no a terminado su recorrido y finalizado su detección no se puede emitir otro tren). La longitud de un tren de pulsos está limitada por la longitud de una fibra de almacenamiento que tenemos que insertar dentro de Alice.

4. Otra ventaja de esta estrategia es que se reduce de forma considerable el número de componentes utilizados en uno de los extremos de la comunicación, concretamente en el extremo inicial, Alice. Teniendo en cuenta las características del sistema descrito, podemos conectar varios de estos Alice a un mismo Bob, abaratando el precio final de una red de distribución de claves por medio de conexiones compartidas. En los capítulos siguientes veremos como podemos implementar esta configuración, y justificaremos su utilización debido al hecho de que un sistema QKD está limitado por el proceso de destilación de la clave, lo que deja tiempo disponible para el intercambio a nivel físico de un único Bob con varios Alice.

El espejo de Faraday[30]

Con esta nueva estrategia necesitamos un nuevo componente para nuestro sistema que no hemos estudiado en los apartados anteriores. Ese nuevo componente es un espejo, encargado de reflejar los pulsos que llegan desde Bob.

El componente comúnmente utilizado es el espejo de Faraday, figura 3.15, que posee una particularidad destacable y es que tras la reflexión, el estado de

[30]Faraday Rotator Mirror.

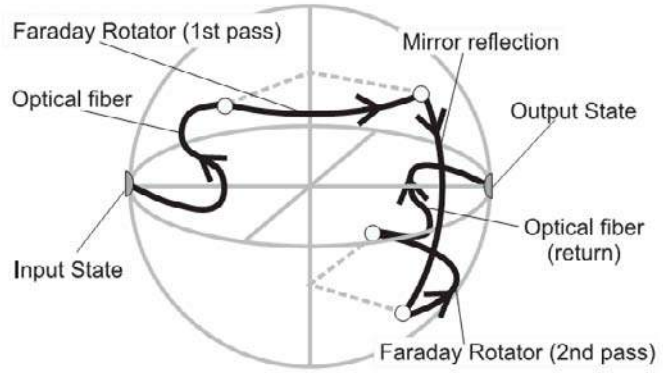


Figura 3.15: El espejo de Faraday. Se compone de un rotador de Faraday de 90º montado sobre un espejo.

la polarización rota 90º sobre la señal de entrada[31].

3.3.4. Fuente común

Finalmente, si lo que deseamos es construir un sistema de distribución cuántica de claves basado en el envío de pares entrelazados, la estrategia de configuración tiene que ser distinta. La diferencia radica en el hecho de un sistema basado en pares EPR utiliza una fuente de fotones común, que se situaría entre Alice y Bob para maximizar su alcance, como muestra la figura 3.16[32]. Esa fuente emite pares de fotones entrelazados hacia los extremos de la comunicación, donde el proceso de medida vuelve a ser similar al proporcionado por el interferómetro de Mach-Zehnder.

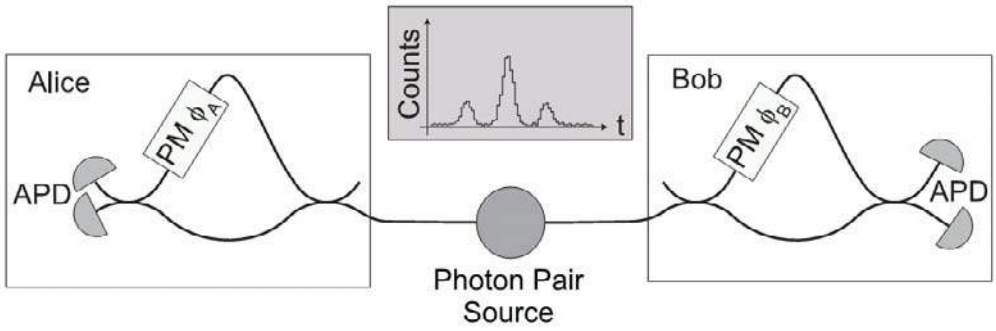


Figura 3.16: Esquema de configuración de un sistema QKD basado en pares EPR.

[31] Imagen extraída de [30], "*Quantum Cryptography*" por N. Gisin et al.
[32] Imagen extraída de [30], "*Quantum Cryptography*" por N. Gisin et al.

3.4. El sistema id-3000 de id-Quantique

La conclusión a la que llegamos después de estudiar los componentes del nivel físico, y las estrategias de configuración para un sistema QKD es la siguiente. La manera más sencilla de implementar una fuente de fotones individuales, es utilizar un emisor láser, medir la potencia del pulso generado a la salida de la fuente, y atenuar dicho pulso. Pero además, podemos integrar esta implementación de una fuente de fotones individuales con el diagrama de configuración de doble dirección mostrado en la sección anterior, obteniendo un sistema *plug & play*, fácil de sincronizar y adaptable a cualquier distancia entre los extremos de la comunicación. Esta es la solución implementada por los equipos id-3000 de id-Quantique que hemos utilizado en este proyecto, y que vamos a describir en este apartado.

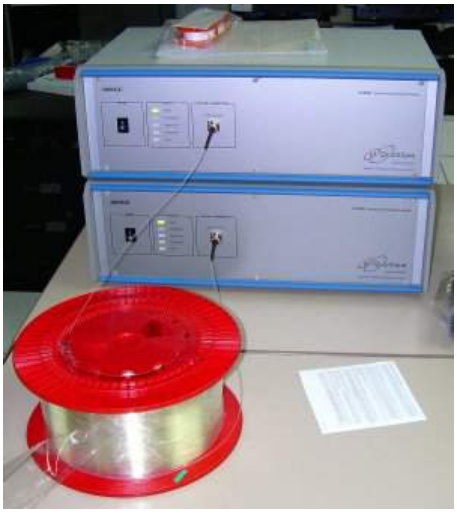


Figura 3.17: Los equipos QKDS-A y QKDS-B del sistema id-3000, y el rollo de fibra de 12,6 Km utilizado en las pruebas de laboratorio.

El sistema id-3000 se compone de dos equipos, QKDS-A y QKDS-B, que corresponden con los extremos de la comunicación, Alice y Bob respectivamente, y que podemos ver en la imagen 3.17. En la misma fotografía observamos el rollo de fibra, de una longitud aproximada de 12,6 Km, utilizado para conectar ambos equipos.

3.4.1. Arquitectura del sistema

QKDS-A

Como acabamos de comentar, el sistema de id Quantique funciona en lo que conocemos como modo *plug & play*, o de doble dirección, y por lo tanto, los principales componentes que podremos encontrar en Alice serán un espejo rotador de Faraday, un modulador de fase y un atenuador óptico variable, VOA. Estos componentes actuarán siguiendo un orden establecido sobre cada uno de los pulsos provenientes de Bob, donde la modulación y el reflejo del pulso

pueden intercambiar las primeras posiciones de la secuencia de actuaciones, mientras que la atenuación será siempre el paso final que produzca la emisión de un único fotón.

Ahora bien, a efectos prácticos la secuencia de funcionamiento del equipo QKDS-A es algo distinta a la descrita, ya que la atenuación se aplica por duplicado[33] (tanto a la entrada, como a la salida del pulso). Y además, encontramos otros componentes adicionales, como los utilizados para la sincronización de la llegada de pulsos, o la línea de retardo que veremos a continuación.

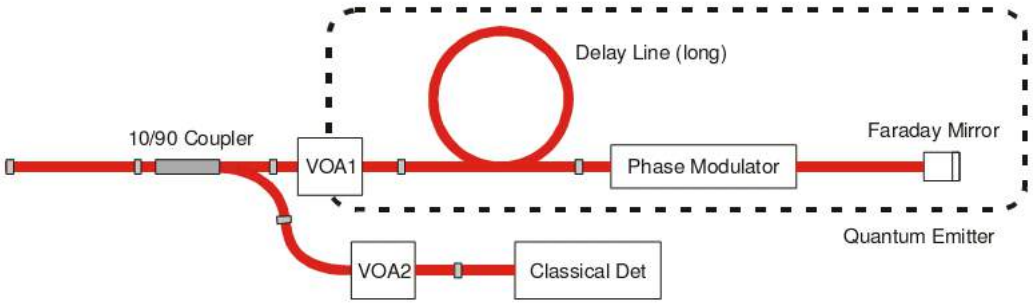


Figura 3.18: QKDS-A (Alice).

Si observamos la figura 3.18[34], a la entrada de Alice, el primer elemento que encontramos es un divisor del haz, 10/90, que provoca la separación de un 10% del pulso entrante hacia un detector clásico, pero antes de llegar al detector clásico, el pulso pasa por un atenuador variable que permite reducir la intensidad de dicho pulso. Este detector realiza una función de sincronización, facilitando la gestión del tiempo que permanece activo el modulador de fase, para la modificación de cada uno de los pulsos emitidos.

El otro elemento que encontramos es una línea de retardo de una longitud considerable, 12.6 Km, que se utiliza para almacenar de forma temporal todos los pulsos atenuados que regresan hacia Bob. La utilización de esta línea adicional de fibra se debe a que los pulsos atenuados reflejados en Alice pueden resultar indistinguibles de la radiación dispersa hacia atrás (*back scattering*), en la misma frecuencia, por los pulsos intensos provenientes de Bob. Teniendo en cuenta que la separación temporal entre dos pulsos es de 200 ns, lo que equivale a una distancia en la fibra de 40 m, el número máximo de pulsos que puede contener la línea interna es de 315. Pero además, puesto que los pulsos que vienen de Bob y los reflejados en Alice no tienen riesgo de "colisión" hasta la salida del atenuador, el rollo interno de fibra puede contener al mismo tiempo los pulsos de ida y vuelta, sumando una capacidad total de 630 pulsos. Este modo de trabajo que agrupa el envío de un número máximo de pulsos, incorpora la definición de un nuevo concepto: el de tren (*frame*), o tren de pulsos; que hace referencia a esa concentración de pulsos que se gestionan de forma segmentada.

Un parámetro adicional es la distancia que recorre un pulso dentro de Alice,

[33]Factor que debemos tener en cuenta a la hora de configurar los parámetros de atenuación, puesto que la atenuación total aplicada será el doble de la establecida en el atenuador.

[34]Imagen extraída de "*Quantum Key Distribution System id 3000 User Guide*" por id-Quantuique, June 2005.

sin tener en cuenta los 12,6 Km de línea de retardo que acabamos de comentar. Puede parecernos de poca relevancia, pero los 6 m de longitud que identifican ese segmento serán clave en el estudio de los pulsos reflejo que veremos en los capítulos finales de este proyecto.

QKDS-B

En el otro extremo de la comunicación utilizamos el equipo QKDS-B, que podemos ver en la figura 3.19[35], y en el que encontramos los elementos esperados: el emisor láser, los foto-detectores de avalancha (APD), el modulador de fase, y los caminos de distinta longitud para el interferómetro.

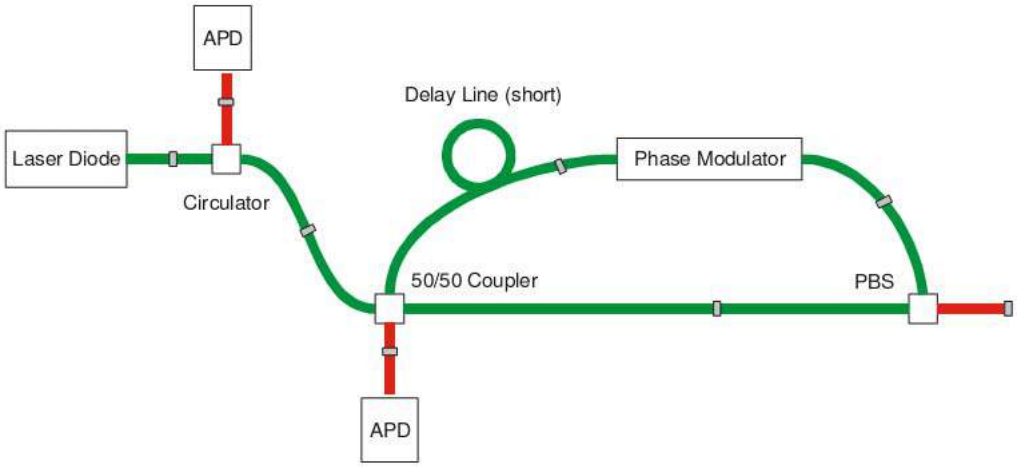


Figura 3.19: QKDS-B (Bob).

Un parámetro importante de la configuración en Bob es la diferencia entre los caminos corto y largo del interferómetro. Esa diferencia va a ser la que identifique la separación del pulso dividido[36] que viaja a través del canal de comunicación, y debe ser inferior a la distancia que separa a dos pulsos emitidos de forma consecutiva. Puesto que sabemos que la longitud del camino largo en Bob es de 17 m, y la longitud del camino corto es de 7 m, la separación entre las componentes de un pulso dividido viajando de ida y vuelta por la fibra será de 10 m, que como esperábamos es inferior a la separación entre dos pulsos, 40 m (200 ns). La figura 3.20 muestra la separación entre pulsos consecutivos (emitidos cada 200 ns) y pulsos divididos (al pasar por caminos distintos con una diferencia de 10 m).

3.4.2. Secuencia de funcionamiento

La secuencia de funcionamiento que siguen los equipos id-3000 es muy similar a la descrita para los sistemas de doble dirección, añadiendo algunas particu-

[35]Imagen extraída de "*Quantum Key Distribution System id 3000 User Guide*" por id-Quantuique, June 2005.

[36]En estado de superposición de los dos caminos.

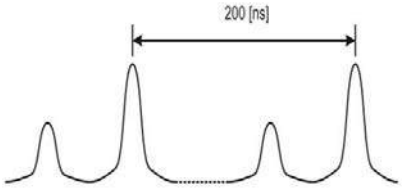


Figura 3.20: Separación entre pulsos divididos y consecutivos.

laridades que veremos en este apartado. El inicio de la comunicación se realiza en Bob, con la emisión de pulsos láser intensos cada 200 ns. Al analizar el equipo QKDS-A conocimos el concepto de tren (*frame*) que el sistema emplea en este instante. Y es que Bob sólo emite un número limitado de pulsos consecutivos, un tren, cuya longitud máxima[37] no debe superar dos veces la longitud de la línea de retardo incluida en Alice.

Antes de salir de Bob, los pulsos deben pasar por el interferómetro de Mach-Zehnder, donde se dividen para pasar por caminos de distinta longitud, y volver a unirse antes de salir de Bob, por lo que a la salida encontramos dos pulsos que recorren el canal de comunicación con una separación de 10 m, la equivalente a la diferencia entre los caminos largo y corto.

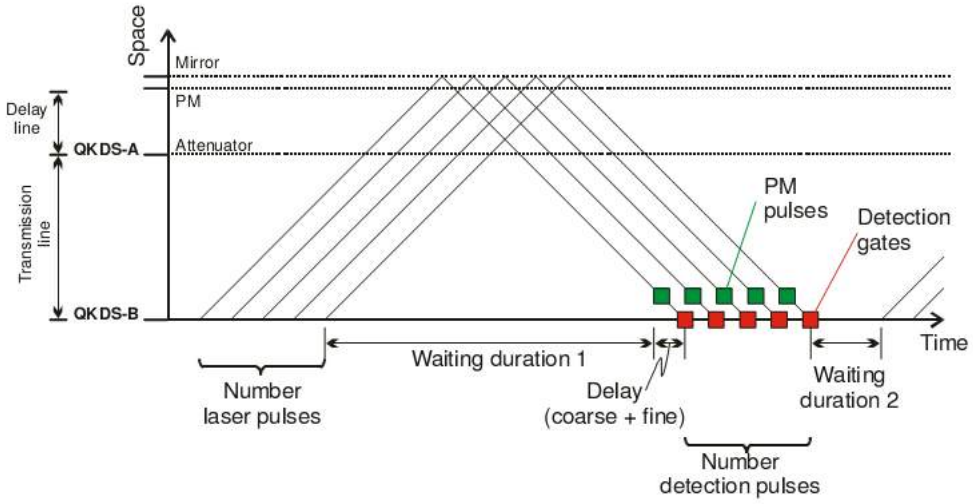


Figura 3.21: Diagrama espacio-tiempo del funcionamiento del sistema.

A la llegada a Alice, los pulsos son atenuados en la entrada y reflejados por el espejo de Faraday situado al final de la línea. A continuación, Alice modifica la fase del segundo de los pulsos divididos[38], y vuelve a aplicar la misma atenuación de la entrada, esta vez a la salida, de tal forma que el número de fotones a la salida de Alice seguirá una distribución de Poisson de media μ, en función de la atenuación elegida. En este punto, antes de que el primer pulso del tren salga de Alice, existirá un instante en el que todos los pulsos del tren se encuentran

[37] La longitud de un tren de pulsos es aquella que separa al primer y último pulso del tren.
[38] El pulso que pasó inicialmente por el camino largo.

dentro de Alice, dejando libre el camino de vuelta para los fotones individuales. Bob no emitirá más pulsos hasta haber realizado todas las detecciones correspondientes al tren.

Cuando los pulsos atenuados vuelven a Bob sufren la misma división que conocimos a la salida, por lo que al final del interferómetro encontramos tres instantes de llegada: en primer lugar llegarán los pulsos que tras dividirse escogieron el camino corto las dos veces que cruzaron el interferómetro, a continuación llegarán los pulsos que escogieron el camino largo y camino corto de forma alternativa, y finalmente llegarán los pulsos que atravesaron por dos veces el camino largo del interferómetro. El sistema tendrá en cuenta sólo aquellos pulsos que llegaron en segundo lugar, es decir, los que recorrieron caminos alternativos del interferómetro (sucesos indistinguibles que pueden producir interferencia). Antes de producirse la interferencia, Bob modificará la fase del pulso que pase por el camino largo, es decir, del pulso que haya llegado en primer lugar al interferómetro. Finalmente, en función del resultado de la interferencia se activará uno de los dos foto-detectores en Bob, con lo que sabremos el valor codificado por Alice siempre y cuando Alice y Bob hayan utilizado la misma base (ver tabla 3.1).

Las figuras 3.21 y 3.22 muestran en un diagrama de espacio-tiempo el recorrido de cada pulso desde su emisión en Bob, QKDS-B, pasando por Alice, QKDS-A, hasta su regreso de nuevo a Bob[39]. En la primera de las figuras, 3.21, podemos apreciar el momento en el que cada pulso sufre un cambio de fase aplicado por Alice o Bob, así como el secuenciamiento de los trenes de pulsos. Como se puede observar en el diagrama, Bob emite pulsos en un mismo tren cuidando que el último de los pulsos emitidos alcance Alice antes de que el primero de los pulsos salga de Alice. En la figura 3.22 se destaca la secuencia de activaciones de las puertas de detección en Bob, encargadas de comprobar el estado de los pulsos que completan el camino de vuelta.

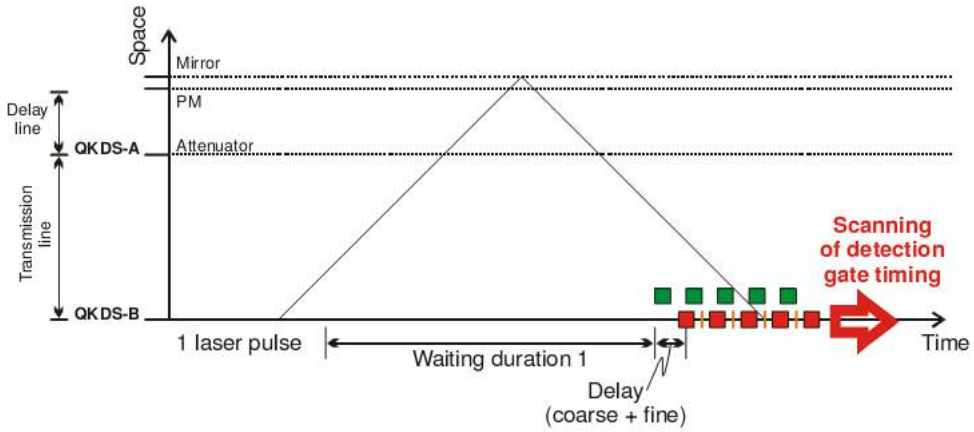


Figura 3.22: Diagrama en espacio-tiempo de la medición del canal cuántico.

[39]Imágenes extraídas de "*Quantum Key Distribution System id 3000 User Guide*" por id-Quantuique, June 2005.

Parámetros del sistema

Finalmente nos queda por conocer los parámetros del sistema más relevantes para su funcionamiento como pueden ser:

- La frecuencia de emisión de los pulsos de un tren es de 5 MHz, lo que equivale a una separación entre pulsos de 200 ns, o 40 m.
- La longitud de la fibra incorporada en el interior del equipo QKDS-A es de: 6 + 12600 = 12606 m.
- La fibra incorporada en el equipo QKDS-B se divide en función de los dos caminos del interferómetro de: 7 m y 17 m.
- La longitud que recorre un pulso a lo largo de todo el sistema depende de la longitud de la fibra que conecta a los equipos QKDS-A y QKDS-B. En nuestro caso, la fibra que une a ambos equipos es de 12,6 Km. Luego la distancia total recorrida por un pulso es:
  - Para el trayecto de ida (suponiendo que pasa inicialmente por el camino corto en el interferómetro de QKDS-B) la distancia recorrida es de: 7 + 12600 + 12600 + 6 = 25213 m.
  - Para el trayecto de vuelta (si inicialmente pasó por el camino corto, el pulso que nos interesa es el que vuelve por el camino largo) la distancia es: 6 + 12600 + 12600 + 17 = 25223 m.
  - La distancia total recorrida es de: 50436 m.
- Para la medición del tiempo que tarda un pulso en recorrer todo el sistema se utilizan tres parámetros, de mayor a menor amplitud (ver figura 3.23[40]):
  - El periodo de espera (*waiting period*). Se activa mediante señales de reloj a 20 MHz, por lo que mide la distancia recorrida por un pulso en intervalos aproximados de 10 m (50 ns).
  - El retardo grueso (*coarse delay*). Mide intervalos de 0,5 ns (aproximadamente 10 cm).
  - El retardo fino (*fine delay*). Mide intervalos de 20 ps (aproximadamente 4 mm).
- La atenuación óptica de un señal que atraviesa los distintos componentes del sistema es:
  - En QKDS-A, la atenuación de un pulso de ida y vuelta, es decir, que entra y sale del equipo sin tener en cuenta el atenuador variable es de: 34,84 dB.

[40]Imagen extraída de "*Quantum Key Distribution System id 3000 User Guide*" por id-Quantuique, June 2005.

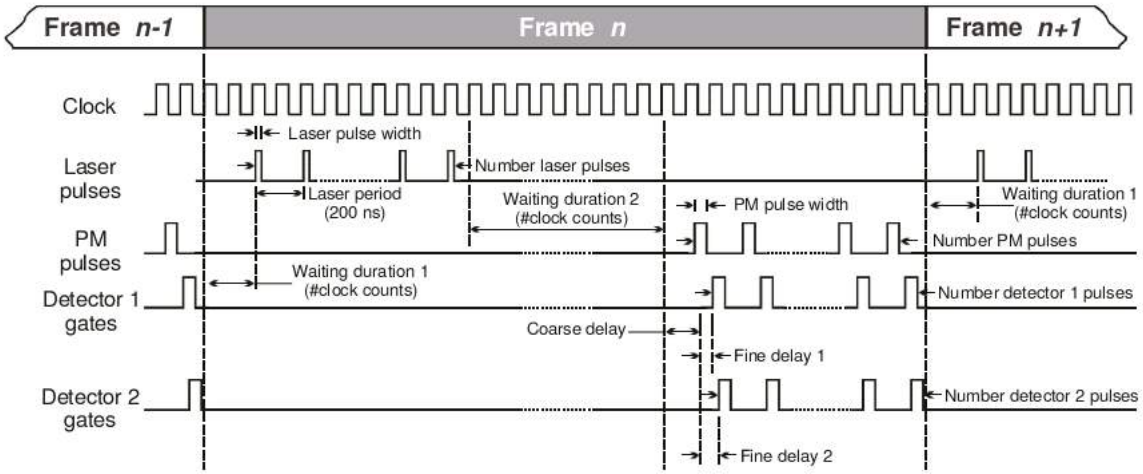


Figura 3.23: Parámetros de medición del sistema.

- En el equipo QKDS-B, las atenuaciones sufridas son: 4,1 dB a la salida del láser, 0,36 dB del circulador, 3,05 dB en el trayecto del camino corto del interferómetro y 5,42 dB en el camino largo.
- La atenuación del canal de comunicación que conecta a los equipos QKDS-A y QKDS-B oscila entre los 0,25 y 0,3 dB por kilómetro de fibra, y 0,15 dB por conector.

Calibrado

Para comenzar a trabajar con el sistema id-3000, lo primero que debemos hacer es medir la potencia a la entrada del equipo QKDS-A (Alice). Para hacer esto tenemos que fijar previamente la línea de fibra óptica que comunica a los extremos del sistema, y utilizando un medidor de potencia, obtener la intensidad del pulso que llega al extremo de la fibra que se conecta al equipo QKDS-A. Con el valor de la potencia a la entrada de Alice, $P_{AliceIn}$, podemos despejar, utilizando la siguiente expresión (3.2), el valor de la atenuación que tenemos que aplicar en Alice para obtener el número promedio de fotones por pulso deseado a la salida.

$$P_{AliceOut}(\mu) = P_{AliceIn} - 1,7dB - \tau_{Alice} - 2 \cdot Attenuation \tag{3.2}$$

Los parámetros del sistema que necesitamos conocer son:

- La atenuación de la señal dentro del equipo QKDS-A, es decir, dentro de Alice: τ_{Alice}. Según las especificaciones de los equipos utilizados proporcionada por el fabricante, la atenuación en Alice es de 34,84 dB.
- La potencia de la señal, o pulso que entra en Alice: $P_{AliceIn}$. En nuestro caso, el valor obtenido tras la medición es de 31,55 dB.
- La potencia deseada para el pulso que sale de Alice: $P_{AliceOut}$. Este valor dependerá del número promedio de fotones por pulso que deseamos obtener a la salida de Alice, μ, y de la energía de un fotón, E, como muestra la siguiente expresión:

$$\mu = \frac{10^{P_{AliceOut}/10}}{E}$$

Para calcular la energía, E, de un fotón que atraviesa nuestro sistema, utilizamos la ecuación:

$$E = \hbar\omega = h\nu = h\frac{c}{\lambda} \tag{3.3}$$

Donde ν es la frecuencia y ω la frecuencia angular[41], h es la constante de Planck[42], $h = 6{,}626 \times 10^{-34} J \cdot s$, c es la velocidad de la luz, $c = 300000 Km/s$, y λ la longitud de onda del fotón, para el caso que nos ocupa la longitud de onda utilizada es $\lambda = 1550nm$. Finalmente, para despejar el valor de la energía debemos tener en cuenta otros factores como son:

- La frecuencia a la que se emiten los pulsos de fotones: 5 MHz.
- Y el ancho de cada pulso: 1000 ps.

Despejando los parámetros de nuestro sistema en la expresión (3.3), el resultado obtenido para la energía de un fotón es:

$$E = \frac{6{,}626 \times 10^{-34} \cdot 3 \times 10^{8}}{1550 \times 10^{-9}} \cdot 5 \times 10^{6} \cdot 10^{3} = 6{,}41226 \times 10^{-10} W$$

[41] $\omega = 2\pi\nu$.

[42] Y $\hbar$ la constante de Dirac, con constante de Planck reducida, $\hbar = h/2\pi$.

Capítulo 4

Arquitectura

Si realizamos un repaso fugaz de lo que hemos visto en los capítulos anteriores, recordaremos cómo partimos de una brillante idea reflejada en un procedimiento de distribución segura de claves para, a continuación, estudiar la forma de construir el mecanismo que implemente esa idea. Ahora, debemos diseñar el sistema que ponga en práctica todo lo visto hasta el momento. En otras palabras, tenemos un conjunto de protocolos definidos a nivel teórico, una implementación a nivel físico capaz de realizar los pasos elementales exigidos por los protocolos, y nos queda por buscar la estructura sobre la que encajan todas esas piezas, así como las herramientas necesarias para unir esa estructura.

Comenzamos el diseño de nuestro sistema estudiando los pilares sobre los que se soporta el resto de la estructura, e identificamos dos ejes que se distinguen por la agrupación de los fundamentos teóricos o prácticos:

- Originalmente, emprendimos el desarrollo de un sistema de distribución cuántica de claves a partir de un **pilar teórico** que define el intercambio seguro de una clave. Ese fundamento teórico se encuentra reflejado en la definición de los protocolos QKD, y su desarrollo nos llevará a la creación de dos niveles, uno para el intercambio de claves desde un plano físico, y otro para la reconciliación de las bases utilizadas en los procesos de codificación y decodificación del nivel anterior.

- Pero esa burbuja en la que desarrollamos todo concepto teórico no tiene en cuenta las imperfecciones del mundo real. Y en consecuencia, al construir los dos primeros niveles de nuestro sistema nos encontramos con una serie de factores que alteran el resultado esperado. Esos factores son los errores, inevitables en cualquier sistema de comunicación, y que debemos corregir para que la información compartida sea idéntica. Estamos ante el desarrollo de un pilar fundamentado en la experimentación, un **pilar práctico**, del que dependerá una nueva estructura. El objetivo de esta estructura será el de gestionar la destilación, o depuración, de la clave intercambiada.

El enfoque bilateral que obtenemos con esta división estructural se puede justificar desde varios puntos de vista. Con una orientación didáctica mostramos, por un lado, la aplicación práctica de un desarrollo teórico y, por otro, las consecuencias de esa aplicación práctica. Desde el pragmatismo funcional, lo que

conseguimos es una división modular, que separa aquellos aspectos que deben recibir un tratamiento particular en su implementación. La estructuración de una idea, o de un sistema, es por lo general útil y beneficiosa para su implementación.

Pero, independientemente de la justificación que aportemos, el resultado final de nuestro estudio es una arquitectura con los niveles que mostramos a continuación:

1. Intercambio de una clave. Parte del objetivo de implementar un concepto teórico, protocolos como el BB84, B92 o SARG04, agrupando los procesos de:

   a) Intercambio de la clave en bruto, obtenida de la interpretación inicial de los fotones intercambiados entre ambos extremos de la comunicación, sin conocimiento previo de la base de medida.

   b) Reconciliación de bases. En este punto es donde obtenemos la primera clave "real" que, en ausencia de imperfecciones y sin la actuación de un agente externo[1], será la clave final del sistema.

2. Destilación de la clave intercambiada. Surge como consecuencia de las imperfecciones de una implementación práctica o la existencia de un atacante, y requiere de dos herramientas:

   a) Un mecanismo de corrección de errores que depure la primera clave idéntica en ambos extremos de la comunicación.

   b) Un soporte para la amplificación de la privacidad que asegure el mejor rendimiento del sistema.

Pero aún no hemos acabado con la estructura del sistema, puesto que nos hemos dejado una herramienta imprescindible para el estudio de su rendimiento real, y esa herramienta es la **entropía**. El problema que encontramos al estudiar la entropía es que no posee ningún rasgo específico que la identifique de forma estructural, es decir, que la sitúe en uno de los niveles de nuestro sistema. Por esta razón, hemos decidido incorporar su estudio entre los apartados de intercambio y destilación de una clave, donde introduciremos el concepto de entropía y su interés en este proyecto.

Desde el punto de vista práctico, trabajaremos con una arquitectura de cinco niveles, como muestra la figura 4.1, donde no realizamos distinción alguna entre los procesos de intercambio y destilación de una clave. Hemos añadido un nivel adicional, de autenticación, que no hemos comentado hasta el momento en la justificación de la arquitectura utilizada, pero su implementación es imprescindible debido a que surge de un supuesto ineludible: el canal público debe estar autenticado.

Como se verá, hemos tenido que diseñar una estructura de cinco niveles para poner en marcha una idea brillante, pero aparentemente sencilla. Esa estructura nos da una idea aproximada de la complejidad del sistema que pretendemos

[1] Un espía o atacante.

Figura 4.1: Arquitectura en niveles de un sistema QKD.

desarrollar. A continuación, pasamos a describir con detalle cada uno de estos niveles, justificando su conveniencia y estudiando las distintas estrategias que podemos utilizar para su implementación.

4.1. Intercambio de una clave

La propia naturaleza de los protocolos de distribución cuántica de claves distingue dos canales de comunicación. Privado o cuántico, público o convencional, ambos canales son necesarios para completar el intercambio de una clave.

Nosotros hemos decidido construir un nivel en nuestra arquitectura para cada uno de esos canales, dividiendo todo protocolo QKD en función del canal utilizado.

Podríamos haber justificado esta decisión como una consecuencia directa del medio empleado. Pero como veremos a continuación, esta justificación es más compleja, puesto que a la hora de realizar esa división hemos tenido en cuenta detalles como la configuración y gestión del canal de comunicación, o el procesamiento de la información transmitida.

4.1.1. Intercambio de una clave en bruto.

El primer nivel en la implementación de un sistema de distribución cuántica de claves, QKDS, utiliza los fundamentos de la física cuántica para completar el intercambio de una clave segura. Esto nos da pie a pensar en este nivel como una capa exclusivamente física, que podemos implementar utilizando los mecanismos descritos en el capítulo anterior. Pero como veremos a continuación, el desarrollo de esta capa requiere una organización a nivel lógico que gestione todo el proceso de intercambio: la configuración del canal privado, la codificación y decodificación de la clave transmitida, o la sincronización entre los extremos de la comunicación, son algunos de los aspectos que debemos administrar.

Esa organización es la que vamos a estudiar en este apartado, y lo haremos desde dos puntos de vista: el antes y el después al proceso de intercambio. Por un lado, estudiaremos cómo debemos preparar el sistema antes de comenzar el intercambio de claves. Para después, una vez completado el intercambio de una

clave, estudiar el impacto de los resultados sobre el comportamiento del sistema. En otras palabras, realizaremos una retroalimentación del sistema, utilizando los resultados generados para recalibrar la ejecución del siguiente intercambio.

Sincronización

Es evidente que dos interlocutores no pueden comunicarse si no hablan el mismo idioma. Pero además, el hecho de que ambos interlocutores utilicen el mismo lenguaje no implica que se comprendan el uno al otro. Lo que nos lleva a la conclusión de que los extremos de una comunicación necesitan acordar: el idioma en el que desean hablar, el canal por el que van a comunicarse, y el grado de volumen que tienen que utilizar para ser escuchados.

Pues bien, si recurrimos a la analogía que pueda presentar nuestro sistema con respecto a otra comunicación, descubrimos tres procesos a realizar antes de comenzar la distribución de claves.

1. **Acuerdo de los parámetros de ejecución**.

   En un QKDS, el principal parámetro que Alice y Bob deben pactar es el tamaño de la clave[2]. La manera más sencilla de negociar este parámetro es que uno de los extremos comunique al otro su decisión, es decir, que Alice notifique a Bob el tamaño de la clave a intercambiar, o viceversa. En ese instante, ambos extremos pueden generar las secuencias de bases y valores aleatorios necesarios para realizar el intercambio. Finalmente, el extremo que recibe el tamaño acordado de la clave, responde con un mensaje de confirmación. En ese momento, ambos extremos de la comunicación están sincronizados y listos para comenzar el intercambio de una clave.

   En este punto, es importante aclarar un detalle. Cualquier sincronización o intercambio de información debe realizarse a través de un canal de comunicación público, como es evidente, cumpliendo siempre la exigencia de utilizar un canal público autenticado. Tanto el intercambio de parámetros, como la sincronización que se deriva de ese intercambio, están relacionados con la seguridad del sistema. Es fundamental que Alice y Bob conozcan de antemano cual va a ser el tamaño inicial de la clave que se pretende intercambiar.

2. **Establecimiento de una línea**.

   Recordemos que un requisito imprescindible de nuestro sistema es la utilización de una línea de comunicación dedicada. Esto nos obliga a establecer una ruta privada antes de comenzar el intercambio en aquellos entornos donde la línea puede ser compartida.

   En los capítulos finales de este proyecto, estudiaremos la forma de integrar los sistemas de distribución cuántica de claves en los entornos de cifrado actuales, lo que involucra el uso de redes de distribución compartidas.

[2]En los equipos id-3000 utilizados, el tamaño de la clave se define en función de dos parámetros: el número de trenes a enviar y el número de pulsos por tren; $\ell_{clave} = n_{trenes} \cdot n_{pulsos/tren}$. La razón por la que utilizamos dos parámetros es estructural, y se justifica en el uso de un rollo de fibra adicional, dentro de Alice, para almacenar todos los pulsos enviados en un tren antes de que ese tren vuelva a Bob, y evitar así los problemas derivados del backscattering en la fibra.

Es entonces cuando tendremos que incorporar a nuestro sistema la implementación de un mecanismo que nos permita el establecimiento de esas rutas o caminos privados.

3. **Medición de la línea**.

   Finalmente, otro requisito imprescindible para la correcta ejecución de este nivel, y en consecuencia de los siguientes niveles de la arquitectura, es la sincronización de la línea de comunicación que conecta a los extremos del sistema, Alice y Bob.

   Una medición precisa de la línea nos permite conocer con exactitud los tiempos en los que nos debe llegar la información. Cualquier suceso fuera del margen de tiempo esperado puede ser el indicio de un posible ataque. En consecuencia, una mala medición de la línea puede llevar a la sospecha, no justificada, de posibles ataques.

   En uno de los capítulos finales de este proyecto estudiaremos las vulnerabilidades de nuestro sistema, y entre esas vulnerabilidades descubriremos algunos factores que pueden afectarnos en la medición de la línea. Factores "naturales", no provocados por ningún atacante, como pueden ser los pulsos reflejo (que estudiaremos como pulsos fantasma).

Calibrado

Todos los procedimientos que acabamos de ver en la sincronización se ejecutan antes de comenzar el intercambio de una clave. Ahora bien, la medición de la línea es un proceso costoso que podemos evitar en una ejecución prolongada de los intercambios. Es decir, podemos considerar que durante un número de transmisiones determinadas las condiciones de nuestra línea no se han modificado, y en consecuencia, podemos completar nuevos intercambios sin necesidad de realizar una medición continua de la línea.

La base de este razonamiento es correcta, y su aplicación se traducirá en una pequeña pérdida de eficiencia, que se irá incrementando conforme varíen las condiciones de la línea, lo cual no sucede de forma brusca. Transcurrido un cierto tiempo, la variación de las condiciones será lo suficientemente importante como para tener que realizar una nueva medición de la línea. Con esta forma de trabajo debemos estimar un punto de equilibrio entre la pérdida de eficiencia, por el tiempo transcurrido desde la última medición de línea, y el coste asociado a un nuevo proceso de medición.

Ahora bien, podemos hacernos una pregunta inmediata ¿existe alguna forma de conocer hacía donde tienden esas variaciones de la línea? A la que respondemos con una gráfica, 4.2, donde mostramos la curva que sigue la probabilidad de detectar un fotón en un espacio de tiempo limitado. Como era de esperar, la probabilidad de encontrar un fotón en un punto determinado sigue distribución normal, o gaussiana, con una anchura de 40 ps equivalente a unos 0,8 m.

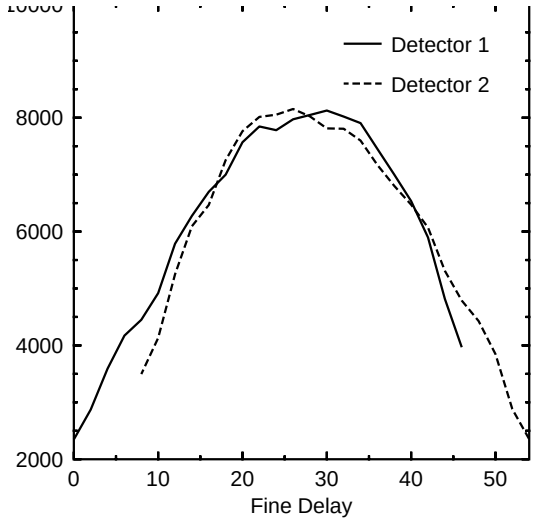


Figura 4.2: Distribución de la probabilidad de detectar un fotón en función del tiempo.

Análisis de los resultados

Ya en este primer nivel podemos analizar la fiabilidad de la información[3] compartida por Alice y Bob. Conocemos la distancia (aproximada) que separa los extremos de la comunicación, las pérdidas de potencia que se producen a lo largo de la línea, así como la precisión de los equipos utilizados. Y a partir de estos parámetros, podemos estimar el número de detecciones esperadas en cada extremo de la comunicación en función del número de pulsos transmitidos. Comparando los resultados obtenidos de la ejecución, con la estimación teórica de los mismos, podemos detectar fluctuaciones en el rendimiento final del sistema, que pueden estar relacionadas con un posible ataque.

La interpretación de los resultados será de la forma:

1. Si el número de detecciones obtenidas es inferior al número de detecciones esperadas, podemos encontrarnos ante dos situaciones:

   a) La línea no está bien sincronizada. Un error de calibrado o una variación de las condiciones de la línea pueden haber provocado esta pérdida de sincronía.

   b) Un atacante está interrumpiendo la transmisión de un cierto número de pulsos. En ataques como el *photon number splitting*, PNS, el espía puede estar interesado en dejar pasar sólo aquellos pulsos en los que la fuente produce más de un fotón, interrumpiendo el paso del resto de fotones o parte del mismo. Pero también podemos imaginar otras estrategias en las que un atacante decida aislar parte de la información intercambiada a través del canal cuántico, con el objetivo de realizar un análisis estadístico de los resultados obtenidos, o por el conocimiento de cualquier tipo de vulnerabilidad en nuestro sistema.

[3] La clave.

2. Si por el contrario, la situación es tal que el número de detecciones obtenidas es superior al número de detecciones esperadas, nos encontramos ante un problema más serio, cuyas posibles causas son:

   a) Estamos recibiendo el reflejo de un pulso[4], bien porque la línea está mal sincronizada, o porque dicho pulso reflejo[5] interfiere sobre el pulso esperado.

   b) Un presunto espía interrumpe todos o parte de los pulsos enviados, realizando una retransmisión de los pulsos interceptados. Las razones por las que se incrementa el número de pulsos detectados pueden ser dos: un error de precisión en el ataque, o el espía intenta asegurarse de alguna forma que todos los pulsos interceptados lleguen a su destino, con el objetivo de incrementar las detecciones en esos pulsos.

3. Finalmente, tenemos un tercer parámetro a estudiar como son las *detecciones dobles*. Este fenómeno aparece cuando en un pulso se activan al mismo tiempo los dos detectores, y puede ocurrir por dos razones:

   a) La imprecisión de un detector que se activa aún cuando no ha recibido nada. Es lo que conocemos como detección oscura, su probabilidad es constante y depende de la tecnología de detección utilizada, así como de algunos parámetros de la configuración (por ejemplo, el *dead time*: tiempo muerto en el que un detector es sometido con la finalidad de disipar cualquier carga atrapada).

   b) Dos pulsos con distinto resultado de interferencia han llegado en intervalos muy cercanos en el tiempo, provocando la activación prácticamente simultanea de los dos detectores.

   Pues bien, cualquier variación en el número esperado de detecciones dobles es una razón de sospecha inmediata, puesto que, una detección doble sólo se produce por el mal funcionamiento de nuestro sistema. Por lo tanto, nunca puede reducirse el número de detecciones dobles, dado que el número de detecciones oscuras es un parámetro constante en las características de nuestros equipos. Y un incremento, supondría la llegada de pulsos no esperados.

4.1.2. Reconciliación de bases.

Una vez que Alice y Bob han intercambiado una clave a nivel físico, utilizando un canal de comunicación privado, ambos interlocutores deben ponerse

[4]Cualquier componente que interfiera en la línea de transmisión puede provocar dos efectos (ambos relacionados). Por un lado, una pérdida de potencia en la transmisión, y por otro, la reflexión de una parte de los pulsos transmitidos. El segundo de los efectos se agrava cuando empleamos de forma conjunta pulsos láser atenuados con una estrategia de doble dirección, puesto que los reflejos producidos durante el trayecto inicial tendrán una potencia suficiente como para reflejarse varias veces con una intensidad que puede enmascarar la señal cuántica de un sólo fotón.

[5]También conocido como pulso fantasma.

de nuevo en contacto, ahora por medio de un canal público autenticado, para intercambiar las bases utilizadas en la codificación y decodificación de la clave intercambiada. El resultado de este proceso es una nueva *clave reconciliada*, que en condiciones ideales[6] debe ser idéntica en ambos extremos de la comunicación.

El tamaño de las secuencias de bases utilizadas por Alice y Bob es idéntico al tamaño de la secuencia de valores codificados por Alice en la clave transmitida. Luego la información que tienen que intercambiar Alice y Bob en el proceso de reconciliación de bases es el doble que en el paso anterior, pero además, nos encontramos con el hecho de que la comunicación se realiza a través de un canal público, y por lo tanto compartido, por lo que en condiciones normales supone el primer cuello de botella de nuestro sistema.

Indexado de las bases

En la práctica, el número de fotones detectados en Bob es considerablemente inferior al número de fotones codificados y enviados por Alice[7], por lo tanto, no es necesario que Alice y Bob intercambien toda la secuencia de bases utilizadas. Podemos optimizar el tamaño total de la información intercambiada enviando tan sólo las bases empleadas en los pulsos donde Bob ha obtenido alguna detección. Esta estrategia de indexado requiere una ligera modificación del proceso de reconciliación de bases descrito en los protocolos originales, puesto que ahora, no es Alice quien inicia el proceso de reconciliación, sino Bob, que es quien conoce el índice de los pulsos donde se han producido detecciones.

La siguiente desigualdad (4.1) permite calcular una cota, en función de la diferencia existente entre el número de detecciones, n_{det}, y el número de pulsos intercambiados, n_{pulsos}, por debajo de la cual esta estrategia es mejor que la transmisión de la secuencia completa de bases utilizadas.

$$n_{pulsos} > n_{det}\,(1 + \lceil \log_2 n_{pulsos} \rceil) \tag{4.1}$$

Donde $\lceil \log_2 n_{pulsos} \rceil$ es el tamaño del registro que necesitamos para almacenar el índice de cada uno de los pulsos detectados, al que añadimos un bit adicional donde guardar el valor detectado.

Por otro lado, si calculamos el número de detecciones a partir del número de pulsos y la probabilidad de detección, $n_{det} = n_{pulsos} \cdot p_{det}$, obtenemos una expresión equivalente (4.2) que acota el número de pulsos en función de de la probabilidad de detección.

$$n_{pulsos} < 2^{\frac{1-p_{det}}{p_{det}}} \tag{4.2}$$

[6]Consideramos que nuestro sistema se ejecuta bajo *condiciones ideales* cuando tenemos la certeza de que la información intercambiada a través del canal de comunicación privado no ha sido modificada, lo que es equivalente a afirmar que trabajamos en un entorno sin errores. Es decir, bajo ese entorno ideal desaparecen las imperfecciones de los equipos utilizados para la comunicación, el ruido del canal, y los posibles espías.

[7]A modo de ejemplo, en una implementación con pulsos láser atenuados, la probabilidad de obtener un fotón a la salida de Alice sigue una distribución de Poisson de media μ. A lo que tenemos que añadir las pérdidas en el canal de comunicación, las pérdidas dentro de Bob, y la eficiencia de los detectores. Lo que supone una probabilidad de detección final bastante baja.

Reindexado. Indexado incremental

Sabemos que la distribución de las detecciones debe ser perfectamente aleatoria, y de esa aleatoriedad radica buena parte de la seguridad de nuestro sistema, ya que de no ser así, un posible espía podría decidir en qué intervalos es más interesante realizar una escucha o intercepción.

Evidentemente, esa aleatoriedad también afecta a los interlocutores, impidiendo realizar una estimación del instante con mayor probabilidad de detectar un pulso. Ahora bien, la razón por la que nos interesa conocer los periodos de probabilidad máxima de una detección es la siguiente. Si conocemos el intervalo en el que va a llegar una detección, o lo que es lo mismo, el espacio promedio de tiempo que separa dos detecciones, podemos mejorar la estrategia de indexado descrita anteriormente, realizando lo que se conoce como indexado incremental o diferencial. Esta nueva estrategia modifica el índice de cada suceso, calculando la diferencia de un suceso con su predecesor, y almacenando dicha diferencia en lugar del índice original. El resultado es una secuencia de índices de menor tamaño, especialmente útil si los sucesos se producen en intervalos relativamente acotados, y el índice del último suceso es holgado.

A modo de ejemplo, imaginemos un sistema que no funcione de la forma esperada, de manera que se produce una detección en intervalos aproximados de 200 pulsos. Si el número total de pulsos enviados es de 1.000.000, utilizando la estrategia definida anteriormente, necesitaría un índice de 20 bits ($2^{20} > 1000000$) para registrar cada uno de los pulsos detectados. Ahora bien, puedo recalcular el índice de cada detección sustrayendo el índice de la detección anterior, lo que genera índices de tamaño aproximado a la separación entre detecciones, que en este sistema imperfecto es de 200 pulsos, y para lo cual, necesito tan sólo un registro de 8 bits por índice de detección ($2^8 > 200$).

La mejora obtenida en la segunda estrategia es considerable. Puesto que las detecciones se producen en intervalos de 200 pulsos (aproximadamente) y el número total de pulsos enviados es de 1.000.000, el número esperado de detecciones será de 5.000. Esto supone que, en la primera estrategia necesitamos un total de $5000 \cdot (20 + 1) = 105000$ bits, mientras que en la segunda estrategia tan sólo necesitamos $5000 \cdot (8 + 1) = 45000$ bits.

Volviendo a nuestro sistema, buscamos la forma de acotar un intervalo que nos permita aplicar la estrategia recién comentada. Para ello, partimos del conocimiento de que la probabilidad de detectar un número determinado de fotones sigue una distribución binomial, como muestran las curvas de la gráfica 4.3.

Esta distribución nos permite calcular la probabilidad de obtener un número determinado de detecciones, n_{det}, en función de la probabilidad de detección, p_{det}, y el número total de pulsos enviados, n_p.

$$Bin(n_p, p_{det}) = \begin{pmatrix} n_p \\ n_{det} \end{pmatrix} p_{det}^{n_{det}} \, (1 - p_{det})^{n_p - n_{det}} \tag{4.3}$$

Esto nos sugiere, en primer lugar, calcular una cota para la cual, la probabilidad de obtener k detecciones es inferior a la probabilidad de obtener k+1 detecciones.

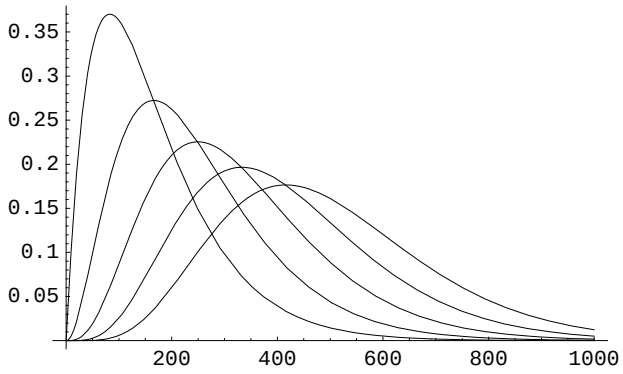


Figura 4.3: Distribuciones de probabilidad para la obtención de 1 a 5 detecciones, con una probabilidad de detección del 1,2 %.

$$\left(\begin{array}{c} n_p \\ k \end{array} \right) p_{det}^{k} \left(1 - p_{det}\right)^{n_p - k} < \left(\begin{array}{c} n_p \\ k+1 \end{array} \right) p_{det}^{k+1} \left(1 - p_{det}\right)^{n_p - (k+1)} \tag{4.4}$$

$$\frac{n_p!}{k!\,(n_p - k)!} p_{det}^{k} \left(1 - p_{det}\right)^{n_p - k} < \frac{n_p!}{(k+1)!\,(n_p - (k+1))!} p_{det}^{k+1} \left(1 - p_{det}\right)^{n_p - k - 1}$$

$$\frac{1}{n_p - k} p_{det}^{k} \left(1 - p_{det}\right)^{n_p - k} < \frac{1}{(k+1)} p_{det}^{k+1} \left(1 - p_{det}\right)^{n_p - k - 1}$$

$$\frac{1}{n_p - k} \left(1 - p_{det}\right)^{n_p - k} < \frac{1}{(k+1)} p_{det} \left(1 - p_{det}\right)^{n_p - k - 1}$$

$$1 < \frac{n_p - k}{k+1} \cdot \frac{p_{det}}{1 - p_{det}}$$

$$n_p > k + (k+1)\,\frac{1 - p_{det}}{p_{det}} \tag{4.5}$$

Donde se ha tenido en cuenta que $p_{det} \geq 0$ y $1 - p_{det} \geq 0$.

Si analizamos el resultado que muestra la ecuación 4.5, para una probabilidad de detección constante, $p_{det} = cte.$, los saltos en los que es más probable obtener k+1 detecciones son lineales. Y por lo tanto, podemos estimar un intervalo promedio de detección, con una probabilidad de detección máxima.

Para estimar el tamaño de ese intervalo, ℓ_{int}, calculamos la diferencia entre dos saltos cualesquiera.

$$\ell_{int} = n_{p_i} - n_{p_{i+1}} = k + (k+1)\,\frac{1 - p_{det}}{p_{det}} - (k-1) - k\frac{1 - p_{det}}{p_{det}}$$

$$\ell_{int} = 1 + \frac{1 - p_{det}}{p_{det}} \tag{4.6}$$

Qué como esperábamos, es una constante.

Tiempo muerto (*dead time*)

Aún podemos añadir una optimización adicional a nuestro sistema. Sabemos que los detectores de fotones individuales que utilizamos[8] poseen una limitación estructural, por la cual el detector permanece inutilizado durante un periodo de tiempo "considerable", conocido como *dead time*, o tiempo muerto. Durante ese intervalo de tiempo tenemos la certeza plena de que no se va a realizar ninguna detección, puesto que el detector no se encuentra activo, y en consecuencia podemos descontar ese tiempo muerto del índice incremental recién descrito.

Por ejemplo, en los sistemas id-3000 que estamos utilizando, esa franja de tiempo muerto de los detectores es de $10\mu s$, por lo que teniendo en cuenta que cada pulso se envía en intervalos de 200 ns, tenemos la seguridad de que en los siguientes 50 pulsos no se producirá ninguna detección. Esto ocurrirá así en todo momento excepto en la primera de las detecciones, que no se ve afectada por el tiempo muerto. Podemos así reducir en 50 unidades el índice incremental de cada detección, lo que equivale a utilizar cinco bits menos para el registro de almacenamiento del índice.

Observación: Como vimos en el capítulo anterior, el tiempo muerto es una característica presente en todos los fotodetectores de avalancha, APD. Su gestión es imprescindible para conseguir un funcionamiento óptimo de estos dispositivos, y provocará un cambio importante en el comportamiento de nuestro sistema QKD. En el capítulo anterior conocimos una de las consecuencias del tiempo muerto, como es la pérdida de rendimiento. Ahora debemos tener presente una segunda consecuencia, y es que la distribución de las detecciones obtenidas no va a ser constante a lo largo del tiempo. En un sistema QKD donde los fotodetectores requieren de la aplicación de un tiempo muerto, la probabilidad de obtener una detección se ve influenciada por la probabilidad de que los detectores se encuentren en un estado de bloqueo, debido a una detección previa cercana en el tiempo. La probabilidad intrínseca de obtener una detección es constante para cada tipo de detector en un entorno definido, pero la probabilidad de que el sistema se encuentre bloqueado (por el tiempo muerto) va a depender del tiempo de ejecución. Cuando un sistema QKD se pone en marcha no existen detecciones previas cuyo tiempo muerto afecte a la probabilidad final de detección, por lo tanto la distribución de las detecciones será mayor al comienzo (en un sistema de doble dirección, la distribución de detecciones será mayor al inicio de cada tren de pulsos). Este hecho nos presenta la posibilidad de mejorar el mecanismo de reindexado recién propuesto, teniendo en cuenta el efecto que supone el tiempo muerto sobre la distribución de las detecciones.

4.2. Entropía y error

Antes de meternos de lleno en el proceso de destilación de una clave, vamos a estudiar algunos conceptos que nos van a ser de utilidad en ese futuro proceso

[8] Fotodiodos de avalancha, APD.

de filtrado o destilación de la clave.

4.2.1. Entropía de la clave reconciliada

Si nuestro sistema funciona de la forma esperada la entropía de la clave reconciliada debe ser máxima, es decir, no debe existir ningún parámetro de la clave que nos pueda proporcionar información acerca de la misma. En consecuencia, tenemos que:

- Los valores de la clave deben ser equiprobables, y su distribución perfectamente aleatoria.
- El tamaño de la clave debe ser el esperado.
- Y el número de bases coincidentes, o el tamaño de la clave reconciliada debe ser el esperado. Recordemos que para el protocolo BB84 podemos hablar de bases coincidentes, pero no así con otros protocolos como el SARG04. En consecuencia, debemos utilizar otros parámetros, como el porcentaje esperado de clave reconciliada frente a la tamaño de la clave en bruto, que para el protocolo BB84 debe ser del 50 %, mientras que para otros protocolos como el B92 y SARG04 será tan sólo del 25 %.

Cuando trabajamos con una única clave el número de muestras puede ser insuficiente para concretar un resultado fiable, por lo que el estudio de la entropía de nuestro sistema debe prolongarse con cada intercambio. En ese estudio, debemos calcular el grado de aleatoriedad de la clave intercambiada comparando las cantidades de ceros y unos contenidos en cada clave. La distancia entre ambas cantidades debe ser cercana a cero, o lo que es lo mismo, la probabilidad de encontrar un cero o un uno en una clave debe ser cercana al 50 %. De igual forma, el número de valores desechados de la clave en bruto debido a la elección de una base incorrecta también debe ser próximo a los porcentajes recién comentados (50 % para el BB84 y 25 % para el B92 y SARG04).

4.2.2. Error cuántico o ruido. QBER[9]

Una magnitud importante es siempre la que nos anticipa los resultados esperables en promedio de un sistema. El estudio del QBER de nuestro sistema va a ser uno de los aspectos que mejor lo caractericen. En un sistema de distribución cuántica de claves utilizamos el término QBER cuando hacemos referencia a la tasa de errores del sistema, que definimos como:

$$QBER = \frac{N_{Erroneos}}{N_{Totales}} = \frac{N_{Erroneos}}{N_{Correctos} + N_{Erroneos}}$$

Donde el número de bits totales, $N_{Totales}$, corresponde con el número de bits obtenidos tras la reconciliación de bases del protocolo utilizado. Es decir, es el número de bits de clave obtenido antes de la corrección de errores y la amplificación de la privacidad.

[9] *Quantum Bit Error Rate*: tasa de error de bits cuánticos.

Ahora bien, para el estudio de la tasa de errores, QBER, la aproximación que utilizamos es la descrita por la siguiente expresión (4.7), donde aparecen descritas las dos componentes más importantes de este error: la componente óptica y la de detección; que estudiaremos a continuación.

$$QBER = QBER_{opt} + QBER_{det} \tag{4.7}$$

De esas dos contribuciones, la primera, el QBER óptico, es independiente de la distancia. Su valor obedece de forma exclusiva a la "calidad óptica" de los componentes utilizados en el sistema. Para el caso que nos ocupa, donde nuestro sistema de comunicación utiliza un mecanismo de codificación basado en la fase, la tasa de error óptico depende únicamente de la visibilidad[10], V, como muestra la siguiente expresión:

$$QBER_{opt} = \frac{1 - V}{2}$$

A modo de ejemplo, comentar que una visibilidad típica del 98 % se traduce en un error óptico del 1 %.

Por otro lado, la segunda de las contribuciones, el QBER de detección, sí es dependiente de la distancia debido a que las detecciones erróneas, también conocidas como conteos oscuros o *dark counts*, se mantienen constantes para cualquier longitud de la fibra utilizada. De esta forma, conforme aumenta la distancia entre los extremos del sistema, el número de detecciones disminuye (por pérdidas en el canal de comunicación, la fibra), mientras que, como acabamos de comentar, el número de conteos oscuros permanece constante, incrementando su grado de influencia con la disminución de las detecciones. Una expresión analítica de lo que acabamos de comentar es la siguiente:

$$QBER_{det} = \frac{p_{dark}}{p_{det} + 2p_{dark}}$$

Donde p_{dark} es la probabilidad de obtener un conteo oscuro, y p_{det} es la probabilidad de detección. Una vez más, a modo de ejemplo podemos fijar un valor aproximado para la probabilidad de los conteos oscuros en la magnitud de los 10^{-5} conteos por cada intento de detección. Para ajustar con mejor detalle este parámetro, encontramos una opción, *Noise measurement*, dentro de la aplicación proporcionada por el distribuidor de nuestro sistema, que nos permite obtener un valor bastante aproximado de la tasa de ruido de los detectores (fuente original de los conteos oscuros). La siguiente imagen, 4.4, muestra una captura de los resultados obtenidos al realizar una medición del ruido en nuestro sistema.

Observación: Las definiciones recién descritas para el cálculo de la tasa de error, QBER, son sólo una aproximación del mismo. Para profundizar en el desarrollo analítico de estos parámetros recomendamos la lectura de las siguientes referencias: [30] y [29].

[10] La visibilidad es una medida de la cantidad o calidad de la interferencia, que dependerá de los desplazamientos de fase de los fotones que atraviesan los distintos caminos del interferómetro. En algunas referencias encontrará alusiones a este término, la visibilidad, como contraste de interferencia o contraste de fase.

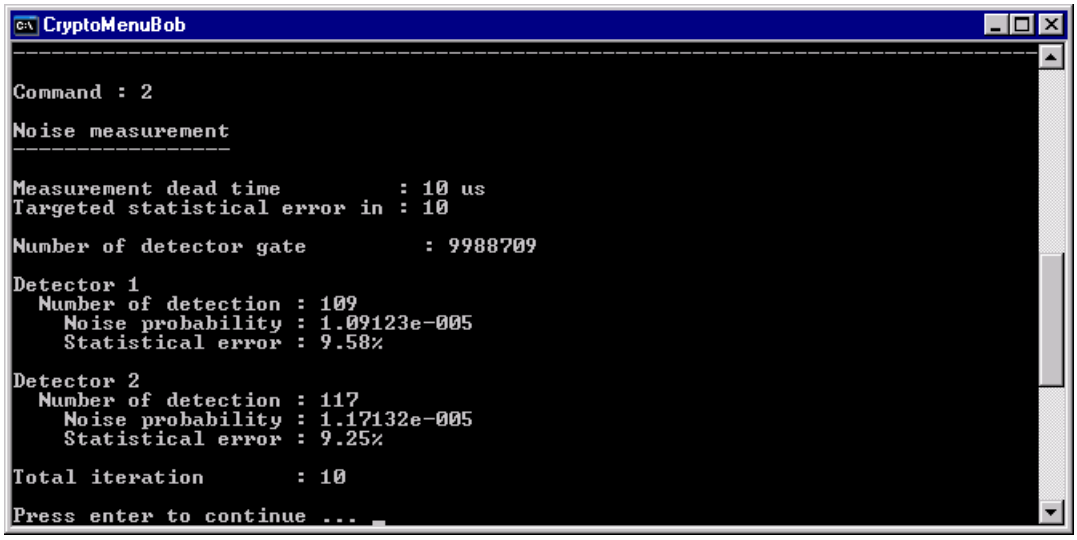


Figura 4.4: Medición del ruido por conteos oscuros con la aplicación *cryptomenu* del sistema id-3000.

Sólo como curiosidad queremos comentar la existencia de una tercera contribución a la tasa de error, $QBER_{acc}$, que podemos encontrar en algunas referencias de criptografía cuántica, cuyo estudio no ha sido desarrollado en este proyecto debido a que su origen se fundamenta en la pérdida de correlación entre pares de fotones, que únicamente aparece en los sistemas basados en pares entrelazados, lo que no es nuestro caso.

Finalmente, a pesar de haber desarrollado analíticamente toda la teoría que estudia el origen y causa de los errores en nuestros sistemas, hemos completado ese trabajo con una simulación práctica[11] de la ejecución de un sistema QKD a distintas distancias. Con esta simulación conseguimos mostrar la distribución del QBER como población (figura 4.5), observando su dispersión y evolución en función de la distancia.

Simulación de distintas distancias para el canal de comunicación

Antes de continuar con el estudio de la arquitectura y comportamiento de un sistema QKD, debemos detenernos para realizar una exposición detallada de cómo hemos realizado las simulaciones. Las razones son dos: por un lado la limitación impuesta por el uso de una línea de comunicación de longitud constante, y por otro, la necesidad de contrastar los resultados de la ejecución de nuestro sistema sobre distintas distancias. Afortunadamente, debido a las característi-

[11] Podemos considerar ambigua la utilización conjunta de dos términos como "simulación práctica", pero la manera con la que obtenemos los resultados que mostramos a continuación, justifica el empleo de ambos términos. Por un lado, nos encontramos ante la exposición de los resultados de una ejecución práctica, experimental. Pero al mismo tiempo, nos encontramos con las restricciones de un banco de pruebas limitado donde, por ejemplo, sólo disponemos de una línea de fibra óptica invariable, lo que nos impone la necesidad de simular ciertas condiciones de trabajo como la utilización de canales de comunicación de distinta longitud. En nuestro caso esto se logra de manera muy exacta simplemente variando el valor del atenuador a la salida de Alice.

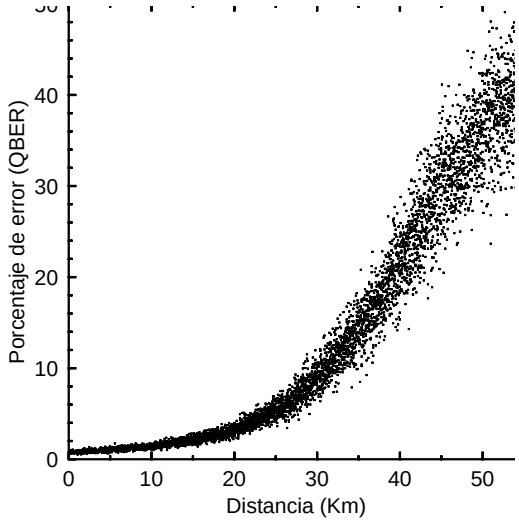


Figura 4.5: Evolución del QBER, *Quantum Bit Error Rate*, frente a la distancia (en Km).

cas del QBER y su comportamiento en función de la distancia, podemos simular la utilización de canales de comunicación de distinta longitud con sólo modificar el valor de la atenuación a la salida de Alice.

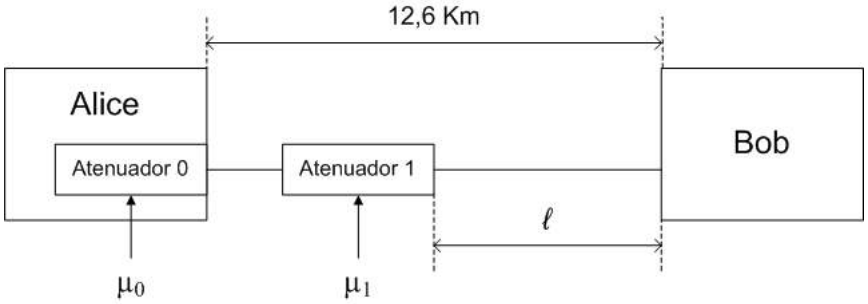


Figura 4.6: Diagrama utilizado para el cálculo de la atenuación en la simulación de canales de comunicación de diferente longitud, donde: μ_0 es el número promedio de fotones a la salida del atenuador 0 (situado dentro de Alice), μ_1 es el número promedio de fotones a la salida del atenuador 1, y ℓ es la longitud de la fibra que va desde el atenuador 1 a Bob.

El diagrama que muestra la figura 4.6 incluye la localización del atenuador intermedio que pretendemos simular con una actuación sobre el atenuador original, el situado en Alice. La idea es aumentar o disminuir la intensidad de los pulsos que salen de Alice, y deducir la longitud equivalente para un μ específico[12]. A partir del diagrama mostrado, podemos representar analíticamente la dependencia entre μ_0 y μ_1 como:

[12]Al hablar de μ hacemos referencia al número promedio de fotones del sistema final que pretendemos simular, es decir, aquel que reúne a los dos atenuadores, 0 y 1, donde Alice y Bob están separados por una fibra de longitud ℓ.

$$\mu_1 = \mu_0 \cdot \tau(12{,}6 - \ell) \Rightarrow \mu_0 = \frac{\mu_1}{\tau(12{,}6 - \ell)} = \frac{\tau(\ell)}{\tau(12{,}6 - \ell)}$$

Donde $\tau(\ell)$ es la transmitancia de una fibra de longitud ℓ, es decir, la pérdida en decibelios de la intensidad de un pulso a través de una fibra óptica de longitud dada.

Partiendo de la expresión anterior podemos calcular el valor de la longitud de la línea que pretendemos simular, en función de la atenuación en Alice, para μ dado. A continuación mostramos los cálculos realizados para los casos de μ óptimo[13] y μ constante.

- Para μ óptimo:

$$\mu_0 = \frac{\mu_1}{\tau(12{,}6 - \ell)} = \frac{10^{\ell \cdot \tau_K + \tau_C/10}}{10^{(12{,}6-\ell)\cdot\tau_K + \tau_C/10}} = 10^{(2\ell - 12{,}6)\cdot\tau_K/10}$$

$$(2\ell - 12{,}6) \cdot \tau_K = 10 \log_{10}(\mu_0)$$

$$\ell = \frac{5 \log_{10}(\mu_0)}{\tau_K} + 6{,}3 \tag{4.8}$$

Donde τ_K es la atenuación en decibelios por kilómetro de fibra óptica (dB/Km), que suele variar entre los 0,25 y 0,3 dB, τ_C es la atenuación en decibelios de un conector de fibra óptica, aproximadamente 0,15 dB, y μ_0 es una función del valor de la atenuación en Alice.

- Para μ constante, $\mu = 0{,}5$:

$$\mu_0 = \frac{\mu_1}{\tau(12{,}6 - \ell)} = \frac{0{,}5}{10^{(12{,}6-\ell)\cdot\tau_K + \tau_C/10}}$$

$$(12{,}6 - \ell) \cdot \tau_K + \tau_C = 10 \log_{10}\left(\frac{\mu_0}{0{,}5}\right) \Rightarrow \ell = \frac{10 \log_{10}(2\mu_0)}{\tau_K} + 12{,}6$$

Donde τ_K, τ_C y μ_0 son los mismos valores que utilizamos en la ecuación anterior (4.8). En la siguiente ecuación (4.9) mostramos la expresión de la longitud en función de un μ constante cualquiera.

$$\ell = \frac{10 \log_{10}(\mu_0/\mu) - 2\tau_C}{\tau_K} + 12{,}6 \tag{4.9}$$

[13] Entendemos por μ óptimo como aquel que maximiza la probabilidad de obtener una detección en Bob, manteniendo siempre el número promedio de pulsos con más de un fotón (a la salida de Alice) inferior al número de pulsos detectados en Bob. En otras palabras, μ óptimo es el valor de μ a partir del cual un atacante podría obtener el 100 % de la clave intercambiada mediante un ataque PNS sin alterar el tamaño final de la clave.

4.2.3. Límite de seguridad

Utilizando el concepto de información mutua que nos proporciona la teoría de la información[14], podemos buscar una cota del error en un sistema QKD que garantice el intercambio de una clave "segura". Si definimos $I(A,B)$ como la información mutua, o información compartida entre Alice y Bob. Y sea $I(A,E)$ e $I(B,E)$ la información compartida por el espía y Alice o Bob respectivamente. Sabemos que, para que Alice y Bob puedan compartir una clave secreta se deben cumplir las siguientes restricciones:

$$I(A,B) \geq \text{máx}\left\{I(A,E), I(B,E)\right\} \tag{4.10}$$

Puesto que ha de cumplirse al mismo tiempo que $I(A,B) \geq I(A,E)$ y $I(A,B) \geq I(B,E)$. Donde, por definición, la información mutua entre Alice y Bob podemos expresarla como:

$$I(A,B) = H(A) - H(A|B) = H(B) - H(B|A)$$

Siendo H la entropía de Shannon, una medida de la incertidumbre.

Podemos expresar la información mutua entre Alice y Bob en función del QBER, ϵ, como:

$$I(A,B) = n\left(1 - H\left(\epsilon\right)\right) = n\left(1 - \epsilon\log_2\left(\epsilon\right) - \left(1-\epsilon\right)\log_2\left(1-\epsilon\right)\right)$$

Donde n es el tamaño de la clave reconciliada.

Pero además, podemos deducir algo intuitivo, y es que la suma de las informaciones mutuas de Bob y el espía con Alice no pueden ser superiores a la información proporcionada por Alice. En otras palabras, Bob y el espía no pueden recibir más información que la que Alice ha enviado. Por lo tanto, para una clave reconciliada de tamaño n, tenemos que:

$$I(A,B) + I(A,E) \leq n$$

Expresión que podemos aplicar a un único qubit, donde $I(A,B) + I(A,E) \leq 1$, que junto a la ecuación inicial (4.10) nos lleva a la interesante conclusión por la cual: para que Alice y Bob compartan una clave secreta, la información mutua entre ambos deber ser mayor o igual a $1/2$.

$$I(A,B) \geq \frac{1}{2}$$

$$I(A,B) = 1 - Q\log_2\left(Q\right) - \left(1-Q\right)\log_2\left(1-Q\right) \geq \frac{1}{2}$$

$$Q\log_2\left(Q\right) + \left(1-Q\right)\log_2\left(1-Q\right) \leq \frac{1}{2} \Rightarrow Q \leq 11\,\%$$

La conclusión a la que llegamos nos proporciona una cota para la tasa de error, QBER, bajo la cual Alice y Bob pueden compartir una clave secreta. Por encima del $11\,\%$ obtenido, la seguridad de la clave no está garantizada y debe ser descartada. Gráficamente podemos mostrar el resultado final obtenido como se refleja en la figura 4.7.

[14] Información mutua de Shannon.

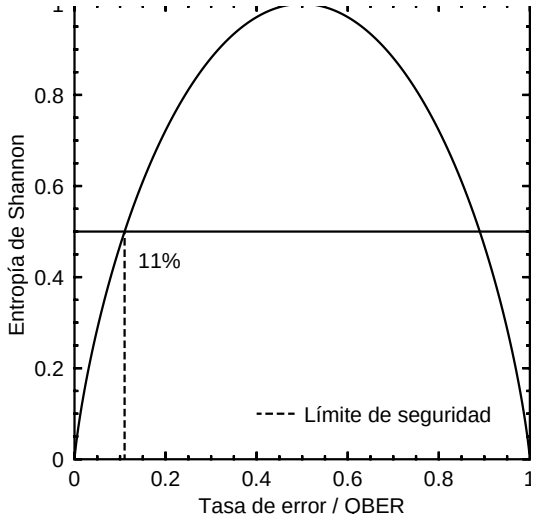


Figura 4.7: Límite del QBER.

Observación: Una vez más hemos dejado aparte la rigurosidad matemática que formaliza los resultados obtenidos, pero que podrá encontrar en algunas de las referencias bibliográficas incluidas [30].

4.3. Destilación de la clave

En un escenario ideal[15] los dos primeros niveles de la arquitectura[16] son suficientes para completar el intercambio de una clave idéntica entre Alice y Bob. Esta es la situación bajo la que se describen los protocolos estudiados en el segundo capítulo de este proyecto, pero esa condición ideal no se cumple en un escenario real, donde aparecen distintos factores que van a provocar distorsiones en los resultados esperados.

- En particular, para la construcción de un sistema "relativamente joven" como el de la distribución cuántica de claves, el principal inconveniente es la **imprecisión** de los componentes utilizados. En nuestro caso, el emisor de pulsos atenuados, los detectores de fotones, o la **sincronización de los sistemas**, son algunas de las fuentes de imprecisión más importantes a tener en cuenta[17].

- A los problemas de una tecnología todavía sin madurar tenemos que añadir también los problemas clásicos de todo sistema de comunicación, como es el **ruido**. El avance tecnológico puede mejorar considerablemente esta imprecisión, pero nunca solucionará por completo el problema, por lo que

[15] En ausencia de errores y espías que modifiquen la información intercambiada.

[16] Intercambio de clave en bruto y reconciliación de bases.

[17] A modo de ejemplo, la probabilidad de obtener algún fotón a la salida del atenuador, utilizando una media de 0.5, no supera el 40 %, mientras que la eficiencia de los detectores que utilizamos actualmente es tan sólo del 10 %. Lo que supone que en condiciones de transmisión ideales, sin pérdidas, el porcentaje de detección máxima será del 4 %.

el ruido es un factor que siempre deberemos tener en cuenta, lo que hace imprescindible el uso de un nivel de corrección de errores y, como justificaremos en este apartado, otro nivel de aplicación de privacidad.

- Finalmente, la actuación de un **espía** puede ser otro factor que provoque la presencia de errores. De forma intencionada o accidental, un atacante puede modificar la información que intentan compartir Alice y Bob. Y en cierto modo, los errores causados por un espía pueden quedar camuflados por la existencia de los errores descritos anteriormente. En consecuencia, la imprecisión de nuestro sistema es la primera de las vulnerabilidades a tener en cuenta a la hora de intentar mejorar la seguridad.

4.3.1. Corrección de errores

Toda comunicación se encuentra expuesta a la presencia de errores. Como acabamos de ver, las imperfecciones, el ruido o un espía pueden ser la causa de esos errores. Pero independientemente de cual sea el origen, el resultado es el mismo, tenemos que corregir cualquier discrepancia entre las claves intercambiadas.

Hasta aquí todo puede parecer normal. Tenemos un sistema de comunicación donde pueden aparecer errores, algo habitual, y queremos detectar y corregir esos errores. Actuando en consecuencia, lo primero que buscamos es una tasa aproximada del porcentaje de errores esperado, que nos permita estudiar la estrategia que mejor se adapte a nuestras necesidades. Y aquí acaba cualquier atisbo de normalidad, al descubrir que la tasa de error de nuestro sistema suele comprender porcentajes superiores al 1 %[18], pudiendo superar incluso tasas del 10 % hasta alcanzar el límite de seguridad del 11 % visto en el apartado anterior. Un dato no más importante que curioso, es que esa elevada tasa de error es el motivo por el cual se utiliza un término específico para su alusión, QBER.

$$QBER = \frac{nerrores}{nerrores + n\neg error} \tag{4.11}$$

Debido a la presencia de errores, debemos buscar un mecanismo que nos permita eliminar las discrepancias entre las claves compartidas por Alice y Bob. Pero esa corrección tiene que realizarse a través de un canal de comunicación público, por lo que en cierto modo se va a proporcionar información a un posible espía[19]. Por esta razón y puesto que la utilización de un canal cuántico es costoso, debemos buscar la manera de minimizar la cantidad de información proporcionada al mismo tiempo que corregimos los errores, con el objetivo de maximizar el tamaño de la clave destilada al final del proceso.

Antes de continuar debemos tener en cuenta los siguientes detalles, que van a influir en el desarrollo de nuestra estrategia para la corrección de errores:

[18] Debemos tener presente que el porcentaje de error esperado en un sistema de comunicación clásico a través de fibra óptica es del orden de 10^{-6}.

[19] Recordemos que el canal de comunicación clásico entre Alice y Bob funciona bajo el supuesto de que dicho canal está autenticado, y por lo tanto, la información transmitida a través de dicho canal no puede ser modificada, pero sí puede ser leída.

- El primero de ellos es que no podemos utilizar ningún mecanismo de redundancia en la clave intercambiada como medio para corregir los posibles errores en la comunicación[20]. Esto es debido a que la redundancia proporciona gran cantidad de información, al mismo tiempo que deshace la aleatoriedad de la clave intercambiada.

- Otro factor a tener en cuenta es que la tasa de error de nuestros sistemas es bastante elevada, lo que nos impide la utilización de la gran mayoría de estrategias utilizadas habitualmente para la corrección de errores.

En consecuencia, nos vemos obligados a la definición e implementación de nuevos mecanismos de corrección de errores, con dos características fundamentales: que corrijan una tasa de error elevada, y que proporcionen una cantidad de información mínima acerca del mensaje (la clave) que intentamos corregir. Como veremos más adelante, a estos dos requisitos se les añade una tercera exigencia: eficiencia; y es que también veremos cómo la corrección de errores va a ser un cuello de botella en el sistema.

Cascade

Con el objetivo de construir un procedimiento de corrección de errores que proporcione una cantidad de información mínima, Gilles Brassard y Louis Salvail definen en 1994 un nuevo protocolo, *Cascade*, basado en el intercambio de paridades por bloque [18]. La cantidad de información proporcionada por este procedimiento se encuentra cerca de los límites teóricos, pero como veremos más adelante, no puede ser implementado de forma eficiente.

Elección de los parámetros de ejecución

Como veremos a continuación, además de definir un nuevo protocolo de corrección de errores, G. Brassard y L. Salvail presentan en ese mismo artículo [18] las herramientas de cálculo necesarias para obtener los parámetros de ejecución óptimos en el *Cascade*. Un parámetro clave es la elección del tamaño de bloque inicial. Una elección adecuada de ese tamaño de bloque inicial provocará que la probabilidad de que el bloque K_v^1 (bloque v en la pasada 1) tenga uno o más errores decrezca de forma exponencial con cada una de las pasadas. Para calcular el tamaño del bloque inicial, ambos autores parten de la definición de $\delta_i(j)$ como la probabilidad de que después del paso i-ésimo, $i \geq 1$, sigan existiendo $2j$ errores en el bloque K_v^1, es decir, la probabilidad de que al menos sigan existiendo j parejas de errores.

$$\delta_1(j) = Pr(X = 2j) + Pr(X = 2j + 1)$$

A continuación, los autores definen E_i como el número de errores en el bloque K_v^1 después de i pasadas. Donde para la primera pasada, $i = 1$, tenemos:

[20] Quizá pueda parecer evidente o innecesaria la exposición de este argumento, pero...

$$E_1 = 2 \sum_{j=1}^{\left\lfloor \frac{k_1}{2} \right\rfloor} j\delta_1(j) = k_1 p - \frac{1-(1-2p)^{k_1}}{2}$$

Ahora bien, la conclusión a la que llegan los autores es la siguiente: si elegimos k_1 de tal forma que satisfaga las dos inecuaciones:

$$\sum_{l=j+1}^{\left\lfloor \frac{k_1}{2} \right\rfloor} \delta_1(l) \leq \frac{1}{4}\delta_1(j) \tag{4.12}$$

$$E_1 \leq -\frac{ln\frac{1}{2}}{2} \tag{4.13}$$

Y para la elección de los tamaños de bloque sucesivos utilizamos el doble de tamaño anterior, $k_i = 2k_{i-1}$. Entonces, podemos limitar la cantidad de información perdida (errores no encontrados) después de n pasos. En otras palabras, la cantidad de errores corregidos depende del número de pasos realizados del algoritmo propuesto, así como del tamaño bloque utilizado para el paso inicial, k_1 (ver figura 4.8).

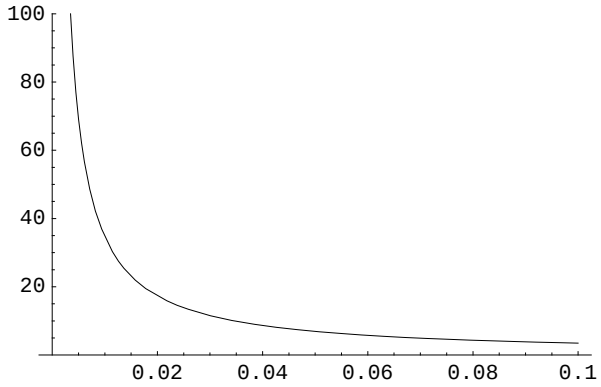


Figura 4.8: Evolución del tamaño máximo recomendado del bloque inicial, k_1, en función de la probabilidad de error para el método de corrección de errores *Cascade*, teniendo en cuenta la definición de E_1 y la segunda inecuación (4.13).

Optimizaciones

La primera propuesta de optimización razonablemente evidente, se basa en procesamiento paralelo de los bloques. Alice y Bob calculan al mismo tiempo las paridades de todos sus bloques e intercambian de forma simultanea ese conjunto de paridades. A continuación, Alice y Bob, comienzan una búsqueda binaria en todos los bloques donde la paridad intercambiada no ha coincidido, permitiendo continuar así con el procesamiento paralelo de los bloques, e intercambiando de forma simultánea todas las paridades involucradas en un mismo nivel.

Esta propuesta y algunas optimizaciones adicionales se encuentran descritas en el siguiente artículo [28], así como un estudio del rendimiento del proceso de corrección de errores.

LDPC[21]. Niágara

Como consecuencia de las limitaciones en cuanto a rendimiento que supone la utilización de un protocolo como el *Cascade*, han aparecido nuevas propuestas, no sólo de optimización, sino orientadas en mecanismos de corrección alternativos, como son los basados en los códigos *Low-Density Parity-Check*, LDPC. En esta familia de protocolos se encuentra el *Niágara*, implementado por BBN Technologies, y referenciado en los artículos del DARPA [48] y [49].

4.3.2. Amplificación de la privacidad

Poco antes de realizar la primera implementación física de un sistema QKD, C. H. Bennett, G. Brassard y J.-M. Robert, adaptan el concepto de amplificación de la privacidad a los sistemas de distribución cuántica de claves [9]. Ese estudio, que data del año 1988, sigue siendo la principal referencia en el incremento de seguridad de un protocolo QKD, complementado con distintas estrategias para el cálculo de la información obtenida por el espía (en términos de entropía), que estudiaremos en los siguientes apartados. Ahora estudiaremos como encaja la amplificación de la privacidad en la pila de protocolos de un sistema QKD, así como su implementación desde el punto de vista práctico.

Originalmente partimos de un protocolo seguro con el objetivo de construir un sistema de distribución cuántica de claves. Desarrollamos un primer nivel del protocolo para el intercambio de una clave en bruto a través de un canal cuántico, y completamos el proceso con un segundo nivel encargado de reconciliar la información de la clave intercambiada utilizando un canal público autenticado. Bajo condiciones de funcionamiento ideales estaríamos ante dos extremos de una comunicación que comparten una clave idéntica, y perfectamente segura; pero aparecen los errores. Las imperfecciones de nuestro sistema y del canal privado que conecta a los extremos de la comunicación, suponen el primer riesgo de seguridad. Un posible espía puede extraer información de nuestro sistema debido a esas imperfecciones. Pero además, necesitamos implementar un tercer nivel para corregir las posibles diferencias en la clave intercambiada. Y esa corrección se vuelve a realizar a través de un canal público, lo que supone una publicación parcial de la clave, o lo que es lo mismo, más información para ese posible espía.

En consecuencia, llegados a este punto tenemos una clave destilada, idéntica en ambos extremos de la comunicación y lista para usar. Pero, puesto que los errores y el ruido de fondo nunca pueden ser eliminados completamente, no se puede garantizar que un posible espía no ha obtenido parte de la información intercambiada. De modo que tenemos una clave de seguridad reducida, lo que nos exige la utilización de un cuarto nivel adicional en nuestra arquitectura,

[21] Low-Density Parity-Check code.

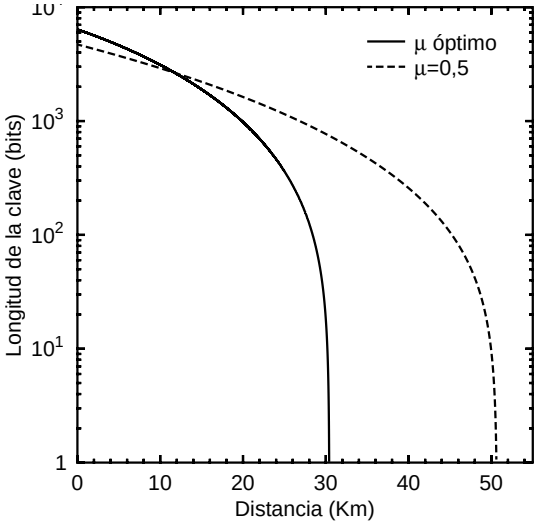


Figura 4.9: Amplificación de la privacidad.

encargado de aumentar "la calidad", o garantizar el grado de seguridad de la clave.

Factores de riesgo

La única forma de ampliar la seguridad de una clave es reduciendo su tamaño. Pero, puesto que el proceso de intercambio de esa clave es costoso, hemos de evaluar el grado de seguridad deseado en función de los factores de riesgo asumidos, con el objetivo de optimizar el tamaño final de la clave.

Algunos de esos factores son:

- El porcentaje de error, QBER, detectado y corregido en el nivel anterior, es el primer factor de riesgo a tener en cuenta. Una parte de ese porcentaje puede haber sido causado por un atacante, pero además, ese porcentaje implica que hemos tenido que transmitir un número determinado de paridades de la clave a través de un canal público. La tasa de error detectada, QBER, y el número de paridades intercambiadas son dos parámetros íntimamente relacionados[22].

- Otro factor de riesgo es la imperfección del sistema. Por ejemplo, en la implementación de un sistema de distribución cuántica de claves con pulsos láser atenuados, sabemos que el número de fotones a la salida de Alice sigue una distribución de Poisson de media μ. Para el protocolo BB84 con una media de 0.5, es decir $\mu = 0,5$, la probabilidad de que a la salida de Alice se envíe más de un fotón es cercana al 8 %, lo que supone un riesgo de seguridad importante.

[22] El número de paridades intercambiadas depende directamente de la tasa de error, es decir, del número de bits erróneos presentes en la clave. Aunque el número total de intercambios puede variar ligeramente en función de la distribución de los errores.

- Finalmente, la última causa de riesgo que descubrimos es la pérdida de aleatoriedad de los datos, o distribución no equiprobable. Cuando los resultados de un sistema QKD no son perfectamente aleatorios, o no están uniformemente distribuidos, un ataque puede usar estrategias condicionadas para obtener una información estadísticamente superior.

Información del espía

El estudio que realizan Bennett et al. analiza la información que un espía puede obtener a través de dos fuentes de información: el ataque al canal privado (cuántico) y la escucha del canal público (intercambio de paridades). Pero de ese análisis separado de los ataques a los canales público y privado llega a una misma conclusión, donde la información obtenida por el espía se procesa de la misma forma, en valores absolutos, independientemente de si la información perdida (o ganada por el atacante) es determinista o no. De esta forma, la información obtenida por el espía a partir de una paridad intercambiada entre los extremos de la comunicación, o por medio de una medida exitosa, es considerada idéntica a efectos de seguridad (y como veremos a continuación, ambas informaciones son eliminadas de la clave final). Este tratamiento análogo de todas las fuentes de información, nos permite aplicar el mismo modelo a cualquier tipo de información obtenida por el atacante.

Implementación. Eliminando la información del espía

Partimos del supuesto por el que Alice y Bob ya comparten una clave de n bits, que podemos interpretar como una variable aleatoria W. Al mismo tiempo sabemos que un espía, Eve, puede conocer hasta t bits de información (determinista o no) de esa clave, $t < n$, que interpretamos con otra variable aleatoria V. En este momento, podemos afirmar que la incertidumbre que el espía tiene de la clave, a partir de la información recopilada por el mismo, es como mínimo el número de bits desconocidos para el espía: $H(W|V) \geq n - t$.

El objetivo a partir de lo visto es reducir la información del espía, en la medida de lo posible, sobre la clave final compartida. Para ello, lo que hacemos es utilizar una función de compresión $g : \{0,1\}^n \rightarrow \{0,1\}^r$, con la que generar una nueva clave, $K = g(W)$, de tamaño inferior al original, $r < n$, pero que proporcione la mínima cantidad de información posible a un hipotético atacante. Como se demuestra en [9] y [19], la función de compresión más adecuada para el objetivo deseado podemos obtenerla de la familia de funciones hash universales[23].

El tamaño de la clave final comprimida, r, dependerá de la cantidad de información conocida por el espía. Continuando bajo el supuesto inicial de que un espía conoce t bis de información de la clave original, podemos calcular el tamaño de la clave final como: $r = n - t - s$, donde s es un **parámetro de seguridad** elegido de tal forma que $s < n - t$, es decir, elegido de manera que siempre nos queden bits de clave disponibles, una vez descontada la información obtenida

[23]Utilizaremos los campos de Galois, $GF_2(2^n)$,para obtener una clase de funciones universales con las que comprimir la clave original.

por el espía y el parámetro de seguridad. La conclusión a la que llega Bennett, Brassard y Robert en [9] acerca de la incertidumbre de la clave final es:

$$H(K|G,V) \geq r - \frac{2^{-s}}{\ln 2}$$

Donde ahora G es una variable aleatoria que representa la elección de una función de compresión tal que $K = G(W)$.

De la expresión anterior podemos realizar varias interpretaciones, donde seguramente la más interesante sea la relativa a la información conocída por el espía (bits de la clave), cuyo valor máximo queda definido por $2^{-s}/\ln 2$. Luego en función de la elección de s, y el conocimiento o estimación de la información comprometida, t, podemos reducir de forma exponencial y tanto como queramos la información del espía sobre la clave final compartida.

El algoritmo 8 muestra los pasos que han de seguir los extremos de la comunicación para amplificar la privacidad de una clave intercambiada mediante QKD.

Observación: Todo el procedimiento de amplificación de la privacidad descrito en este apartado, se aplica a los protocolos basados en el intercambio de qubits no correlacionadas, BB84, B92 y SARG04. Para los protocolos basados en el entrelazamiento de fotones existen mecanismos de amplificación de la privacidad de aplicación directa sobre el nivel cuántico.

Hash universal

Una función hash, o función resumen, asigna a cada elemento de un conjunto de entrada un único elemento de un conjunto de salida mucho mas pequeño. Típicamente, en informática, tenemos que asignar valores de m valores del subconjunto n, donde $m \gg n$. Esto hace que ese valor de salida o resumen, por ejemplo, se pueda utilizar como representante de la entrada para su almacenamiento en una tabla o similar, algo muy utilizado en estructuras de datos.

En criptografía el uso de estas funciones es más específico, ya que se busca que sean funciones de un solo sentido. Es decir, que a partir de una entrada sea fácil encontrar la salida pero que se dificulte lo más posible el camino inverso. Esto nos permite hacer firmado digital de textos reduciendo el consumo computacional, ya que se puede realizar directamente sobre el valor resumen.

Ejemplos de funciones hash utilizadas en informática son los algoritmos SHA o MD5.

La seguridad de estas funciones reside en lo que se conoce como colisiones. Una colisión se produce cuando dos valores distintos de entrada generan un mismo valor de salida. Evidentemente, si el conjunto de entrada es mucho mayor que el de salida esto debe producirse en algún momento. Evitar y minimizar estas colisiones (que equivale a distribuirlas uniformemente en todo el conjunto de salida) es una característica muy apreciada en las funciones hash. Estas son las que se conocen como funciones **hash universales** de un solo sentido (*UOWHF* en sus siglas inglesas: *Universal One-Way Hash Function*).

Un ejemplo típico y práctico de una función hash universal son aquellas que se realizan utilizando campos de *Galois* de orden 2^n, $GF(2^n)$, y son las que

se suelen utilizar en amplificación de la privacidad (DARPA utiliza este tipo de funciones, por ejemplo [48]).

4.4. Estimación de la información de un espía

Existen múltiples trabajos que nos orientan en el cálculo de la información que ha podido extraer un posible espía de la clave final intercambiada, pero en nuestro estudio nos centraremos en tan sólo dos de ellos, los trabajos realizados por Bennett et al. y Slutsky et al. El primero y más antiguo de ellos corresponde a Bennett et al., y se basa en un estudio de la información máxima disponible por un espía a partir de la tasa de error, QBER, y la probabilidad de acierto de cada medida realizada. El segundo de estos trabajos, el de Slutsky et al., aplica los conceptos de función y frontera de defensa al estudio de la seguridad de los sistemas de distribución cuántica de claves.

4.4.1. Entropía de Bennett et al.

En 1992, C. H. Bennett, F. Bessette, G. Brassard, L. Salvail y J. Smolin, presentan la primera estimación práctica de la información que puede obtener un espía en función de la estrategia de ataque utilizada [11], centrándose en dos opciones: intercepción y reenvío, y división del haz (*beamsplitting*).

Intercepta y reenvía

Para la estrategia de intercepción y reenvío, la estimación de la información que puede obtener un espía se centra en la tasa de error, QBER, a través de la cual los interlocutores de una comunicación, Alice y Bob, pueden valorar la actuación realizada por un hipotético atacante. En primer lugar, lo que realizamos es una estimación de la cantidad de información que puede estar comprometida sabiendo que, la tasa error introducida por este ataque es del 25 %. Luego, si e es el número de errores detectados, una cota máxima de la información afectada podría ser:

$$4e + 5\sqrt{12e}$$

Donde $5\sqrt{12e}$ es un factor de 5 sobre la desviación estándar del error.

Ahora bien, el conocimiento que el espía obtiene de la clave se puede maximizar utilizando la base de *Breidbart* [14] [30] (descrita en el capítulo de ataques), donde la información obtenida es siempre inferior a $1/\sqrt{2}$[24]. Esto nos aporta una nueva cota, ahora de la información estadísticamente disponible por el atacante:

$$\frac{4e}{\sqrt{2}} + 5\sqrt{\left(4 + 2\sqrt{2}\right)e}$$

[24]La información obtenida a través de una medida realizada con la base de Breidbart no es determinista. A pesar de esto, utilizamos la cota máxima de la información estadísticamente acertada por esta medida como límite se seguridad.

División del haz

La otra estrategia de ataque que estudian los autores de esta entropía es la división del número de fotones, también conocido como PNS[25], o *beamsplitting*; ataque fundamentado en una deficiencia funcional de las fuentes de emisión de fotones, comentada en el capítulo anterior y que volveremos a revisar en los capítulos finales de este proyecto. Existen múltiples versiones de este ataque, algunas de ellas devastadoras en cuanto al compromiso de la seguridad del sistema, pero sólo nos interesa una de estas versiones, aquella en la que el atacante recoge la información adicional proporcionada por los pulsos que contienen más de un fotón. En tal caso, podemos obtener una estimación de la información obtenida por el espía como:

$$N\mu + 5\sqrt{N\mu(1-\mu)}$$

Donde N es el número de bits reconciliados, o tamaño de la clave reconciliada, y μ es el número promedio de fotones a la salida de Alice. Si analizamos con detalle la expresión, reconoceremos la primera parte de la misma como el número de pulsos con más de un fotón[26] y la segunda, una vez más, la desviación estándar de la información comprometida.

Estimación final de la información obtenida por el espía

Finalmente, si unimos los resultados obtenidos para ambas estrategias de ataque, llegamos a la siguiente expresión que nos da una cota superior del número de bits de la clave, ℓ, conocidos por un hipotético espía:

$$\ell = N\rho + 5\sqrt{N\left(\mu(1-\mu) + \left(4+2\sqrt{2}\right)Q\right)} \tag{4.14}$$

Donde Q es la tasa de error, o QBER, y $\rho = \mu + 4Q/\sqrt{2}$.

Observación: El cálculo de las desviaciones estándar incluidas en cada expresión aparece documentado en el apéndice del artículo original de los autores [11].

4.4.2. Entropía de Slutsky et al.

Con el objetivo de definir, desde el punto de vista de la teoría de la información clásica, la cantidad de información que puede obtener un espía sobre cada bit una clave intercambiada mediante QKD, Slutsky et al. [25] comienzan definiendo los conceptos de información compartida en función de la tasa de errores, QBER. Para ello, definen un primer conjunto C de bits para los cuales el resultado es concluyente, es decir, el conjunto de bits reconciliados por Alice y Bob; y un

[25] *Photon Number Splitting*: división por número de fotones.

[26] En condiciones ideales (por ejemplo, justo a la salida de Alice) el número de bits de clave en bruto coincide con el número promedio de fotones esperado, μ. Por lo que para un protocolo como el BB84, tras la reconciliación de bases el número de bits de clave obtenidos será, $N = \mu/2$. Luego, $N\mu = \mu^2/2$, que corresponde con la aproximación del número de pulsos esperados con más de un fotón.

segundo conjunto E de tal forma que para todo elemento de C, $i \in C$, tenemos $E_i = 0$ si Bob recibe el bit correcto enviado desde Alice, y $E_i = 1$ si Bob recibe un error. Este supuesto nos permite obtener una expresión del número total de errores como:

$$E_T = \sum_{i \in C} E_i$$

Con esto, ya podemos definir la información del espía como mostramos a continuación.

Información del espía

Para cada bit, podemos definir la información obtenida por el espía en función de las entropías de *Rényi* y *Shannon* como:

$$\begin{aligned} I_i^R &= 1 + \log_2 \left[P_i^2 + (1 - P_i)^2\right] \\ I_i^H &= 1 + P_i \log_2 P_i + (1 - P_i) \log_2 (1 - P_i) \end{aligned} \tag{4.15}$$

Donde P_i es la probabilidad, desde el punto de vista del espía, de obtener una medida correcta, en aquellos bits donde no se produjo un error y que, por lo tanto, no fueron descartados de la clave. Además, puesto que el valor de la información del espía sobre cada bit es independiente del resto, podemos calcular la información total obtenida por un ataque al canal cuántico como la suma de las informaciones individuales:

$$\begin{aligned} I_T^R &= \sum_{i \in C} (1 - E_i) I_i^R \\ I_T^H &= \sum_{i \in C} (1 - E_i) I_i^H \end{aligned} \tag{4.16}$$

Conocido esto, el objetivo de Alice y Bob es encontrar una **función de defensa**, $t(n, e_T)$, que sirva como límite superior de la información del espía, I_T^R. Desafortunadamente, I_T^R es una variable aleatoria cuyo único límite superior de forma determinista es la longitud completa de la clave, $n - e_T$; estimación inaceptable para Alice y Bob, puesto que le exigiría descartar toda la transmisión. En consecuencia, buscamos un límite estadístico tal que sólo ocurra con una pequeña probabilidad (aunque no cero) la posibilidad de la información del espía se encuentre por encima de nuestra función de defensa: $I_T^R > t(n, e_T)$.

Debemos tener en cuenta también que este límite debe combinarse con los límites del resto de fuentes de pérdida de información, puesto que sólo estamos estudiando la seguridad de la clave intercambiada a partir de la tasa de error obtenida, QBER.

Ataque exitoso

Decimos que un ataque es exitoso cuando introduce un número de errores e_T en la clave reconciliada, de n bits, y la información obtenida por el espía es

mayor que la función de defensa utilizada: $I_T^R > t(n, e_T)$. Siguiendo esta definición el objetivo de Alice y Bob es construir una función de defensa, $t(n, e_T)$, que minimice el logro de un ataque exitoso.

Una vez que Slutsky et al. definen el concepto de ataque exitoso, proceden a su análisis en función de la probabilidad de acierto. Para lo cual, llamamos S al evento que representa un ataque exitoso, y estudiamos su minimización desde tres interpretaciones matemáticas:

1. En términos de probabilidad a priori, $\Pr(S)$.
2. En términos de probabilidad condicional, a posteriori, $\Pr(S|N = n, E_T = e_T)$, sobre una distribución cuántica de claves que resulta en n bits reconciliados con e_T errores.
3. En términos de probabilidad intermedia, $\Pr(S|N = n)$, sobre una distribución cuántica de claves que resulta en n bits reconciliados.

Como comenta Slutsky et al., en los casos 1 y 3 podemos construir una función de defensa tal que la probabilidad de que la información del espía sea superior, no exceda de un valor α pequeño, $\alpha > 0$. Sin embargo, en el segundo de los casos, es decir, con la probabilidad a posteriori, no podemos obtener una función de defensa, puesto que la probabilidad no puede ser reducida tanto como queramos.

Función de defensa

Los criterios de los que parte Slutsky et al. para la búsqueda de una función de defensa son los descritos a continuación. En primer lugar, la función de defensa de Alice y Bob dada por $t = t_1(n, e_T)$, debe ser monótonamente creciente en función de e_T para un n fijo. Definiendo el suceso S correspondiente a un ataque exitoso según la definición anterior, así como las siguientes expresiones para los valores ponderados del error y la información del espía:

$$\overline{E} \hat{=} E[E_i | i \in C] = \sum_{i \in C} P_i E_i$$

$$\overline{I}^R \hat{=} E[I_i^R | i \in C, E_i = 0] = \sum_{i \in C} P_i I_i^R$$

Entonces, Slutsky et al. demuestran que: para cualquier x, $x > 0$, tal que $x < n\overline{E}$ y $t_1(n, x)/(n - x) > I_i^R$,

$$\begin{aligned} \Pr(S|N = n) \leq & \frac{1}{2}\left(1 - erf\left[\sqrt{2n}\left(\overline{E} - \frac{x}{n}\right)\right]\right) \\ & + \frac{1}{2}\left(1 - erf\left[\sqrt{2(n-x)}\left(\frac{t_1(n,x)}{n-x} - \overline{I^R}\right)\right]\right) \end{aligned}$$

Donde la función de error estándar es:

$$erf(z) = (2/\sqrt{n}) \int_0^z \exp(-\zeta^2) d\zeta$$

Este resultado nos permite definir una primera expresión de la función de defensa, según la cual: para cualquier estrategia de ataque dada, Alice y Bob pueden reducir el éxito del espía en el sentido de probabilidad "intermedia", $\Pr(S|N = n)$, a un nivel α tan bajo como deseemos utilizando la función de defensa:

$$t_1(n, e_T) = n \begin{cases} 0 & e_T/n < \overline{E} - \xi \\ (1 - (\overline{E} - \xi))\overline{I}^R + \xi\sqrt{1 - (\overline{E} - \xi)} & e_T/n \geq \overline{E} - \xi \end{cases} \tag{4.17}$$

Donde: $\xi = (1/\sqrt{2n})erf^{-1}(1 - \alpha)$, y los valores de $\overline{E}$ e $\overline{I}^R$ son los descritos anteriormente.

Esta misma función de defensa también nos sirve para limitar la probabilidad "a priori", $\Pr(S)$, a partir de la desigualdad:

$$\Pr(S) \leq \max_n \{\Pr(S|N = n)\} < \alpha$$

Lo que nos da la primera herramienta para definir un límite de seguridad en nuestro intercambio a partir de los errores detectados en el mismo.

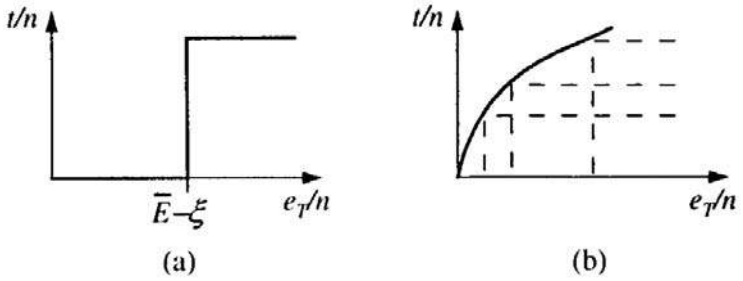


Figura 4.10: Función y frontera de defensa.

Frontera de defensa

En la sección anterior asumimos que el espía tenía a su disposición tan sólo una estrategia de ataque, caracterizada por las medias condicionales $\overline{E}$ e $\overline{I}^R$. Eliminemos esa limitación. Entonces, Alice y Bob deben calcular las cantidades de $\overline{E}$ e $\overline{I}^R$ para dibujar la función de defensa de cada estrategia posible, de tal forma que podemos construir una **frontera de defensa**, $t_F(n, e_T)$, a partir de todas las funciones de defensa individuales que describimos con: $t_1(n, e_T)$.

Es intuitivo comprobar que a partir de la ecuación de la función de defensa, $t_1(n, e_T)$, dos estrategias distintas con idénticos valores de $\overline{E}$ se ordenan según el valor de $\overline{I}^R$, es decir, el valor de la función de defensa será mayor para aquella estrategia que tenga el mayor valor de $\overline{I}^R$. Luego la frontera de defensa se caracterizará por los valores de $\overline{I}^R$ máximos para cada valor de $\overline{E}$, que denotaremos por $\overline{I}^R_{\text{máx}}(\overline{E})$.

A partir de la ecuación de la función de defensa también podemos ver que las esquinas de la función de defensa vienen dadas por las ecuaciones:

$$\frac{e_T}{n} = \overline{E} - \zeta$$

De las que deducimos:

$$t = (n - e_T)\overline{I^R}_{\text{máx}}\left(\frac{e_T}{n} + \xi\right) + \xi\sqrt{n(n - e_T)}$$

Expresión con la que podemos calcular la frontera de defensa que nos permitirá asegurar los intercambios de claves de nuestro sistema QKD:

$$t_F(n, e_T) = \max_{e \leq e_T}\left\{(n - e)\overline{I^R}_{\text{máx}}\left(\frac{e_T}{n} + \xi\right) + \xi\sqrt{n(n - e)}\right\} \tag{4.18}$$

Necesitamos la expresión que nos permita calcular la información máxima obtenida por el espía sobre los datos corregidos. Esta expresión ha sido calculada por Slutsky et al. en otro artículo [26] para distintos protocolos, entre los que se encuentra el BB84, donde el resultado alcanzado por Slutsky et al. es el siguiente:

$$\overline{I^R}_{\text{máx}}(\overline{E}) = 1 + \log_2\left[1 - \frac{1}{2}\left(\frac{1 - 3\overline{E}}{1 - \overline{E}}\right)^2\right] \tag{4.19}$$

4.4.3. Otras estimaciones

Otros cálculos para la información que puede obtener un posible espía son los propuestos por las entropías de Myers-Pearson y Shor-Preskill. Las expresiones de estas entropías han sido extraídas de [48], y algunos de los parámetros de los que dependen son los descritos a continuación:

- Tamaño de la clave reconciliada, b.
- Número de errores en la clave reconciliada, e.
- Número total de bits transmitidos, n.
- Número de bits de paridad divulgados durante la corrección de errores, d.
- Medida de no-aleatoriedad obtenidas de las pruebas de aleatoriedad, r.
- Parámetro de confianza (riesgo), c.

Para todas las entropías, el cálculo final de la clave se realizará como:

$$t - r - d - m_1 n - m_2 b \tag{4.20}$$

Entropía de Myers-Pearson

$$t = \max_{R \in (1,2)}\left[(b - e)\frac{b - e}{1 - R}\log_2\left(p_E^R + (1 - p_E)^R\right) - \log_2\left(\frac{R}{c(R - 1)}\right) - 2\right] \tag{4.21}$$

Donde $p_E = \frac{1}{2} + \sqrt{\frac{p}{1-p}\left(1 - \frac{p}{1-p}\right)}$, $\sum_{i=0}^{e}\binom{b}{i} p^i(1 - p)^{n-i} = c$.

Entropía de Shor-Preskill

$$t = (b - e)(1 + p\log_2(p) + (1 - p)\log_2(1 - p)) + 2\log_2(c) \tag{4.22}$$

4.5. Autenticación

El último nivel que añadimos es el de autenticación.

Retrocedemos hasta el comienzo de este capítulo simplemente por el hecho de recordar la insistente defensa que hicimos del primer nivel de nuestra arquitectura. Y la razón de subrayar esa justificación es porque nos vamos a encontrar con un problema similar en este nivel. Podemos discrepar acerca de que la autenticación sea o no una parte congénita de los protocolos de distribución de claves, pero no existe debate alguno cuando afirmamos que la autenticación es una herramienta imprescindible, y que debemos tener presente en todo momento que el canal público utilizado debe estar autenticado.

4.6. Conclusiones

Con el diseño de nuestro sistema finalizado, hemos construido un prototipo con el que obtener los primeros resultados. Es el momento de mostrar esos resultados.

4.6.1. Evolución de la clave

Evolución de la clave reconciliada

El factor principal que influye sobre el tamaño de la clave intercambiada es la distancia. A mayor distancia, mayores son las pérdidas causadas por el transporte de la información, y en consecuencia, menor es el número final de detecciones obtenidas. Esto provoca que la clave en bruto, de la que partimos para reconstruir la clave final, se reduzca de forma exponencial en función de la distancia. Por extensión, la clave reconciliada sufre la misma reducción exponencial frente a la distancia.

La gráfica 4.11 muestra dos ejemplos de la evolución del tamaño de clave intercambiada. Como indica la leyenda, la evolución mostrada con una línea continua ha sido obtenida ajustando el valor del número promedio de fotones, μ, a la salida de Alice en función de la distancia, para ajustarlo al valor óptimo definido como aquél para el que un ataque PNS completo deja de tener un éxito del 100 %. Este valor de μ varía con la distancia de la transmisión. La otra curva ha sido calculada manteniendo constante el número promedio de fotones a la salida de Alice.

Evolución de la clave destilada

Pero la distancia no es la única razón por la que se reduce el tamaño de la clave compartida. La pila de protocolos que acabamos de describir influye

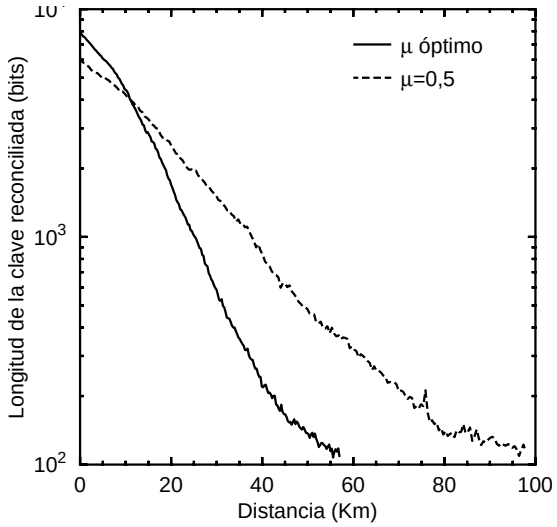


Figura 4.11: Evolución del tamaño de la clave reconciliada en función de la distancia.

directamente en la clave como muestra la siguiente figura, 4.12.

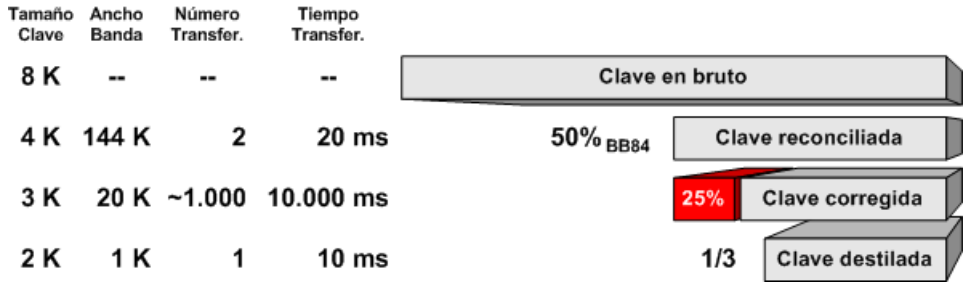


Figura 4.12: Evolución del tamaño de la clave intercambiada a través de los distintos niveles de un sistema QKD. La pérdida de un 25 % de la clave reconciliada durante el proceso de corrección de errores se asume bajo la hipótesis de que la tasa de errores, QBER, es del 3 %.

El proceso de reconciliación de bases utilizado en el protocolo BB84 elimina un 50 % de la clave, descartando aquellos valores donde la base utilizada para la decodificación no ha sido la correcta. A continuación, en la corrección de errores necesitamos intercambiar de forma aproximada un 25 % de clave en paridades, lo que puede ser eliminado de forma directa o parcial en el siguiente nivel de amplificación de la privacidad. Finalmente, la clave final compartida se reducirán en función de las estimaciones de seguridad consideradas. Pero no acaba aquí la reducción de la clave. Si hacemos memoria, podremos recordar uno de los supuestos de los que partimos para la implementación de nuestro sistema, y es que el canal público se encuentre autenticado. Pues bien, esa autenticación debe realizarse de forma automática por el sistema y para ello utilizará parte de la clave final destilada. En consecuencia, el resultado será una reducción adicional de la clave final.

Algoritmo 6 Corrección de errores. Cascade

1. Alice y Bob acuerdan el tamaño de bloque inicial, k_1, calculado a partir de una tasa de error esperada utilizando la ecuación 4.13. Esa tasa de error puede ser estimada intercambiando una pequeña muestra de la clave.

2. Dividimos la clave en bloques de tamaño máximo definido, k_i, en función del número de iteración, i, que estamos realizando.

3. Para cada uno de los bloques definidos en el paso anterior:

   a) Calculamos la paridad del bloque.

   b) Intercambiamos la paridades calculadas entre Alice y Bob.

      1) Si la paridad es idéntica, saltamos al siguiente bloque (no hemos detectado ningún error).
      2) Si la paridad es diferente, realizamos una búsqueda binaria del error en el bloque actual (ver el algoritmo 7).
      Encontrar un error en el paso n-ésimo, implica que en los pasos anteriores hemos dejado errores sin corregir. Puesto que las paridades eran idénticas, el error encontrado nos lleva a deducir que existe otro error en los bloques estudiados con anterioridad (un número par de errores por bloque producen una paridad idéntica a la del bloque sin errores). Por lo tanto, debemos volver a los bloques de los pasos anteriores donde aparezca el bit corregido para realizar una nueva búsqueda.
      La vuelta atrás se realiza en dos pasos. En primer lugar se regresa a los tamaños de bloque inferiores para corregir aquellos bloques donde se detectó el nuevo error. Y una vez corregidos los errores en bloques de tamaño inferior, se debe avanzar de nuevo a los bloques de mayor tamaño ya estudiados donde puede haberse camuflado un nuevo par de errores.
      De igual forma, la corrección de un nuevo error por la vuelta atrás implica la reiteración de una nueva vuelta atrás para corregir los errores acumulados (y asi sucesivamente, de forma anidada).

4. Una vez que hemos acabado de procesar las paridades de todos los bloques, calculamos el nuevo tamaño de bloque como el doble del tamaño de bloque anterior: $k_{i+1} = 2k_i$.

5. Si el nuevo tamaño de bloque definido es superior al tamaño de la clave, hemos acabado el proceso de corrección de errores. De igual forma, si se han completado al menos cuatro iteraciones del proceso de corrección nuestra clave debe ser idéntica en ambos extremos de la comunicación, y por lo tanto también habríamos acabado.

6. En otro caso, reordenamos la clave de forma aleatoria y similar en Alice y Bob.

7. Realizamos una nueva iteración desde el paso 2.

Algoritmo 7 Corrección de errores. Búsqueda binaria

1. Si el bloque es de tamaño 1, hemos encontrado el error. Invertimos el bit del bloque en Bob y finalizamos el procedimiento de búsqueda.
2. En otro caso, dividimos el bloque actual en dos nuevos subbloques de igual tamaño.
3. Calculamos la paridad del primer subbloque.
4. Intercambiamos la paridad del primer subbloque entre Alice y Bob.
   a) Si la paridad compartida es idéntica, el error está en el segundo subbloque. Saltamos al paso 1 con el segundo subbloque.
   b) Si la paridad es diferente, el error está en este primer subbloque. Saltamos al paso 1 con el primer bloque.

Algoritmo 8 Amplificación de la privacidad.

1. Partimos del supuesto de que Alice y Bob comparten una clave idéntica, K_{Sifted}, de tamaño n.
2. Uno de los interlocutores, Alice o Bob, estima la cantidad de información disponible por un hipotético espía, t, en función de los parámetros de riesgo, como pueden ser el QBER y el número promedio de fotones a la salida de Alice, μ.
3. Alice o Bob selecciona un parámetro de seguridad s, calcula el tamaño de la clave final, $r = n - t - s$, y elige una función de compresión dentro de la familia de funciones hash universales, $GF(2^n)$:

   $$g : \{0,1\}^n \rightarrow \{0,1\}^r$$

   Anuncia al otro interlocutor, a través del canal público, la descripción de la función seleccionada.
4. Alice y Bob calculan la clave final, K_{Secret}, utilizando la función intercambiada:

   $$K_{Secret} = g(K_{Sifted})$$

Capítulo 5

Implementación del software

5.1. Entorno de desarrollo

La elección del entorno de desarrollo nos ha sido impuesta por los equipos utilizados para el nivel nivel físico, el sistema id-3000 de id-Quantique. Estos equipos son sistemas de desarrollo que, como se ha visto, implementan una capa física muy básica. Se hacen por encargo y son básicamente más equipos de laboratorio que sistemas de comunicaciones. El sistema final viene preparado para trabajar con la siguiente configuración:

- Los dispositivos, QKDS-A y QKDS-B, se conectan a un ordenador utilizando un puerto USB estándar.
- El sistema incorpora un controlador compatible con el sistema de Microsoft Windows XP.
- Para la implementación de una solución a partir de los equipos utilizados, el sistema proporciona un conjunto de bibliotecas de desarrollo, DLL's, y dos interfaces de acceso al hardware. El lenguaje en el que han sido programadas estas interfaces es C++, y el entorno de desarrollo recomendado por el fabricante es Visual Studio, de Microsoft, en su versión de 2003[1].

Estrategias de futuro

A pesar de las imposiciones en cuanto a lenguaje y entorno de desarrollo, consideramos la elección de C++ como decisión acertada, por su disponibilidad en la gran mayoría de las plataformas utilizadas, su potencia, y la existencia de un estándar para la utilización de este lenguaje. Por esta razón, hemos intentado cuidar una implementación compatible con distintas plataformas, pensando en un futuro donde no tengamos esa exigencia del nivel físico, y necesitemos portar todo el código desarrollado a otro sistema. Hay que tener en cuenta que el objetivo es su instalación junto con un equipamiento normal de comunicaciones, por lo que muy posiblemente acabará funcionando en un sistema empotrado. En el intento por cumplir esa línea de buenas intenciones, nos vemos obligados

[1] Existe una versión más reciente de este entorno de desarrollo, disponible a partir de 2005, que hemos descartado a fin de evitar cualquier posible problema de compatibilidad entre versiones.

a seguir tres estándares: ANSI[2], ISO C++ y POSIX[3] (donde el orden exposición no sugiere un criterio distinto al alfabético). Y muy a pesar de todas las precauciones tomadas, no hemos tenido más remedio que utilizar algunas funciones específicas del sistema, como son las llamadas para la creación de hilos (*Threads*) y los mecanismos de sincronización, especificados en la tabla 5.1.

Hilos de ejecución	Sincronización	Tipos
`HANDLE`	`CRITICAL_SECTION`	`BYTE`
`CreateThread()`	`InitializeCriticalSection()`	`WORD`
`TerminateThread()`	`DeleteCriticalSection()`	`DWORD`
`CloseHandle()`	`EnterCriticalSection()`	`VOID`
`CloseHandle()`	`LeaveCriticalSection()`	`...`

Tabla 5.1: Lista de tipos y funciones dependientes de la plataforma de desarrollo utilizada (Microsoft Visual Studio 2003).

5.1.1. Control de versiones

El desarrollo del proyecto se mantiene a través de un sistema de control de versiones, que gestiona una copia de seguridad del código, así como el acceso simultáneo de varios usuarios, permitiendo un desarrollo paralelo de los distintos módulos del sistema. El software utilizado para este sistema de control de versiones es *Subversion*, SVN, y los datos necesarios para acceder al repositorio con el código, son:

- Dirección del servidor: `bender.ls.fi.upm.es` (138.100.12.203)
- Directorio raíz: `/var/svn`
- Nombre del repositorio: `qkds`

El acceso al servidor SVN debe realizarse a través de una conexión cifrada, a través de un túnel SSH, por lo que la cadena de acceso al repositorio del proyecto será de la forma:

`svn+ssh://`*`usuario`*`@bender.ls.fi.upm.es/var/svn/qkds`

5.2. Diseño del proyecto

5.2.1. Estructura

Según lo comentado en el apartado inicial, no hemos justificado la selección de un lenguaje de desarrollo, ya que éste viene impuesto por la interfaz utilizada para el acceso al hardware. A pesar de esto, la estructuración en clases facilitada por el lenguaje C++ nos va a permitir diseñar, de una forma modular, cada uno de los niveles de la arquitectura estudiada en los capítulos anteriores.

[2]American National Standards Institute.
[3]Portable Operating System Interface for uniX.

Arquitectura que podremos reflejar en una jerarquía de clases como la mostrada en la figura 5.1, donde encontramos hasta un total de 6 niveles con la siguiente interpretación:

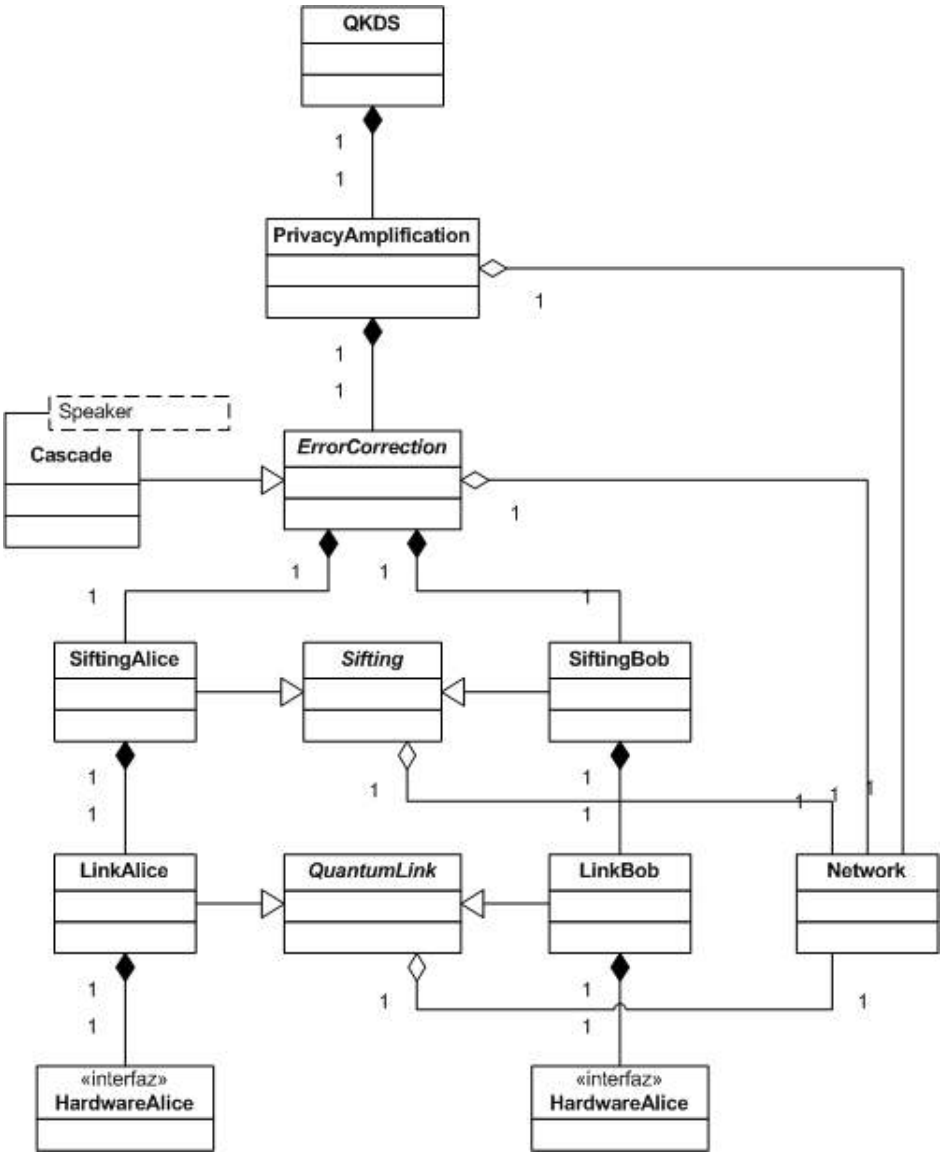


Figura 5.1: Jerarquía de clases del sistema QKD implementado.

1. En la parte inferior de la figura encontramos las dos interfaces de acceso al hardware. Estas interfaces son distintas en función del equipo gestionado, Alice o Bob, y no corresponden con ningún nivel equivalente en la arquitectura estudiada para un sistema QKD. Su vinculación con el hardware es directa, por lo que podemos interpretar esta sección de la jerarquía como el nivel físico de nuestro sistema.

2. En el siguiente escalón de la jerarquía de clases comienza la equivalencia directa con los niveles de la arquitectura de nuestro sistema QKD. Es el

primer nivel, para el intercambio de una clave en bruto, al que denominamos nivel de enlace cuántico (*Quantum Link*). Como podemos observar en el diagrama mostrado, la clase que implementa este nivel es una clase abstracta, que sólo puede ser instanciada utilizando una de sus dos clases hijas, para la ejecución del protocolo desde cada uno de los extremos de la comunicación: Alice o Bob. Esta distinción entre las clases de Alice y Bob en el acceso a la capa de enlace es debida a que los procesos a ejecutar son distintos para cada uno de los extremos.

3. La doble interpretación de una clase abstracta que acabamos de ver, también se extiende al siguiente nivel de reconciliación de bases (*Sifting*), donde las tareas que ejecuta cada uno de los extremos de la comunicación siguen presentando ligeras diferencias, que sugieren la utilización de clases independientes.

4. El nivel de corrección de errores vuelve a estar representado por una clase abstracta, pero esta vez el objetivo es distinto. Como podemos ver en la jerarquía de clases, sólo existe un hijo para la corrección de errores. Ese hijo es la clase *Cascade* que implementa el procedimiento de corrección de errores anteriormente ya descrito. La razón por la que esta clase no requiere una instanciación distinta para Alice o Bob se debe al hecho de que su ejecución es prácticamente simétrica (idéntica en ambos extremos de la comunicación). El uso de una clase abstracta a este nivel viene dado sólamente por dejar la posibilidad de ampliar y mejorar este nivel, que es crítico para el rendimiento del sistema. En un futuro cercano deseamos incorporar otros mecanismos para la corrección de errores, incluso es probable que la utilización de un procedimiento u otro dependa de la tasa de errores a corregir (o lo que es lo mismo, de la distancia que separa a los extremos de la comunicación). Luego es una sugerencia interesante el partir de una clase abstracta pensando en el porvenir de los sistemas QKD.

5. La amplificación de la privacidad es un caso muy similar al del nivel de corrección de errores. Volvemos a encontrarnos con distintos mecanismos para ampliar la seguridad de la clave intercambiada, todos ellos compatibles con este nivel de amplificación de la privacidad. Pero hay algo más importante que debemos comentar en este momento, y es que la ejecución de este nivel depende de todos los cálculos de la entropía de la clave intercambiada, obtenidos desde cada uno de los niveles anteriores. Para realizar ese cálculo utilizamos una clase específica, enlazada a cada uno de los niveles de la arquitectura (similar a la clase *Network* mostrada en el diagrama). Clase que hemos decidido ocultar con el objetivo de no ofuscar el diagrama final, donde se mantienen aquellas secciones clave en la identificación de la arquitectura.

6. Finalmente, deberíamos encontrar un nivel de autenticación, según la justificación realizada en el capítulo donde estudiamos la arquitectura de un sistema QKD. Pero en nuestra implementación, hemos decidido sustituir ese nivel por una interfaz de acceso al nivel de aplicación, que proporcione

la clave final destilada por el sistema. El usuario, o la aplicación, que extraiga esa clave será el responsable de gestionar la autenticación del canal público utilizado.

Bajo la misma justificación que exponemos en el nivel de amplificación de la privacidad, hemos decidido esconder otras clases del sistema implementado. Algunas de las clases ocultadas son las correspondientes a la gestión de buffers, registros para la administración de errores, avisos o información (*Logs*), generadores de números aleatorios, o configuración. Estas clases han sido tenidas en cuenta a la hora de diseñar el proyecto, pero su exposición en el diagrama mostrado supondría una pérdida de nitidez sobre la arquitectura que pretendemos mostrar en esta jerarquía.

5.2.2. Sincronización

Como es habitual en un protocolo de red, la clave para el correcto funcionamiento de nuestro sistema se encuentra en la sincronización. La secuencia del código que ejecuta cada uno de los extremos es distinta, sobre todo en los niveles inferiores de la arquitectura (ver diagrama de jerarquía de clases, figura 5.1), por lo que desde el inicio debemos controlar la sincronía de nuestro sistema. Es evidente que para la ejecución del nivel físico necesitamos que los extremos de la comunicación estén sincronizados, desde todos los puntos de vista, como por ejemplo: en tiempo, parámetros de ejecución, medición de la canal cuántico, etc. Y podemos deducir que esa sincronización es necesaria en el resto de niveles, los procesos de reconciliación de bases, corrección de errores y amplificación de la privacidad, deben completarse al mismo tiempo en los dos extremos de la comunicación. Lo cual es cierto. Pero también sabemos que la destilación de una clave es un proceso costoso en tiempo, durante el que no se utilizan los dispositivos de nivel físico para la QKD, luego podemos ejecutar de forma simultánea, es decir, asíncrona, los procesos de intercambio y destilación de una clave.

Hilos de ejecución

Las siguientes figuras, 5.2 y 5.3, muestran el pseudocódigo correspondiente a los hilos de ejecución que utilizan Alice y Bob para intercambiar una clave en bruto. Estos hilos son lo suficientemente descriptivos como para comprender los conceptos de sincronía y asincronía que acabamos de comentar.

Si observamos ambas figuras descubrimos un bloque de ejecución continua, sobre el que vamos a trabajar. Ese bloque se inicia con una sentencia de sincronización, distinta en cada proceso, y a través de la cual Bob envía a Alice los parámetros de ejecución que ambos interlocutores deben compartir en el intercambio de una clave (el tamaño de esa clave). Esta sentencia de sincronización es doblemente útil, puesto que se ejecuta sobre una comunicación por red que podemos definir como bloqueante. Ese bloqueo nos asegura que la ejecución de la sentencia en Bob no se completa hasta que Alice ha recibido los parámetros, y de igual forma, la sentencia en Alice no se completará hasta que Bob envíe esos parámetros. Luego Alice y Bob ejecutan al mismo tiempo la siguiente orden, con

```
DWORD WINAPI LinkAlice_Kernel(LPVOID Param)
{
    if(Alice->Check() != LINK_OK)
        ExitThread(LINK_ERROR);
    Alice->Reset();

    Alice->SetStatus(LINK_STATUS_RUNNING);
    while(1) // for ever
    {
        Alice->RecvSync(NumberPulses, NumberFrames);
        Alice->Set(NumberPulses, NumberFrames);
        if(Alice->SetBitstream() == LINK_OK)
        {
            Alice->SendSync();
            if(Alice->GetDetections() == LINK_OK)
            Alice->RawKeyReady();
        Alice->UnsetBitstream();
        }
    }
    Alice->SetStatus(LINK_STATUS_STOPPED);
    ExitThread(LINK_OK);
}
```

Figura 5.2: Hilo de ejecución en Alice.

la que generan las secuencias de bases y valores necesarios para el intercambio de una clave. Ahora, la secuencia de sincronización se invierte, y es Alice quien se comunica con Bob para informarle de que está lista para comenzar el intercambio[4].

Una vez que ha comenzado el intercambio de pulsos, Bob sabe cuando acaba dicho intercambio, ya que es él quien envía los pulsos, y Alice también sabe cuando acaba el intercambio, puesto que comprueba cada llegada de un pulso intenso a la entrada. Por esta razón, no es necesaria ninguna sincronización posterior al intercambio de la clave. Ambos interlocutores tienen una interpretación propia de la clave intercambiada que deben proporcionar al nivel superior. El proceso a través del cual se proporciona la clave intercambiada a un nivel superior es de nuevo otro proceso de comunicación, pero ahora no interviene ninguna red, puesto que el mensaje pasa de un nivel del sistema a otro dentro de una misma máquina. Esa comunicación vuelve a servirnos de sincronización, ahora entre dos niveles que se ejecutan de forma asíncrona, por lo que no deseamos que se realice de forma bloqueante sino mediante el paso de algún mensaje.

[4]Recordemos que estamos utilizando un sistema de doble dirección, por lo que es Bob quien comienza el intercambio de una clave emitiendo trenes de pulsos intensos hacia Alice.

```
DWORD WINAPI LinkBob_Kernel(LPVOID Param)
{
    if(Bob->Check() != LINK_OK)
        ExitThread(LINK_ERROR);
    Bob->Reset(NumberPulses, NumberFrames);

    Bob->SetStatus(LINK_STATUS_RUNNING);
    while(1) // for ever
    {
        Bob->SendSync(NumberPulses, NumberFrames);
        if(Bob->SetBitstream() == LINK_OK)
        {
            Bob->RecvSync();
            if(Bob->GetDetections() == LINK_OK)
                Bob->RawKeyReady();
            Bob->UnsetBitstream();
        }
    }
    Bob->SetStatus(LINK_STATUS_STOPPED);
    ExitThread(LINK_OK);
}
```

Figura 5.3: Hilo de ejecución en Bob.

5.3. Requisitos estructurales

5.3.1. Sistema de ficheros

Forzados por la interfaz que nos proporciona el hardware, el envío de datos a Bob se realiza a través de un archivo. Es decir, cada vez que deseamos realizar el intercambio de una clave, tenemos que generar un archivo con las bases "supuestamente aleatorias" que Bob va a utilizar para la modulación de los pulsos recibidos.

Nos encontramos entonces con el primer riesgo estructural, y es que los datos utilizados por el hardware de Bob dependen en cierta medida del sistema operativo. Cualquier mecanismo de caché, junto con una sobrecarga del sistema provocará que los datos no lleguen a Bob con la suficiente antelación.

La solución adoptada hasta el momento es forzar un retardo temporal. Después de crear el archivo de datos que enviamos a Bob, esperamos un tiempo determinado antes de comenzar el intercambio de la clave.

5.3.2. Puertos USB

El hardware utilizado se conecta a nuestro entorno de desarrollo a través de una conexión por USB. A pesar de que cualquiera de los estándar, 1.1 y 2.0,

soporta una velocidad de transferencia superior a la necesaria[5], volvemos a encontrarnos con otro riesgo estructural, debido al sistema operativo encargado de la gestión de dichos puertos, que puede provocar un retardo en la comunicación.

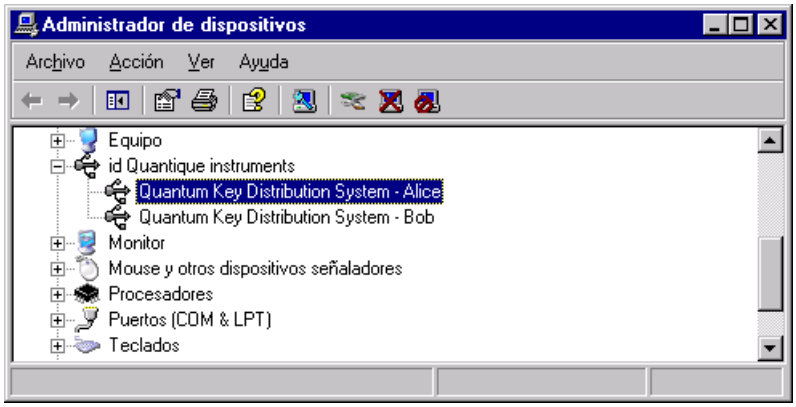


Figura 5.4: Controladores de dispositivo de los equipos id-3000 en el sistema operativo Microsoft Windows XP.

5.4. Ejecución

5.4.1. Medición de la línea

El primer paso que debemos realizar antes de poner en marcha nuestra aplicación de intercambio de claves es una medición de la línea que conecta a los extremos del sistema, Alice y Bob. Una de las principales ventajas de los sistemas de doble dirección es que están preparados para adaptarse de forma sencilla a distintas longitudes en el canal de comunicación. Aunque esta adaptación no es inmediata cuando la distancia del canal cambia de forma significativa (debido fundamentalmente a que los valores de la atenuación en Alice deben ajustarse a las características de la nueva fibra), sí es cierto que este tipo de sistemas pueden detectar pequeñas variaciones en la longitud del canal de comunicación. Las variaciones en la longitud de una misma línea de fibra óptica pueden ser debidas a cambios en las condiciones de trabajo, donde el factor a tener en cuenta es la temperatura.

Puesto que una pequeña variación de la temperatura puede afectar sobre la longitud de una fibra, que en consecuencia puede alterar notablemente el rendimiento del sistema QKD, deberemos ejecutar una medición del canal de comunicación siempre antes de comenzar un intercambio de claves. La figura 5.5 muestra un ejemplo de una medición de línea realizada con el programa *CryptoMenuBob* proporcionado por id Quantique, en el equipo que utilizamos en este proyecto: id-3000.

[5]La velocidad máxima de un puerto USB 1.1 es de 12 Mbps, mientras que la velocidad de la versión 2.0 puede alcanzar los 480 Mbps.

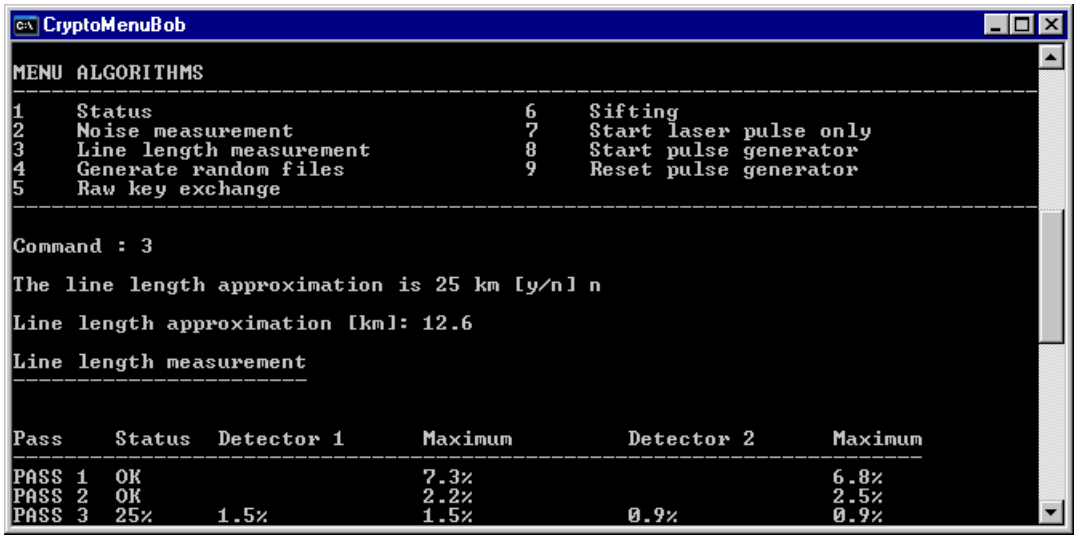


Figura 5.5: Medición de la línea.

Calibrado de línea

Una vez que el sistema QKD está en marcha también se pueden producir cambios en la temperatura que afecten directamente al rendimiento del sistema. Para corregir esta situación no realizamos una medición completa de la línea, debido a excesivo coste en tiempo que acarrea. En su lugar realizamos un calibrado de la línea que únicamente examina el entorno de la longitud actual.

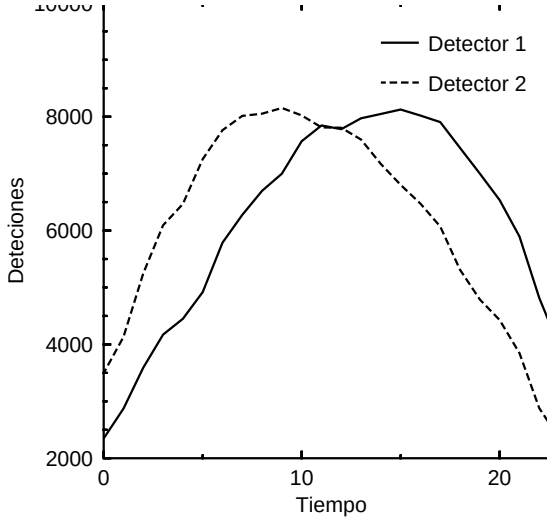


Figura 5.6: Distribución de detecciones en función del tiempo, utilizadas para el calibrado de la línea.

La figura 5.6 muestra la distribución de detecciones de cada uno de los detectores en función del tiempo. Puesto que la elección de las bases utilizadas es totalmente aleatoria, así como la codificación de la clave intercambiada, la distribución de las detecciones debe ser similar en uno y otro detector. En con-

secuencia, podemos recalibrar la longitud de nuestra línea buscando un punto cercano en el tiempo que equilibre las detecciones en ambos detectores.

La siguiente secuencia de ejecución muestra la salida de nuestro programa durante el proceso de calibrado:

```
Calibrating...
Calibrate: 00 51 06 14 02401 03869
Calibrate: 01 51 08 16 02719 04478
Calibrate: 02 51 10 18 03123 05251
Calibrate: 03 51 12 20 03855 06165
Calibrate: 04 51 14 22 04096 06872
Calibrate: 05 51 16 24 04660 06954
Calibrate: 06 51 18 26 05092 07376
Calibrate: 07 51 20 28 05817 07609
Calibrate: 08 51 22 30 06194 07700
Calibrate: 09 51 24 32 06555 07573
Calibrate: 10 52 02 10 06292 07627
Calibrate: 11 52 04 12 06693 07698
Calibrate: 12 52 06 14 06952 07587
Calibrate: 13 52 08 16 07091 07402
Calibrate: 14 52 10 18 07192 07258
Calibrate: 15 52 12 20 07267 06800
Calibrate: 16 52 14 22 07314 06337
Calibrate: 17 52 16 24 07247 06011
Calibrate: 18 52 18 26 07001 05560
Calibrate: 19 52 20 28 06504 04858
Calibrate: 20 52 22 30 05985 04380
Calibrate: 21 52 24 32 05460 04357
Calibrate: 22 53 02 10 05749 04168
Calibrate: 23 53 04 12 04552 03372
Calibration Index = 14
```

La primera de las columnas de números muestra el índice de calibrado, que corresponde con una secuencia que indexa cada una de las medidas realizadas. En la segunda columna aparece el valor del *Coarse Delay*. En las columnas tres y cuatro se muestran los valores del *Fine Delay 1* y *Fine Delay 2*, correspondientes a cada uno de los detectores. Y en las dos últimas columnas se muestran las detecciones obtenidas por el detector 1 y el detector 2 respectivamente.

5.4.2. Resultados

Finalmente, una vez que ya tenemos todo el sistema preparado, con la medición del canal de comunicación y la configuración de los parámetros de ejecución (con especial interés por la atenuación a aplicar en Alice), podemos poner en marcha el sistema ejecutando la aplicación en ambos extremos.

A continuación mostramos un ejemplo de las trazas de salida que generará la aplicación desarrollada durante su ejecución en cada uno de los extremos: Alice y Bob. Esas trazas serán enviadas por la salida estándar del sistema, por lo que las podremos ver por consola mientras se ejecuta la aplicación. Para mantener

un registro de las trazas generadas por la aplicación deberemos desviar la salida estándar del sistema hacia el archivo deseado.

Las trazas de ambos extremos producirán una línea común, idéntica en ambos interlocutores, donde se mostrará la siguiente información (en diferentes columnas):

- La línea comenzará por el carácter de admiración: '!'.
- Atenuación aplicada en Alice.
- Número promedio de fotones, μ, a la salida de Alice.
- Longitud de la línea.
- Índice secuencial del intercambio realizado.
- Tamaño de la clave intercambiada.
- Número de errores.
- Tara de error, QBER.
- Tasa de error teórico (calculado a partir de la longitud de la línea y utilizado para la estimación del tamaño de bloque inicial en la corrección de errores).
- Número de paridades intercambiadas.

Además, cada cierto número de intercambios, N, la aplicación mostrará una nueva línea similar a la anterior, donde se suprime la secuencia del intercambio realizado y se reemplaza el número de errores por la tasa de error acumulada (en los últimos N intercambios). Esta nueva línea comenzará por un carácter distinto, '#'.

Alice

La ejecución en Alice mostrará en primer lugar el nombre del archivo que contiene los valores aleatorios que se están utilizando por la aplicación. La razón por la que utilizamos un archivo como entrada de valores aleatorios se debe a que nuestros sistemas no disponen de un generador de números aleatorios, por lo que decidimos utilizar una secuencia previamente comprobada de valores perfectamente aleatorios. El tamaño de este archivo es limitado, y por lo tanto, transcurrido un periodo de tiempo aparecerá una nueva traza de salida indicando el nombre de un nuevo fichero que contiene más valores aleatorios. El número de archivos utilizados también es limitado, por lo que al consumir todos todos valores disponibles se volverá a utilizar el contenido del primer archivo utilizado.

Al comienzo de la ejecución también aparecerá el valor de la atenuación utilizado en Alice. Esa atenuación debe coincidir con el valor deseado, puesto que es un parámetro de entrada de la aplicación.

A continuación aparecerán una serie de líneas por cada uno de los intercambios realizados, que informan acerca de los resultados de cada uno de los

niveles en la pila de protocolos QKD. La primera línea mostrará el número de detecciones que ha observado Alice a través de su detector clásico, y debe coincidir con los pulsos intensos emitidos por Bob. A continuación se mostrará el número de bits de clave con los que se queda Alice después de realizar el intercambio de bases con Bob. Finalmente, el programa mostrará el número de bits de clave invertidos por el proceso de corrección de errores, así como la cantidad de paridades intercambiadas durante ese proceso.

```
Random file = Data\10megs.001
Attenuation = 14.1
Mu = opt
0.000010
Sifting[Alice]: Total detections = 6158
6158 -> 3124
Key diffs = 63 (2.02%)
QBER = 1.156190 (5.78%)
Cascade: k1 = 12
Sifting[Alice]: Total detections = 6218
6218 -> 3146
Key diffs = 53 (1.68%)
Cascade: Interchanges = 775
! 14.10 0.4 +013.48 01 03124 063 02.02 01.16 0775
QBER = 1.156190 (5.78%)
Cascade: k1 = 12
Sifting[Alice]: Total detections = 6216
6216 -> 3140
Key diffs = 65 (2.07%)
Cascade: Interchanges = 728
! 14.10 0.4 +013.48 02 03146 053 01.68 01.16 0728
QBER = 1.156190 (5.78%)
Cascade: k1 = 12
Sifting[Alice]: Total detections = 6198
6198 -> 3166
Key diffs = 69 (2.18%)
Cascade: Interchanges = 776
! 14.10 0.4 +013.48 03 03140 065 02.07 01.16 0776
...
# 14.10 00.37 +013.48 03111 058 01.88 01.16 0747
```

Bob

Al igual que ocurre con la ejecución en Alice, Bob utiliza también archivos con valores aleatorios generados previamente, por lo que encontraremos trazas de salida cada vez que la aplicación selecciona uno de los ficheros disponibles. A continuación, antes de las trazas correspondientes a cada intercambio, la aplicación mostrará un registro de las detecciones acumuladas en cada detector. Ese registro será indicativo del desplazamiento de la línea, que quedará patente por un incremento en las detecciones de uno de los detectores.

En cuanto a las trazas de cada intercambio, encontraremos las líneas que describimos a continuación. En el nivel más bajo, de la capa de enlace, la traza de salida mostrará una línea con los siguientes campos (siguiendo el orden indicado):

- Fecha y hora de la finalización del intercambio de clave.
- Índice de intercambio.
- Longitud de la línea dada por: *Coarse Delay*, *Fine Delay 1* y *Fine Delay 2*.
- Número de puertas de detección abiertas.
- Detecciones: totales, y por detector (incluyendo los porcentajes de detección de cada uno de los detectores).

Para el nivel de corrección de errores, la traza nos mostrará tres líneas distintas donde conoceremos: el número de errores corregidos (bits intercambiados), la tasa de error (QBER) y el número de paridades intercambiadas.

Un ejemplo de la traza de ejecución en Bob es el siguiente:

```
Random file = Data\10megs.009
Attenuation = 14.1
Mu = opt
Calibrating...
0.000010
Calibrate: 00 42 08 16 02100 03329
Calibrate: 01 42 10 18 02687 03850
Calibrate: 02 42 12 20 03306 05053
Calibrate: 03 42 14 22 03935 05717
Calibrate: 04 42 16 24 04289 06223
Calibrate: 05 42 18 26 04818 06806
Calibrate: 06 42 20 28 05612 07391
Calibrate: 07 42 22 30 06064 07724
Calibrate: 08 42 24 32 06501 07483
Calibrate: 09 43 02 10 06665 07873
Calibrate: 10 43 04 12 07268 07932
Calibrate: 11 43 06 14 07596 07764
Calibrate: 12 43 08 16 07716 07738
Calibrate: 13 43 10 18 07923 07546
Calibrate: 14 43 12 20 08063 07238
Calibrate: 15 43 14 22 08086 06850
Calibrate: 16 43 16 24 07951 06743
Calibrate: 17 43 18 26 07839 06276
Calibrate: 18 43 20 28 07594 05503
Calibrate: 19 43 22 30 07130 05046
Calibrate: 20 43 24 32 06519 05104
Calibrate: 21 44 02 10 06712 04924
Calibrate: 22 44 04 12 05845 04140
Calibrate: 23 44 06 14 05155 03417
```

```
Calibration Index = 12
QBER = 1.156190 (5.78%)
Cascade: k1 = 12
05/04/2007 13:55:30 0000001 43 08 16 752554 6162 3137 3021 50.9% 49.1%
Cascade: Reversed bits = 63
! 14.10 0.4 +013.48 01 03124 063 02.02 01.16 0775
QBER = 1.156190 (5.78%)
Cascade: k1 = 12
05/04/2007 13:55:58 0000002 43 08 16 749372 6225 3042 3176 48.9% 51.1%
Cascade: Reversed bits = 53
! 14.10 0.4 +013.48 02 03146 053 01.68 01.16 0728
QBER = 1.156190 (5.78%)
Cascade: k1 = 12
05/04/2007 13:56:26 0000003 43 08 16 749818 6225 3082 3134 49.6% 50.4%
Cascade: Reversed bits = 65
! 14.10 0.4 +013.48 03 03140 065 02.07 01.16 0776
...
# 14.10 00.37 +013.48 03111 058 01.88 01.16 0747
```

5.4.3. Registros de información

Si deseamos realizar una depuración más exhaustiva de la aplicación podemos utilizar unos archivos especiales, donde se almacena la información intercambiada entre los distintos niveles de la pila de protocolos QKD implementada. El nombre de cada fichero de depuración seguirá una nomenclatura especial, donde se identifica:

- Nivel y secuencia de ejecución: enlace, L, reconciliación de bases, S, clave reconciliada, K, y corrección de errores, E.
- Índice del intercambio.
- Extremo del sistema: Alice, A, o Bob, B.

La descripción del contenido de cada fichero es la siguiente:

L0_0000000_B.txt Detecciones de Bob (bits) contando las detecciones dobles.
Formato: *<índice:5> <identificador del pulso:7> <detector 1> <detector 2>*

L1_0000000_B.txt Detecciones de Bob (bits) sin contar las detecciones dobles.
Formato: *<índice:5> <identificador del pulso:7> <detector>*

S1_0000000_B.txt Bases elegidas por Bob en las detecciones obtenidas.
Formato: *<nº detección:5> <identificador del pulso:7> <base>*

S1_0000000_A.txt Lista de bases recibida por Alice.
Formato: *<nº detección:5> <identificador del pulso:7> <base>*

S2_0000000_A.txt Bases coincidentes comprobadas por Alice.
Formato: *<nº detección:5> <identificador del pulso:7> <base>*

K2_0000000_A.txt Valores utilizados por Alice en las bases coincidentes.
Formato: <*nº detección:5*> <*identificador del pulso:7*> <*valor*>

S2_0000000_B.txt Lista de bases coincidentes recibida por Bob.
Formato: <*nº detección:5*> <*identificador del pulso:7*> <*base*>

K2_0000000_B.txt Valores utilizados por Bob en las bases coincidentes.
Formato: <*nº detección:5*> <*identificador del pulso:7*> <*valor*>

E2_0000000_A.txt Clave en Alice sin corrección de errores.

E2_0000000_B.txt Clave en Bob sin corrección de errores.

E3_0000000_A.txt Clave en Alice después de la corrección de errores.

E3_0000000_B.txt Clave en Bob después de la corrección de errores.

Banderas de ejecución

Los ficheros que hemos descrito en el apartado anterior no se generan de forma habitual en la ejecución de la aplicación, debido especialmente a la pérdida de rendimiento que supone la gestión de los mismos. Para activar este registro utilizamos unas banderas de compilación que definimos mediante la siguiente secuencia de código en C++:

```
#define CREATE_LOG_FILE_L0 1
#define CREATE_LOG_FILE_L1 0
#define CREATE_LOG_FILE_S1 0
#define CREATE_LOG_FILE_S2 0
#define CREATE_LOG_FILE_K2 0
#define CREATE_LOG_FILE_E2 0
#define CREATE_LOG_FILE_E3 0

#define TEST_RAWKEY 0
```

Para activar el registro de cada uno de los ficheros, debemos introducir un 1 al final de su definición.

Salida

Una captura parcial de la información registrada en los archivos descritos es la siguiente (el archivo mostrado es el correspondiente a las detecciones obtenidas por Bob en la capa de enlace, L0_0000042_B.txt):

```
00000 0000037 1 0
00001 0000115 1 0
00002 0000244 0 1
00003 0000464 0 1
00004 0000517 0 1
00005 0000598 1 0
00006 0000628 1 0
00007 0000708 0 1
```

```
00008 0000796 0 1
00009 0000935 1 0
...
```

Parte III

Redes

Capítulo 6

Integración

El objetivo final de un sistema QKD no es funcionar de manera aislada; sus beneficios sólo pueden conseguirse si se integran en la infraestructura de comunicaciones actual. La integración significaría terminar con su carácter punto a punto para pasar a disponer de un sistema por el que dos puntos cualesquiera de una red de comunicaciones pueden compartir una clave secreta. Este es el objetivo de las redes de distribución cuántica de claves, QKDN[1].

Pero antes de meternos de lleno en el diseño de estas redes, tenemos que dar un pequeño paso atrás para completar el desarrollo de nuestro sistema, QKDS[2]. En los capítulos anteriores, construimos un sistema a nivel físico, y diseñamos e implementamos la arquitectura de ese sistema, obteniendo dos equipos capaces de intercambiar una clave segura. Nos queda un último cometido para completar el desarrollo de una solución final. Es el momento de integrar la distribución cuántica de claves con las herramientas de cifrado actuales, y comprobar que:

- La nueva solución aporta beneficios considerables a los sistemas de cifrado actuales, por lo que resulta atractiva la idea de actualizar cualquier solución existente. Entendiendo por beneficio la ganancia en seguridad que obtiene la distribución cuántica de claves frente a mecanismos como el cifrado asimétrico[3].

- La integración de una solución completa es viable, y esa solución puede habilitarse en los entornos donde hoy día se requiere algún mecanismo de seguridad.

- La solución final es estable, y tiene visos de ser duradera en el tiempo. Debemos exigir que la integración de nuestro sistema se realice sobre un núcleo esencial, arraigado en los sistemas actuales, que asegure su longevidad. En otras palabras, debemos evitar que una solución tecnológicamente moderna quede obsoleta por culpa de una inadecuada integración.

[1] Quantum Key Distribution Network.

[2] Quantum Key Distribution System.

[3] Hablamos exclusivamente de beneficio, y no de la relación beneficio-coste puesto que estamos trabajando con una tecnología en desarrollo, relativamente joven, sobre la que aún deben completarse distintas fases de investigación, y cuyo coste en el momento actual puede quedar lejos de ser rentable, excepto en ciertos nichos de mercado.

La integración de nuestro sistema es un proceso íntimamente ligado a los protocolos de red, y por esta razón, hemos decidido estudiar su contenido dentro de la parte de redes del presente proyecto.

6.1. Cifrado y distribución actual de claves

En la actualidad son dos los tipos de cifrado que solemos utilizar, y que conocemos bajo el nombre que describe su modo de trabajo: simétrico y asimétrico. Del primero de ellos, el **cifrado simétrico**, no vamos a comentar gran cosa en este proyecto, y es que la criptografía cuántica no afecta en nada a esta disciplina de la criptografía convencional. De él tan sólo nos interesa conocer un detalle: funciona bajo el supuesto de que dos interlocutores comparten una clave secreta, con la que ambos extremos pueden cifrar y descifrar cada mensaje intercambiado. En consecuencia, el cifrado simétrico viene a resolver tan sólo uno de los problemas en la búsqueda de un sistema de comunicación seguro: el cifrado propiamente dicho; pero deja otro problema pendiente: la distribución de claves.

De forma complementaria al cifrado simétrico surge el **cifrado asimétrico**, que basa su funcionamiento en la utilización de dos claves, una pública y otra privada, que estudiaremos con más detalle a continuación. Pero antes de comenzar ese estudio queremos destacar un hecho importante, y es que el cifrado asimétrico sí proporciona un mecanismo para el intercambio de claves, siento ésta la razón por la que nos detendremos un poco más en su estudio, para comprender cómo funciona y en qué medida la criptografía cuántica puede reemplazar al cifrado asimétrico.

6.1.1. Cifrado asimétrico o de clave pública

La distribución cuántica de claves viene a ser una alternativa para los mecanismos actuales de intercambio de claves, basados en lo que conocemos como **cifrado de clave pública**. Estos mecanismos de cifrado utilizan dos claves, una pública, K_{Pu}, y otra privada, K_{Pr}, con las que puede cifrar y descifrar un mensaje respectivamente; y por esta razón, a este tipo de mecanismos se les conoce también con el nombre de sistemas de cifrado asimétrico. La seguridad de estos mecanismos reside en el hecho de que la clave privada, K_{Pr}, sólo es conocida por la persona o entidad que desea recibir y descifrar un mensaje, mientras que la clave pública puede ser proporcionada a todos aquellos individuos o entidades que deseen enviar un mensaje cifrado al único conocedor de la clave privada, que es el única entidad capaz de descifrar el mensaje original. Evidentemente se asume que el conocimiento de K_{Pu} no proporciona ninguna información sobre K_{Pr}. El funcionamiento es el siguiente: cuando un interlocutor (llamémoslo Alice) desea enviar un mensaje cifrado a otro interlocutor (Bob), el primero, Alice, localiza la clave pública del segundo, Bob, y cifra el mensaje que desea enviar con dicha clave, K_{Pu}, obteniendo un mensaje cifrado, $K_{Pu}(M)$, que sólo Bob puede descifrar con su clave privada, K_{Pr}.

$$M = K_{Pr}\left(K_{Pu}(M)\right)$$

Una vez que comprendemos el modo de funcionamiento del cifrado asimétrico, surgen de forma inmediata dos preguntas ¿qué mecanismo de cifrado debemos utilizar, simétrico o asimétrico?, y ¿por qué? A primera vista, sin más datos que los conocidos hasta el momento, parece evidente que el cifrado asimétrico es una alternativa mejor frente al cifrado simétrico, debido a la mayor seguridad que proporciona la disponibilidad de un soporte en el intercambio de claves. Utilizando el cifrado asimétrico, un usuario tan sólo tiene que proporcionar su clave pública al interlocutor que desea ponerse en contacto con él, esperar el mensaje cifrado con dicha clave (la clave pública), y descifrar ese mensaje con su clave privada, manteniendo la plena seguridad de que nadie ha podido descifrar antes el mensaje[4]. Ahora bien, el cifrado asimétrico posee un inconveniente práctico, que proviene de su implementación y que reside en el tiempo de cómputo necesario para realizar el cifrado y descifrado de un mensaje. Esta limitación del cifrado asimétrico convierte a este mecanismo en una estrategia nada interesante para el cifrado de grandes volúmenes de información.

La solución final, de uso muy común en la actualidad, es utilizar el cifrado asimétrico (o de clave pública) para intercambiar una clave secreta entre dos interlocutores. A esa clave se le suele conocer con el nombre de clave de sesión, y se utiliza para cifrar el resto de la comunicación entre los dos interlocutores a través de un mecanismo de cifrado simétrico, cuyo rendimiento es considerablemente mejor al del cifrado asimétrico.

Firma digital

El proceso anteriormente descrito se puede diseñar de modo que se pueda realizar en el sentido inverso, es decir, cifrando inicialmente un mensaje con la clave privada, $K_{Pr}(M)$, y descifrando después dicho mensaje con la clave pública, $M = K_{Pu}\left(K_{Pr}(M)\right)$, obtenemos el mensaje original. La peculiaridad de este proceso inverso es que un mensaje descifrado a través de la clave pública sólo puede haber sido cifrado utilizando la clave privada, lo que permite implementar un nuevo concepto, la **firma digital**. La forma de desarrollar esta idea es la siguiente: el interlocutor que desea firmar su mensaje escoge una segmento del mensaje original, $M_0 = hash(M)$, cifra ese segmento con su clave privada, $S = K_{Pr}(M_0)$, y envía a otro interlocutor el mensaje original incluido el segmento cifrado, es decir, la firma, $M + S$. El receptor del mensaje sólo tiene que comprobar que, al descifrar la firma incluida, S, con la clave pública del emisor, obtiene un segmento del mensaje recibido, ¿$hash(M) = K_{Pu}\left(S\right)$?

Seguridad

Hoy día, los sistemas de cifrado asimétrico utilizados para la distribución de claves, como pueden ser el RSA[5] o Diffie-Hellman[6], son considerados seguros

[4]Una vez más queda un detalle pendiente, la certificación de que la clave pública utilizada corresponde realmente al usuario con el que deseamos comunicarnos. Pero ese requisito de certificación es común a ambos mecanismos de cifrado, simétrico y asimétrico, por lo que no lo contemplamos en la comparación de ambos.

[5]Propuesto por R. Rivest, A. Shamir y L. Adleman.

[6]Debido a Whitfield Diffie y Martin Hellman.

"en la práctica", debido a que estos esquemas basan su seguridad en la complejidad computacional de un problema como es la factorización de números grandes.

Ahora bien, no existen parámetros concretos que definan la seguridad de estos procedimientos, ni de las claves intercambiadas, siendo más un arte, o misterio, la elección de valores como: el tamaño de la clave a intercambiar, o el volumen máximo de datos que podemos cifrar con una clave. Sin embargo, sí somos conocedores de un factor importante, y es que la seguridad de un mecanismo de cifrado decrece exponencialmente con cada bit conocido de la clave. Somos conscientes, por lo tanto, de la rigurosidad con la que debemos examinar y evaluar cualquier mecanismo de distribución de claves, y aquí es donde aparece uno de los factores que quizá, en un futuro, pueda constituir una de las justificaciones clave para la utilización de la criptografía cuántica. Hablamos del algoritmo de Shor[7].

Entornos de aplicación

Hasta el momento hemos propuesto a la criptografía cuántica como alternativa para los mecanismos de distribución de claves, pero su similitud funcional con los sistemas de cifrado asimétricos puede plantearnos escenarios alternativos. Para ello, debemos conocer en qué entornos se aplica el cifrado de clave pública, y bajo qué condiciones funciona.

Analizando los entornos de aplicación del cifrado asimétrico encontramos dos principalmente:

- La **distribución de claves**. Es la parte que acabamos de ver. Somos conscientes de que el cifrado asimétrico se utiliza principalmente para la distribución de claves, y esa es la razón fundamental por la que lo estudiamos.
- Pero además, existe otra aplicación del cifrado asimétrico que no hemos visto hasta el momento: la **autenticación**. Hasta el momento hemos hablado tan sólo de la aplicación de la criptografía cuántica en cuanto a la distribución de claves, QKD. Pero, como veremos en los capítulos siguientes, los protocolos QKD estudiados también tienen una aplicación directa en procesos como la certificación, basados en la identificación y autenticación de usuarios.

6.1.2. Cifrado de Vernam

A comienzos del siglo XX encontramos la propuesta de uno de los mecanismos de cifrado más conocido, y al mismo tiempo sencillo, que identificamos bajo en nombre de cifrador de Vernam[8]. Como ya hemos comentado, su funciona-

[7] El algoritmo de Shor es uno de los desarrollos teóricos más importantes en computación cuántica, que destaca por el hecho de encontrar un procedimiento eficiente para resolver el problema de la factorización de números enteros. La posibilidad que plantea una futura implementación de este algoritmo en un ordenador cuántico supone, en algunas corrientes de opinión, uno de los riesgos principales de la criptografía asimétrica. Sin lugar a dudas, su implementación —posible o no— sería el final de los sistemas de cifrado asimétricos actuales, RSA y Diffie-Hellman entre otros.

[8] Nombre adquirido del autor que propuso su utilización, Gilbert Vernam.

miento es bastante simple, consistiendo en un único cálculo de una operación XOR (ó exclusivo) entre cada segmento del mensaje a cifrar y una clave previamente compartida. Esta única operación sirve tanto para el cifrado, como para el descifrado del mensaje, y debe aplicarse sobre segmentos de tamaño similar a la clave utilizada.

Existe una variante del cifrador de Vernam, que conocemos bajo el nombre de *one-time pad*, o cuaderno de un sólo uso, cuya seguridad ha sido la única demostrada analíticamente (además de la QKD). La modificación que incluye este nuevo mecanismo es la actualización de la clave para cada segmento del mensaje a cifrar, por lo que, su implementación requiere de un clave total de tamaño igual al del mensaje. El interés por este mecanismo de cifrado reside en el hecho de que es la única forma de transportar una clave, a través de canales de comunicación independientes, manteniendo el nivel de seguridad original[9].

6.2. Integración con los sistemas de cifrado actuales

IPsec es, con casi toda certeza, el protocolo de seguridad más utilizado en la actualidad para la conexión de redes seguras. Entre las razones que justifican su utilización destaca el hecho de que IPsec es un estándar, que su uso es obligatorio en las futuras redes IPv6[10], y que el funcionamiento final del protocolo es totalmente transparente para el usuario. Parece, a primera vista, que IPsec puede ser el candidato ideal para integrar nuestro sistema de distribución cuántica de claves en un entorno real de trabajo.

Sin lugar a duda, la primera de las justificaciones[11] es más que suficiente para forzar la búsqueda de una solución que permita integrar nuestro sistema dentro de IPsec. Pero debemos añadir otro motivo, sin llegar a hacer apología de este razonamiento, y es que las referencias encontradas de otras implementaciones similares han elegido esta estrategia [52] [53].

En consecuencia, nos vemos obligados a estudiar dicho protocolo de seguridad, con el objetivo de encontrar los componentes que debemos modificar para utilizar nuestro sistema de intercambio de claves. Pero también estudiaremos como sería la integración desde niveles superiores al de red[12], como puede ser el nivel de transporte[13] y el nivel de aplicación.

[9] En realidad, es imposible mantener un nivel de seguridad idéntico al original, puesto que, cada extremo de una comunicación supone un punto de riesgo adicional al sistema o transmisión global. Este riesgo, o pérdida de seguridad, no viene determinado por el método de cifrado sino por la debilidad inherente que lleva asociada cualquier implementación práctica de un mecanismo de cifrado: defectos en su implementación, acceso a nivel físico, etc.

[10] En la práctica actual, la gran mayoría de las redes utilizan todavía la versión del protocolo de Internet IPv4.

[11] El hecho de que IPsec es un estándar.

[12] Nivel IP.

[13] Nivel TCP.

6.2.1. IPsec. Seguridad a nivel de red, IP

Analizar con detalle el protocolo IPsec[14] es algo que se sale del objetivo final de este proyecto, no por falta de interés sino por la extensión del mismo, ya que se trata de un protocolo que, como veremos, soporta un número considerable de combinaciones entre mecanismos de seguridad, modos de funcionamiento y gestión de la información[15]. A pesar de esto, nos vemos obligados a profundizar en aquellos aspectos del protocolo que están relacionados con la gestión de claves de sesión y el intercambio de las mismas, puesto que son los mecanismos que pretendemos cambiar con nuestro sistema de distribución cuántica de claves.

Comenzaremos estudiando la arquitectura de IPsec, a grandes rasgos, para pasar a ver uno a uno aquellos aspectos más importantes del protocolo, como son: los modos de funcionamiento, los tipos de protección, la gestión de la información; y acabar con la parte que más nos interesa del protocolo: la gestión[16] de claves.

Arquitectura

Resulta complicado exponer de forma clara y concisa la arquitectura de un protocolo tan complejo como es IPsec. Para empezar, el concepto de protocolo no es lo suficientemente extenso como para ser aplicado a IPsec, puesto que al desmenuzar este "mecanismo" de seguridad, lo primero que descubrimos es un nuevo conjunto de protocolos, de carácter —eso sí— más específico. Pero ésta no es la única dificultad que nos encontramos, existe otro problema añadido al carácter genérico de este protocolo, y es que no podemos encajarlo en un lugar determinado de la capa de red, dentro de la pila de protocolos. Por su nombre, podemos deducir de forma más o menos evidente que trabaja en el nivel IP, pero como veremos a continuación, dependiendo del modo de trabajo en el que se esté ejecutando el protocolo, su situación real se localizará por encima o debajo de ese nivel IP.

Sin una definición específica como protocolo, ya que éste consta a su vez de distintos protocolos. Y sin una localización concreta, es aventurado mostrar una estructura lógica comprensible; por lo que el diagrama siguiente (figura 6.1) es sólo una aproximación a la arquitectura de IPsec.

De esta arquitectura nos interesa destacar la presencia de tres bloques:

- Un primer bloque que identifica el protocolo propiamente dicho, IPsec, y que consta de dos herramientas que definirán los mecanismos de seguridad

[14]

> *The spelling "IPsec" is preferred and used throughout this and all related IPsec standards. All other capitalizations of IPsec (e.g., IPSEC, IPSec, ipsec) are deprecated. However, any capitalization of the sequence of letters "IPsec" should be understood to refer to the IPsec protocols.*

RFC-4301: Security Architecture for the Internet Protocol.

[15]Toda la información extraída acerca del funcionamiento del protocolo IPsec, así como el estudio de otras estrategias de integración a nivel de red, transporte o aplicación, ha sido contrastada con expertos en la materia: Dr. Nicolás Barcia Vazquez, y Dr. Jorge Dávila Muro, profesores en la Universidad Politécnica de Madrid.

[16]Almacenamiento y distribución.

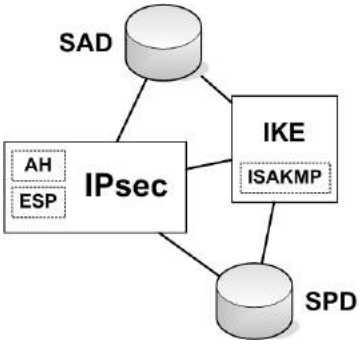


Figura 6.1: Arquitectura de IPsec.

del protocolo, AH[17] y ESP[18].

- También localizamos dos bases de datos, SAD[19] y SPD[20].
- Un gestor de claves, IKE[21], gestionado a través de un protocolo propio, ISAKMP[22].

A continuación, veremos con más detalle cada uno de estos componentes.

Modos de trabajo[23]

Una de las dudas que nos surge al estudiar el modo de funcionamiento de IPsec es acerca del instante en el que el protocolo accede a los paquetes del nivel IP para cifrarlos siguiendo los criterios de seguridad definidos. Deseamos conocer en qué momento IPsec decide coger la información disponible y cifrarla, pero no encontramos una respuesta precisa debido a que ese salto, dentro de la capa IP, depende del *modo de trabajo* que se este aplicando.

La manera en la que IPsec va a trabajar con una trama del nivel IP debe estar definida en una asociación de seguridad, SA[24], por lo que es independiente y exclusiva para la conexión con un equipo determinado, o lo que es igual, para una dirección IP en concreto. En los apartados siguientes veremos que son estas "asociaciones de seguridad" y para qué se utilizan.

Visto esto, sabemos que los modos de trabajo disponibles en IPsec son dos:

1. **Modo transporte**. Autentica y cifra tan sólo el bloque de datos de un paquete IP, dejando intacta su cabecera. Por lo que en el fondo, este modo de trabajo tan sólo realiza una modificación de los datos procedentes de la

[17]Authentication Header.
[18]Encapsulating Security Payload.
[19]Security Association Database.
[20]Security Policy Database.
[21]Internet Key Exchange.
[22]Internet Security Association and Key Management Protocol.
[23]RFC 2401, Security Architecture for IP.
[24]Security Association.

capa anterior[25], y en consecuencia, podemos considerar el modo de transporte como un protocolo de nivel superior al IP.

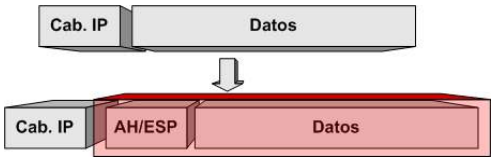


Figura 6.2: Modo transporte.

La figura anterior (6.2) muestra cómo queda un paquete IP modificado por IPsec, cuando éste se ejecuta en modo transporte. La zona sombreada es parte del paquete que queda protegida por el protocolo de seguridad.

2. **Modo túnel**. Autentica y cifra el paquete IP al completo. La implicación directa de este modo de trabajo es que el protocolo necesita generar una nueva cabecera, lo que provoca un aumento de la información de control necesaria para transmitir un mensaje, que se traduce en una pérdida de eficiencia. A cambio, conseguimos que dos extremos se comuniquen como si estuvieran conectados el uno al otro, directamente, sin que nadie pueda leer o modificar los paquetes que intercambian.

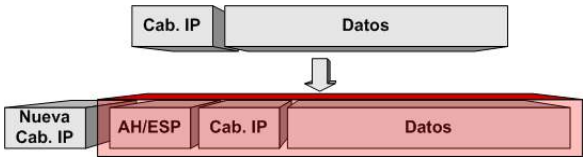


Figura 6.3: Modo túnel.

En esta nueva figura (6.3) podemos comprobar como el paquete al completo se encaja dentro de la zona sombreada, o zona de seguridad. Por lo que, al contrario que ocurría con el modo de trabajo anterior, ahora obtenemos una localización del protocolo por debajo del nivel IP.

Quizá, antes de llegar a este punto se haya hecho la siguiente pregunta, ¿por qué tenemos tanto interés en conocer el instante en el que se ejecuta IPsec? La razón es sencilla, y es que conocer dicho instante nos da una idea bastante aproximada de donde se está ejecutando el protocolo, lo que nos allana el camino de cara a comprender cómo funciona el protocolo, así como en la posible implementación de un mecanismo similar.

La figura 6.4 nos muestra bastante bien la primera conclusión a la que llegamos en el estudio del protocolo IPsec, con dos capas (encima y debajo de la capa IP), como si de dos protocolos distintos se tratara, en función del modo de trabajo.

[25]En la pila de protocolos TCP/IP, el nivel superior es la capa de transporte TCP.

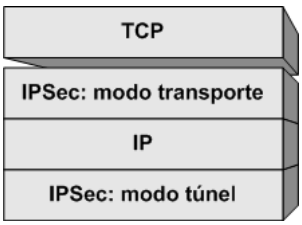


Figura 6.4: Pila de protocolos TCP/IP con IPsec.

Tipos de protección

Como ya vimos en apartados anteriores de este capítulo, la criptografía cuántica sólo viene a ser una alternativa para ciertos procedimientos de la criptografía convencional, como es la distribución de claves. Otros aspectos como el cifrado simétrico —o cifrado propiamente dicho— de un mensaje, no se ven afectados. Ahora bien, si intentamos mejorar los mecanismos de distribución de claves mediante el uso de una tecnología avanzada, resulta poco menos que necesario asegurar su integración con un sistema criptográfico actualizado. Y ese es nuestro único interés aquí, comprobar que IPsec nos aporta los cuatro pilares fundamentales en seguridad:

- Autenticación. ambos extremos en la comunicación poseen la certeza de que el otro extremo está validado, es decir, que es quien dice ser.
- Integridad. Permite asegurar que un mensaje no ha sido modificado.
- Confidencialidad. Es el cifrado propiamente dicho. Asegura que un mensaje sólo puede ser leído por la persona que conozca la clave correcta con la que descifrar ese mensaje.
- No-repudio.

Para ofrecer estos servicios, IPsec utiliza dos protocolos:

1. **Cabecera de autenticación**, AH. Ofrece los servicios de autenticación, integridad y no-repudio.
2. **Encapsulación segura de la carga útil**[26], ESP. Incluye todos los servicios proporcionados por la cabecera de autenticación, AH, añadiendo el servicio de confidencialidad, es decir, el cifrado de los datos. DES[27], Triple DES, AES[28], IDEA[29]

[26]Hoy día existen países en los que el cifrado de la información está prohibido, por lo que la utilización de protocolo ESP no está permitida.
[27]Data Encryption Standard.
[28]Advanced Encryption Standard.
[29]International Data Encryption Algorithm.

Información gestionada por IPsec

Una de los puntos fuertes de IPsec es su versatilidad. Como hemos visto hasta el momento, IPsec incorpora múltiples algoritmos de cifrado (DES, 3DES, ...), distintos modos de trabajo (túnel o transporte), otros tantos mecanismos de seguridad (AH o ESP). Pero además, el funcionamiento de IPsec puede ser híbrido, por lo que puede estar utilizando distintas estrategias de seguridad en función de la conexión realizada. Para esto, IPsec requiere de un registro central donde gestionar las estrategias específicas y generales del protocolo, que se almacenan en forma de:

1. **Políticas de seguridad**, SPD. Es una base de datos donde IPsec registra las directrices generales del protocolo, es decir, las directrices que se aplicarán a todas las conexiones definidas.

2. **Asociaciones de seguridad**, SAD. Es un registro específico de cada conexión donde el protocolo almacena información temporal acerca de dicha conexión.

   En una asociación de seguridad encontraremos algunos de los parámetros más importantes de la seguridad de nuestro sistema, como son: el tiempo durante el cual se ha estado utilizando una misma clave de sesión, o el volumen de datos que se ha cifrado con esa clave.

La gestión de claves

Ya sólo nos queda por ver la sección más interesante, desde nuestro punto de vista, que es la gestión de claves. Sabemos cómo puede trabajar IPsec, qué mecanismos de seguridad implementa, y donde registra la información necesaria, pero nos falta saber de donde obtiene las claves para cifrar y cómo las gestiona. Y para ello, IPsec utiliza dos nuevos protocolos:

1. **Intercambio de claves** a través de Internet, IKE. Es el protocolo encargado de intercambiar las claves de sesión y por lo tanto, es la parte de IPsec que queremos reemplazar por nuestro sistema de distribución cuántica de claves.

2. **ISAKMP**. Es el protocolo encargado de la gestión de las claves de sesión, una vez que estas han sido intercambiadas (compartidas). Cada clave de sesión es registrada en una asociación de seguridad y utilizada para una conexión en concreto bajo los criterios[30] definidos en esa SA.

Limitaciones de la seguridad

Uno de los problemas al que nos enfrentamos con el cifrado a nivel IP aparece en la retransmisión de paquetes. Esa retransmisión puede ser causada por la presencia de errores en la línea, de una sobrecarga del tráfico, o en el peor de

[30] Como ya comentamos, entre los criterios de una asociación de seguridad destaca el límite de utilización de una clave, definido en función de tiempo y volumen de cifrado.

los casos, por un atacante que pretende descifrar la comunicación extrayendo información de los paquetes retransmitidos.

Si la retransmisión se realiza a nivel IP, no existe riesgo alguno para la seguridad de nuestros datos, ya que el paquete retransmitido vuelve a ser idéntico al original. El problema aparece cuando se produce la retransmisión en el nivel de transporte, puesto que en ese caso, la trama retransmitida sería idéntica a la original salvo por el cambio del número de secuencia registrado en la cabecera TCP. El riesgo reside en el hecho de que ciframos dos mensajes prácticamente idénticos con una misma clave, donde conocemos con relativa precisión la posición y el valor diferenciado en ambos mensajes.

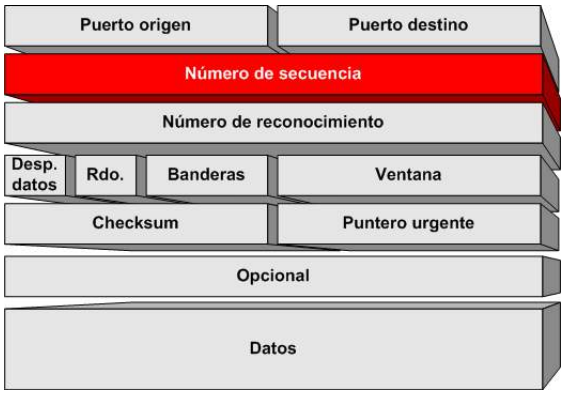


Figura 6.5: Cabecera TCP. Aparece marcado de distinto color el campo que incluye el número de secuencia, que constituye uno de los riesgos en las retransmisiones a nivel TCP.

No vamos a entrar en detalle para explicar la forma en la que podemos provocar la retransmisión de un paquete TCP, o el ataque al protocolo IPsec por retransmisión de paquetes. Pero sí es interesante conocer que un segmento TCP puede tener un tamaño máximo de 64K, lo que supone una cantidad de información considerable, y un riesgo muy alto el asumido por la retransmisión de un paquete.

Evidentemente, damos por supuesto que cualquier implementación —medianamente decente— de IPsec, ha tenido en cuenta este riesgo estructural, que a primera vista podemos solucionar de varias formas:

- En primer lugar, podríamos pensar en eludir el cifrado de cualquier cabecera TCP, o similar. Pero siempre es impredecible el comportamiento de los protocolos que se encuentran por encima del nivel de transporte, y la aparición de un problema similar en una capa superior.

- Ampliando el objetivo a una solución final, el planteamiento a resolver es evitar que una parte del mensaje sea razonablemente fácil de "adivinar", y una solución puede ser comprimir el mensaje. Cuando compactamos una información, lo primero que realizamos es la eliminación de redundancias, lo que en el fondo se traduce en contenido de distribución aleatoria[31]. Por

[31] Sino perfectamente aleatoria, sí más aleatoria que el contenido inicial.

lo tanto, al aplicar el cifrado sobre el mensaje comprimido, no sólo estamos evitando la presencia de contenidos fáciles de adivinar, sino que incrementamos la seguridad del mensaje final cifrado.

Conclusiones

La conclusión principal a la que llegamos es que IPsec es un protocolo muy completo, actualizado, y disponible a un nivel de la capa de red que lo convierte en transparente para todos los usuarios. Además, por su diseño los mecanismos de criptografía cuántica pueden integrarse dentro del protocolo, sustituyendo tan sólo uno de sus módulos —probablemente el más importante—, IKE, así como el protocolo asociado, ISAKMP. Como principal inconveniente surge el hecho de que, en función del sistema operativo, el protocolo puede encontrarte integrado dentro del propio sistema, lo que dificulta —llegando incluso a imposibilitar— la modificación del mismo.

Si el objetivo es realizar una demostración de funcionamiento, u obtener un primer protocolo de conexión para la implementación de una red QKD; incluso como partida para un desarrollo modular, puede ser interesante comenzar en un nivel de red superior, como los descritos a continuación.

6.2.2. SSL. Seguridad a nivel de transporte, TCP

Si seguimos un camino ascendente en la pila de protocolos TCP/IP, el siguiente nivel que encontrarmos es la capa de trasporte. El objetivo sigue siendo el mismo, buscar la mejor forma de integración de nuestra solución. Pero en este nivel no somos capaces de encontrar una alternativa aceptable a la de una integración de nivel IP.

La dificultad en el desarrollo será bastante similar a la del nivel anterior, tenemos que modificar una capa, por lo general, fuera del alcance del usuario e integrada en el sistema operativo. La transparencia proporcionada por un desarrollo a este nivel puede ser válida de cara a un usuario, es decir, podemos cifrar una comunicación desde este nivel de transporte sin que ello suponga un trabajo adicional para el usuario. Pero ahora, al encapsular entre los dos extremos una transferencia estamos metiendo una capa de transporte sobre otra capa de transporte, que como veremos a continuación, es algo ineficiente y de poca utilidad.

A pesar de las críticas mostradas para una integración a nivel de transporte, puede darse el caso en el que deseemos adaptar un mecanismo de cifrado de este nivel con nuestro sistema de distribución de claves. La idea no es tan descabellada, y puede ser factible principalmente por el hecho de que un QKDS aporta un camino sencillo y fiable para el intercambio de claves. Lo que puede ser adaptado fácilmente a cualquier estándar de seguridad en este nivel: SSL, TSL o Socks v5.

6.2.3. SSH. Seguridad a nivel de aplicación

Seguramente, la forma más sencilla de integrar un sistema de distribución de claves es a través del nivel de aplicación. Las razones son evidentes y es que:

- No dependemos en primer lugar del sistema operativo. En los casos anteriores, de integración a nivel de red y nivel de transporte, necesitamos acceso a la pila de protocolos TCP/IP implementada por el sistema. Lo que implica que, o bien tenemos acceso al código fuente del sistema operativo, o necesitamos un mecanismo alternativo para procesar la información que se transporta entre las distintas capas de la pila de protocolos.

- Disponemos de un número considerable de aplicaciones que nos permiten crear un túnel de comunicación cifrado. Ese número se reduce cuando lo que buscamos son aplicaciones *open source* que nos permitan incorporar nuestro sistema de distribución de claves.

- A este nivel es fácil encontrar un entorno de desarrollo, lo suficientemente versátil, para que nos permita desarrollar una solución propia en un espacio de tiempo considerablemente corto.

- La depuración es otro punto a favor de los desarrollos a nivel de aplicación.

Siendo consecuentes con lo expuesto, parece recomendable comenzar la integración de nuestro sistema en este nivel. Quizá buscando alguna aplicación *open source* que nos permita adaptar nuestra solución de forma rápida y sencilla. Pero sin perder de vista el objetivo de desarrollar una aplicación propia, en la que implementaremos "de forma modular" los componentes principales de un sistema criptográfico, que luego podamos reutilizar en los niveles de integración inferiores (nivel de transporte, TCP, y nivel de red, IP).

Es importante ser conscientes de que cualquier implementación a este nivel es siempre menos eficiente y efectiva que las soluciones propuestas en los niveles anteriores, principalmente por la falta de alcance o limitación de acceso sobre otras capas y servicios de red.

Transparencia

El principal inconveniente de utilizar un túnel a nivel de aplicación es la falta de transparencia que éste ofrece. El usuario está obligado a configurar todos los servicios o aplicaciones que quiera securizar a través de ese túnel, y esos servicios o aplicaciones tienen que soportar la utilización de un proxy.

Dos requisitos, configuración y soporte, que son suficientes para difuminar la idea de utilizar el nivel de aplicación, pero podemos paliar con la utilización de interfaces de red como muestra el siguiente script:

```
#!/bin/sh

PPPD=/usr/sbin/pppd
SSH=/usr/bin/ssh

test -x $PPPD || exit 0
test -x $SSH || exit 0

HOST=cecilia.ls.fi.upm.es
```

```
USER=jmartinez

CMD="$SSH -P $HOST -l$USER -o Batchmode=yes sudo $PPPD nodetach notty noauth"

$PPPD updetach noauth passive pty $CMD ipparam \
    vpn 192.168.42.1:192.168.42.2
```

Al ejecutar el script anterior deberemos obtener un mensaje de salida como el siguiente:

```
Using interface ppp0
Connect: ppp0 <--> /dev/pts/5
Deflate (15) compression enabled
Cannot determine ethernet address for proxy ARP
local  IP address 192.168.42.1
remote IP address 192.168.42.2
```

Que será indicativo de que la conexión se completó correctamente. La salida nos muestra el nombre de la nueva interfaz de red que se ha creado, en este caso: ppp0. Y para comprobar que dicha interfaz de red se creó correctamente, deberemos ejecutar el siguiente comando, con una salida similar a la mostrada a continuación:

```
# ip addr show ppp0
6: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP,10000> mtu 1500 qdisc pfifo_fast qlen 3
    link/ppp
    inet 192.168.42.1 peer 192.168.42.2/32 scope global ppp0
```

En este momento ya tenemos los dos extremos conectados, y comparten una conexión directa a través de un protocolo SSH. Esa conexión se enmascara con una interfaz de red, que podemos utilizar como si de un nivel IP se tratara. Así por ejemplo, si ejecutamos un comando PING, podremos alcanzar el otro extremo.

```
# ping 192.168.42.2
PING 192.168.42.2 (192.168.42.2) 56(84) bytes of data.
64 bytes from 192.168.42.2: icmp_seq=1 ttl=64 time=93.6 ms
64 bytes from 192.168.42.2: icmp_seq=2 ttl=64 time=86.1 ms
64 bytes from 192.168.42.2: icmp_seq=3 ttl=64 time=87.3 ms
64 bytes from 192.168.42.2: icmp_seq=4 ttl=64 time=86.6 ms

--- 192.168.42.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 86.149/88.428/93.612/3.022 ms
```

El paso final para dejar una red configurada sería establecer las rutas de comunicación entre los extremos, y si así se deseara, las limitaciones de tráfico y calidad de servicio.

Una vez más, al igual que ocurría en el nivel de transporte, hemos construido una capa TCP sobre otra capa TCP, con la consiguiente pérdida de eficiencia que pasamos a describir a continuación.

Túneles a nivel de aplicación. TCP sobre TCP

Cuando construimos un túnel desde el nivel de aplicación lo que estamos construyendo es algo similar a lo mostrado en la siguiente figura (6.6). Como se puede comprobar a primera vista, incorporamos dos nuevas capas de red y transporte sobre los niveles TCP/IP originales, lo que, sin lugar a dudas es algo ineficiente.

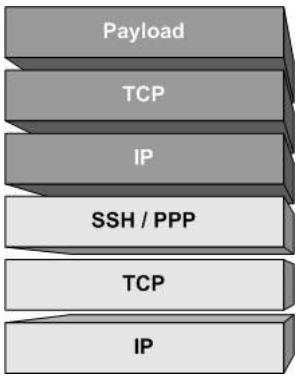


Figura 6.6: TCP sobre TCP.

Podríamos exponer distintas justificaciones de por qué esta configuración no es eficiente, de los problemas que ocasiona el utilizar dos ventanas de retransmisión de paquetes en distintos niveles, y de los riesgos de seguridad que ello implica. Pero la razón más evidente es suficiente para descartar esta solución.

Capítulo 7

Redes de distribución cuántica de claves

Seguimos avanzando desde el punto de vista de la integración, pero ahora miramos nuestro sistema desde otra perspectiva. Queremos estudiar su comportamiento junto a otro entorno de distribución, una red de información. Hasta el momento, hemos trabajado con un sistema de distribución de claves aislado del mundo real, con un canal de comunicación dedicado, una sincronización prácticamente idéntica entre los extremos, y una proximidad mayor de la deseada en un entorno práctico o real. Ha llegado la hora de comprobar la estabilidad esperable de nuestro sistema frente a una carga de trabajo real y utilizando una red de comunicación compartida.

En este capítulo presentaremos, de forma teórica, las distintas estrategias de conexión que se pueden dar en una red de distribución de claves, QKDN[1]. Partiremos de la configuración más sencilla, como es la utilizada hasta el momento, donde dos extremos están conectados directamente por medio de una asociación punto a punto[2], para pasar a describir una extensión de esa primera configuración en una red de distribución. A continuación estudiaremos dos de las topologías más interesantes desde el punto de vista de la distribución de claves, como son las topologías en anillo y estrella. Finalmente, acabaremos con el desarrollo de una estrategia mixta que aproveche las ventajas de cada una de las configuraciones estudiadas. Por otro lado, también estudiaremos las implicaciones de los componentes utilizados actualmente en las redes de fibra óptica, como es el caso de la multiplexación en frecuencias.

Desde un punto de vista práctico, el objetivo principal deberá ser la construcción de una primera red de distribución compuesta por tres nodos. Ese anillo trinodal nos permitirá estudiar, entre otras cosas:

- El rendimiento de los mecanismos de distribución de claves entre nodos no adyacentes. Lo que conoceremos como **transporte** de claves.
- Estrategias de intercambio de claves a través de un nodo compartido, *distribución a tres*.

[1] Quantum Key Distribution Network.
[2] Una red privada virtual.

- Latencias entre cambios de sincronización[3].

Todo esto sin la necesidad de establecer rutas entre los nodos, por lo que quedará pendiente ese punto, el enrutado, para trabajos posteriores.

Un detalle importante a tener en cuenta es que todo el diseño de las redes de QKD ha sido realizado bajo el supuesto de que los equipos utilizados en Alice y Bob son distintos. Por lo tanto, esta configuración puede variar con la implementación de sistemas basados en pares EPR, para los que la configuración de los equipos en Alice y Bob es idéntica.

7.1. Punto a punto. Red privada virtual

Un escenario real, no muy distinto al estudiado en un prototipo de QKD, es el que involucra a dos sedes conectadas por medio de una red privada virtual, VPN[4], a través de una red pública. Esa red privada virtual utiliza una clave de sesión para cifrar la información que ambas sedes comparten a través de la red pública. Y esa clave de sesión es la que nuestros sistemas pueden proporcionar de una forma sencilla y segura. El único requisito para poder implementar este escenario es que los extremos de la comunicación compartan un canal cuántico, o privado, por el que intercambiar las claves de sesión. La imagen 7.1 ilustra este escenario.

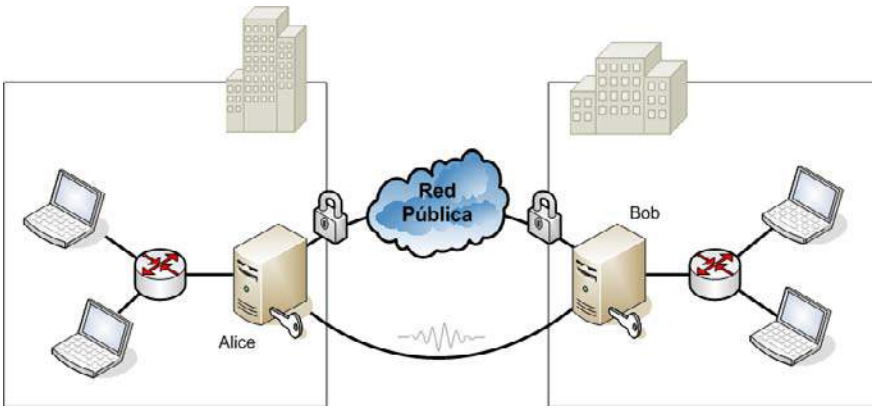


Figura 7.1: Red privada virtual.

Aparentemente, no debemos tener problemas para configurar un escenario como el que acabamos de describir, y parece un buen punto de partida para estudiar el rendimiento inicial de nuestro sistema. Utilizando este escenario, podemos medir el ratio de bits compartidos por los extremos de la comunicación, la tasa de error detectada, y las pérdidas de sincronización del canal privado, entre otras cosas.

Las limitaciones de esta estrategia de conexión aparecen cuando queremos comunicar más de dos sedes, y es lo que vamos a intentar solucionar en el siguiente apartado.

[3]En el desarrollo teórico de este capítulo veremos como el tiempo empleado en el calibrado de una línea tiene una implicación directa en el número de intercambios a realizar.

[4]Virtual Private Network.

7.2. Anillo de distribución

Una vez que hemos comprobado la estabilidad de nuestra red privada virtual, podemos pasar a construir la primera red de distribución cuántica de claves. Comenzamos con un objetivo sencillo como es la configuración de un anillo con tres nodos, que podemos representar como una ampliación directa de la VPN recién descrita. La siguiente figura (7.2) muestra un ejemplo de esta primera aproximación. El anillo está compuesto por tres nodos y dos equipos por nodo, conectados dos a dos a través de una fibra óptica dedicada.

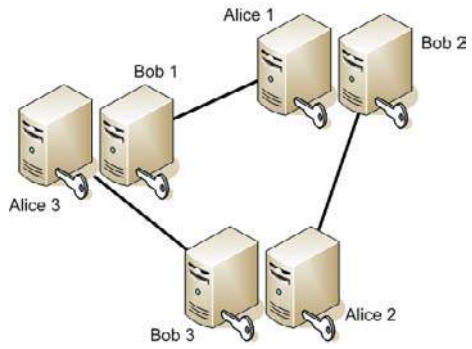


Figura 7.2: Anillo de distribución cuántica de claves (versión realizada a partir de un diagrama compartido de redes privadas virtuales punto a punto entre los nodos del anillo).

La simplicidad de esta primera aproximación la convierte en una solución viable a partir de una configuración básica como es la de una red privada virtual punto a punto. Pero desde el punto de vista práctico su implementación es costosa, puesto que requiere el uso de dos equipos por nodo y una línea de comunicación dedicada entre cada uno de los nodos. Como contrapartida tenemos la ventaja de que el anillo soporta un funcionamiento concurrente, es decir, los tres nodos del anillo pueden estar intercambiando claves al mismo tiempo, lo que es un resultado interesante para la construcción de núcleos de distribución de claves con una elevada carga de trabajo.

Antes de dar el siguiente paso y ampliar el tamaño de nuestra red, debemos detenemos ligeramente para optimizar el resultado de la configuración anterior. Seguimos trabajando con un anillo de tres nodos, pero ahora priorizamos el cumplimiento de dos objetivos:

1. Reducir el número de equipos.
2. Compartir el uso de las líneas de comunicación.

7.3. Configuración en estrella

Para alcanzar el primer objetivo de los descritos en el apartado anterior, partimos de un supuesto que nos obligue a elegir un único equipo en cada nodo de la red. Y dado que la asignación nodo-equipo es anónima, sólo tenemos cuatro posibles configuraciones, que son las mostradas en la tabla 7.1.

Equipo 1	Equipo 2	Equipo 3
Alice	Alice	Alice
Alice	Alice	Bob
Alice	Bob	Bob
Bob	Bob	Bob

Tabla 7.1: Configuraciones disponibles para una red de distribución cuántica de claves con tres nodos, y un único equipo en cada nodo.

De esas cuatro configuraciones, dos no son factibles de implementar, puesto que sólo incluyen equipos de un tipo de extremo. Lo que nos deja tan sólo dos opciones cuyo funcionamiento es aparentemente similar. Tenemos que elegir entre utilizar dos Alice y un Bob, o dos Bob y un Alice. Y la elección debe ser principalmente económica, puesto que, a esta configuración hemos llegado a partir de la decisión de abaratar la configuración original, reduciendo el número de equipos. Y siguiendo este principio de economía, la opción más interesante es la de utilizar dos Alice[5].

El inconveniente de utilizar un sólo equipo por nodo es que, en una red de tres nodos, habrá dos equipos que no podrán intercambiar claves directamente. Por lo tanto, esos equipos deberán compartir claves utilizando la presencia del tercer equipo. Es decir, imaginemos un escenario donde damos nombre y tipo a los nodos de nuestro anillo:

Nodo A	Nodo B	Nodo C
Alice	Bob	Alice

Tabla 7.2: Configuración específica.

En esta configuración, los nodos A y B pueden intercambiar claves directamente, al igual que los nodos B y C. Pero el nodo A no puede intercambiar una clave directa con el nodo C, puesto que toda comunicación debe involucrar la utilización de dos extremos distintos, un Alice y un Bob. Ahora bien, que no puedan intercambiar una clave de forma directa no quiere decir que no puedan compartir una clave. Ambos equipos pueden utilizar al tercer nodo, B, como intermediario. Entonces, los nodos A y C intercambiar una clave con B de forma simultanea, y a continuación comparten una clave directa, de la forma: $\kappa_{AB} \oplus \kappa_{BC}$, donde κ_{AB} es la clave compartida por los nodos A y B, y κ_{AB} es la clave que comparten B y C.

[5]Actualmente, la parte más delicada y costosa de los equipos que estamos utilizando son los detectores de fotones. Estos se encuentran sólo dentro de los equipos receptores, es decir, dentro de Bob, por lo que en principio, estos son los equipos más caros en la actualidad. También debemos tener en cuenta el tipo de estrategia que utilizan Alice y Bob para comunicarse, puesto que en una estrategia de doble dirección, o plug & play, Bob contiene también el emisor de fotones, así como los dos caminos del interferómetro del Mach-Zehnder. Con esta estrategia, Alice sólo necesita un atenuador y un espejo de Faraday, con lo que el coste final del equipo es considerablemente más económico que el de Bob.

7.4. Canal compartido

Seguramente, el principal inconveniente de los sistemas de distribución segura de claves reside en la necesidad de disponer de un canal de comunicación dedicado. Esto es costoso, y en consecuencia inviable en la mayoría de los casos, por lo que debemos buscar una solución a la integración de estos sistemas dentro de la soluciones actuales de comunicación.

A continuación, veremos dos estrategias que utilizaremos de forma conjunta:

- Conmutadores ópticos.
- Modulación.

7.4.1. Conmutadores ópticos

El segundo objetivo es algo más complicado de lograr debido a que la implementación de un sistema de distribución cuántica de claves requiere de un canal de comunicación dedicado con un comportamiento pasivo[6]. Lo que nos obliga a utilizar redes con un comportamiento especial y con un componente nuevo que se comporte a modo de conmutador óptico, para el establecimiento dinámico de rutas dedicadas.

La siguiente figura (7.3) muestra los cambios sugeridos con el fin de abaratar el diseño de nuestra red.

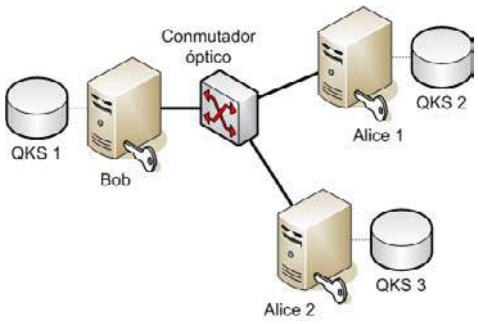


Figura 7.3: Anillo de distribución cuántica de claves (versión compartida).

El conmutador óptico-pasivo será una pieza clave para la construcción de las redes de distribución cuántica de claves. Su utilización es imprescindible para establecer un canal dedicado a través de una sucesión de líneas compartidas. Y su gestión debe realizarse desde el nivel de enlace de la arquitectura de nuestro sistema, justo antes de comenzar el intercambio de una clave.

[6]Un canal de comunicación se considera pasivo cuando éste no incluye elementos de conexión que puedan alterar los qubits transmitidos por el canal. En nuestro caso, el canal de comunicación es la fibra óptica, y algunos de los elementos que pueden alterar la transmisión de nuestros pulsos atenuados son: amplificadores, switches y moduladores en tiempo.

7.4.2. Multiplexación en frecuencia. WDM[7]

La primera sugerencia puede ser una derivación de la multiplexación en el tiempo. Siguiendo esta estrategia, podemos dividir el uso de las líneas en periodos de tiempo, entre los cuales, los correspondientes a la distribución cuántica de claves se utilizarán de forma exclusiva. Esta distribución en el tiempo no es novedosa, y ya se ha utilizado en canales de fibra antiguos, pero hoy día se utiliza otra estrategia parecida, multiplexación en frecuencia, que permite la división de una fibra en varios canales de comunicación, incrementando el ancho de banda disponible y en consecuencia el rendimiento final de la fibra. Esta estrategia es muy común hoy día, y para su implementación se utilizan los modems WDM.

En función del número de canales que obtenemos de la división en frecuencia encontramos dos tecnologías:

- **CWDM**[8]. Emplea una separación de no menos de 20 nm entre canales, utilizando principalmente las longitudes de onda disponibles entre los 1270 nm y los 1610 nm (con una desviación aproximada de $\pm 6 - 7nm$ sobre el valor central de cada canal). Esta especificación nos permitiría disponer de hasta 18 canales, aunque en la práctica los sistemas comerciales no facilitan más de 15 canales [18].

- **DWDM**[9]. Los sistemas DWDM actuales utilizan canales con una separación de 100 GHz o 50 GHz, consiguiendo hasta un máximo de 150 o 300 canales respectivamente [10] (en el Apéndice de este proyecto se han incluido varias tablas con las frecuencias por canal recomendadas para equipos DWDM con una separación por canal de 100 GHz y de 50 GHz, en las bandas L, C y S).

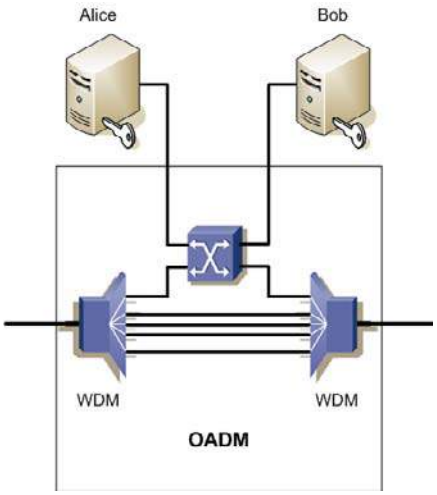


Figura 7.4: Nodo con multiplexación en frecuencia.

[7]Wavelength Division Multiplexing
[8]Coarse Wavelength Division Multiplexing.
[9]Dense Wavelength Division Multiplexing.

La figura (7.4) muestra un ejemplo de la posible utilización de la tecnología de modulación en frecuencia. Está configuración corresponde a los moduladores OADM[10], que se utilizan en las configuraciones de fibra óptica en anillo. Cada vez que la fibra llega hasta un nodo, un multiplexador DWDM extrae los distintos canales de la fibra y desvía sólo aquellos correspondientes al nodo en cuestión. A continuación, vuelve a multiplexar todos los canales, para continuar el trayecto hasta el siguiente nodo.

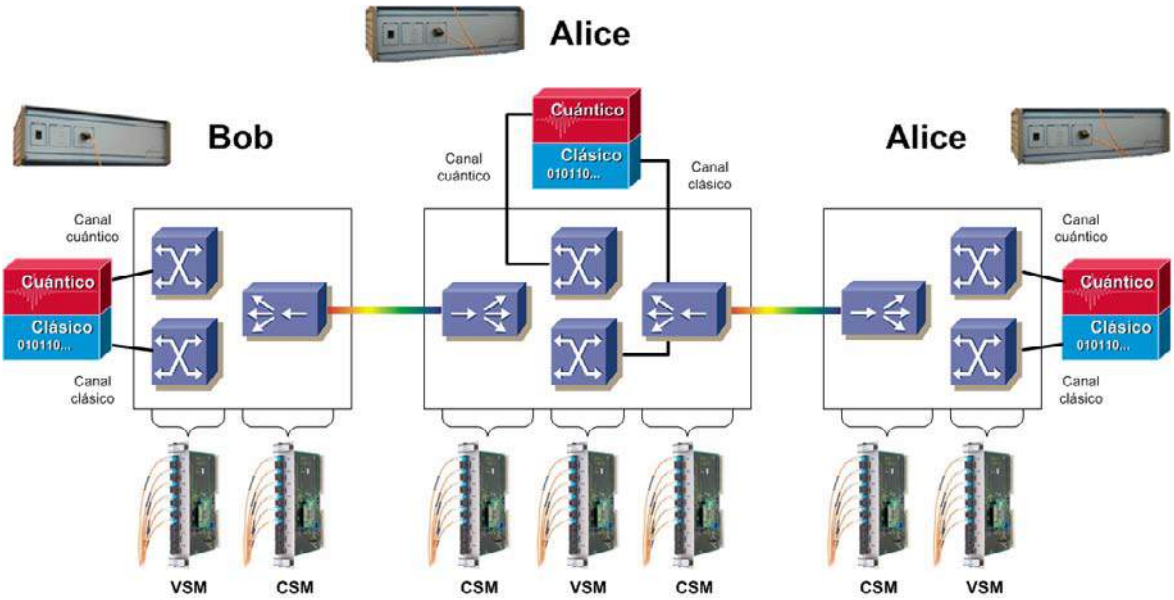


Figura 7.5: Esquema ROADM.

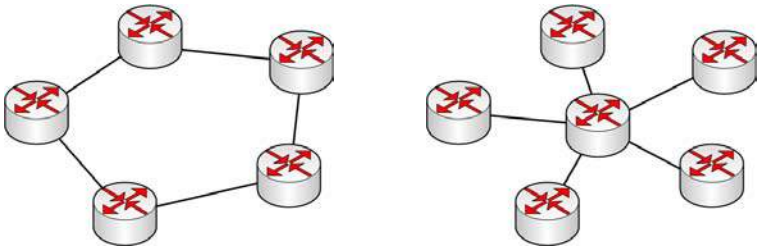
Otra configuración admisible utilizando mecanismos de multiplexación en frecuencia es el que observamos en la figura 7.5, donde aparecen tres nodos conectados de forma secuencial a través de dos fibras independientes. En cada nodo conectamos un equipo QKD (individual) con un único Bob situado en uno de los extremos, y dos Alice en el resto de nodos. Puesto que el intercambio de claves sólo puede realizarse entre equipos distintos, los dos Alice deberán conectarse con Bob compartiendo la misma fibra que une a Bob con el Alice central. Una forma de compartir ese canal de comunicación es utilizar frecuencias distintas para la emisión de fotones, y gestionar en cada nodo las frecuencias correspondientes al mismo como muestra el diagrama ROADM[11] indicado.

7.5. Topologías en QKDN

Uno de los grandes inconvenientes de la distribución cuántica de claves es la distancia. Como vimos en el capítulo donde estudiamos su arquitectura, la eficiencia de este tipo de sistemas disminuye de forma exponencial con la separación de los extremos de la comunicación. Por esta razón, es más que justificado el uso de nodos intermedios para el intercambio de claves a través de largas

[10] *Optical Add-Drop Multiplexer.*
[11] *Reconfigurable Optical Add-Drop Multiplexer.*

distancias, y lo que veremos a continuación es la forma de conectar esos nodos intermedios con el objetivo de obtener el mayor rendimiento final de los equipos utilizados.



(a) Anillo (b) Estrella

Figura 7.6: Topologías.

Repasando lo que hemos visto hasta el momento en este capítulo, descubrimos que hemos utilizado la configuración de dos topologías clásicas (figura 7.6), particularmente interesantes en redes de distribución cuántica de claves, pero con una interpretación distinta a la estudiada en redes convencionales. Veamos cual es esa interpretación.

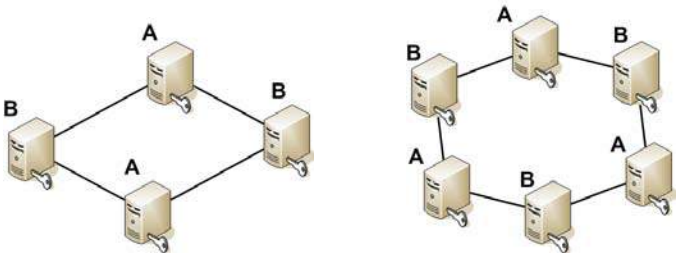
7.5.1. Topología en anillo

Para comprender el funcionamiento de las redes "circulares" de distribución cuántica de claves, o en forma de anillo, tenemos que volver al inicio del capítulo, donde escalamos una red privada virtual de dos extremos a un anillo de tres nodos como muestra la figura (7.2) anterior. La dimensión del anillo, con tan sólo 3 nodos, no es lo suficientemente grande como para comprender el modo de funcionamiento de esta topología, pero en ese diagrama sí podemos identificar la versatilidad de esta configuración. Utilizando dos equipos por nodo[12], como muestra la figura, conseguimos que todos los canales de la red estén intercambiando claves de forma continua, con el consiguiente beneficio en eficiencia.

Nodos pares

Existe un caso especial en el que no es necesario duplicar los equipos por nodo. Cuando el anillo está formado por un número par de nodos, podemos diseñar una configuración donde los equipos de distinto tipo están intercalados, de tal forma que cualquier nodo del anillo puede comunicarse con sus adyacentes. En la figura siguiente (7.7) vemos dos ejemplos de estas configuraciones.

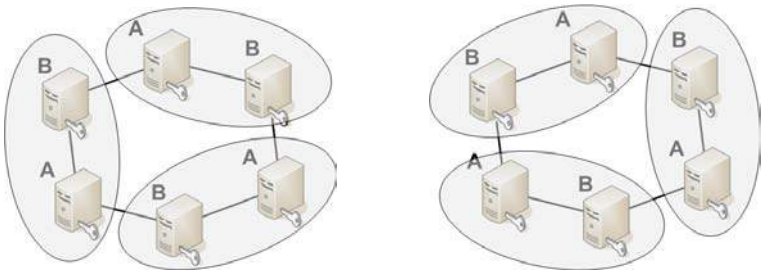
[12]La tecnología actual para la implementación de un sistema QKD requiere el uso de dos equipos distintos en cada extremo de la comunicación. Este hecho tiene visos de ser duradero en el tiempo, debido a que la funcionalidad de ambos equipos es distinta. En consecuencia, los componentes de cada uno de los equipos también difieren, lo que provoca que trabajemos con extremos que no comparten los rasgos suficientes como para que puedan ser compatibles, es decir, los extremos de un sistema QKD no pueden intercambiar sus roles: un emisor puede estar conectado a uno o varios receptores, pero nunca a otro emisor (y viceversa, un receptor puede estar conectado a uno o varios emisores, pero no a otro receptor).



(a) 4 nodos (b) 6 nodos

Figura 7.7: Anillos con un número par de nodos.

Debemos asumir una limitación con esta modificación, y es que un nodo no puede estar comunicándose con sus nodos vecinos de forma simultanea. Una limitación exclusiva del nivel cuántico, puesto que el resto de capas de la arquitectura pueden coincidir en cualquier momento de la ejecución. Podemos paliar parte de esta limitación si sincronizamos los intercambios entre todos los equipos. Al trabajar bajo el supuesto de que el anillo está formado por un número par de nodos, nunca encontraremos nodos desocupados, consiguiendo así un rendimiento máximo de los equipos.



(a) Pares (b) Impares

Figura 7.8: Intercambios sincronizados en tiempos pares e impares.

Los inconvenientes de esta estrategia son evidentes:

1. En primer lugar, el cambio en la secuencia de intercambio conlleva a un proceso de medida del canal cuántico, para sincronizar las líneas que unen cada nueva pareja. Esa sincronización de la línea es costosa, en tiempo, lo que implica una ineficiencia directa sobre la sincronización global de todas las parejas. La única forma de paliar esta contrariedad es incrementando el número de intercambios realizados por cada pareja antes de saltar a la siguiente configuración.

2. Pero el principal inconveniente es la sincronización. La longitud de los canales que unen cada par de nodos no tiene porque ser idéntica, lo que puede degenerar en una falta de sincronización entre las parejas del anillo. Esa pérdida de sincronización será causada por las siguientes razones:

   a) Una longitud de línea distinta provoca que el tiempo empleado por cada pulso para recorrer el camino que une a los extremos del sistema

sea distinto. Además, en la actualidad el tiempo que tarda un pulso en recorrer el canal de comunicación es despreciable con respecto a la frecuencia de envío, lo que podría llevarnos a pensar que el tiempo empleado para completar el intercambio de fotones a nivel físico es prácticamente idéntico aunque la distancia sea superior. Pero esto no es cierto en su totalidad ya que, existen configuraciones de los sistemas QKD en las cuales el tiempo que tarda un fotón en recorrer el canal de comunicación no va a ser tan despreciable.

La siguiente expresión nos permite calcular una aproximación del tiempo de intercambio, t_{raw}, en un sistema QKD donde no se producen interrupciones en el envío de fotones:

$$t_{raw} \simeq \frac{\ell_{line}}{c_{fiber}} + n \cdot T$$

Donde ℓ_{line} es la longitud de la línea, c_{fiber} es la velocidad de la luz en la fibra óptica, $c_{fiber} = 2 \times 10^8 m/s$, n es el número de pulsos intercambiados y T el periodo o tiempo que separa a dos pulsos consecutivos, en nuestro caso: $T \simeq 200ns$.

Por otro lado, como ocurre en los sistemas QKD de doble dirección, el intercambio de fotones se agrupa en trenes que son enviados de forma independiente, lo que provoca una interrupción del envío de pulsos hasta que se completa la recepción del último tren. La expresión para calcular el tiempo empleado en el intercambio de fotones a nivel físico, t_{raw}, es la siguiente:

$$t_{raw} \simeq N \cdot \left(\frac{\ell_{line}}{c_{fiber}} + n \cdot T \right)$$

Donde ahora n es el número de pulsos por tren y N es el número de trenes intercambiados.

Podemos ver con un ejemplo práctico como se varía el tiempo de intercambio de fotones a nivel físico en el peor de los casos, es decir, en los sistemas de doble dirección. En el cálculo realizado hemos empleado unos parámetros similares a los utilizados por el sistema id-3000 de id Quantique: con 500 pulsos por tren y un total de 2000 trenes para completar el envío de un millón de fotones, un periodo de emisión de pulsos de 200 ns, y dos longitudes de canal de comunicación distintas, de $12{,}6$ y $25{,}2$ km.

$$t_{raw,1} = 2000 \cdot \left(\frac{12{,}6 \times 10^3}{2 \times 10^8} + 500 \cdot 200 \times 10^{-9} \right) \simeq 0{,}326s$$

$$t_{raw,2} = 2000 \cdot \left(\frac{25{,}2 \times 10^3}{2 \times 10^8} + 500 \cdot 200 \times 10^{-9} \right) \simeq 0{,}452s$$

Como muestran los resultados, el incremento en el segundo de los resultados es cercano al 40 % con respecto al tiempo calculado para el primer caso. Luego podemos afirmar que en los sistemas de doble

dirección una variación de la distancia va a tener un impacto notable sobre la sincronización de los nodos de un anillo.

b) Ahora bien, cuando la distancia aumenta también aumentan las pérdidas de la línea, lo que implica que el número de detecciones final será inferior al de un trayecto más corto. Esto provocará un aumento directo de la tasa de error, lo que se traduce en otro incremento del proceso de destilación de una clave.

En consecuencia, un aumento de la longitud de la línea provoca un incremento en el tiempo de destilación de la clave, causado por un aumento del QBER. Luego la pérdida de sincronización se refleja principalmente en el proceso clásico, y no en el intercambio de nivel físico.

Escalabilidad

Cualquier limitación es siempre una barrera a la escalabilidad de un sistema, y nuestro caso no es una excepción. La obligación de utilizar un número par de nodos supone una imposición estructural que no podemos corregir, aunque esto no quiere decir que no podamos construir esta configuración con un número impar de nodos. Existe una alternativa costosa e ineficiente, que es la utilización de dos equipos en uno de los nodos, de tal forma que se siga manteniendo la paridad entre todos los equipos conectados.

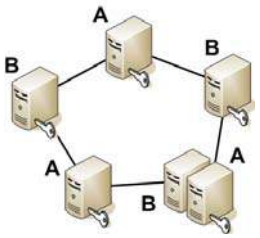


Figura 7.9: Configuración de un anillo óptimo con 5 nodos.

7.5.2. Topología en estrella

La otra configuración topológica que nos vamos a encontrar en una red de distribución cuántica de claves es la de estrella. La ventajas de esta estrategia con respecto a la configuración en anillo son evidentes, y es que:

- En todo momento podemos conectar un nuevo nodo a la red utilizando un equipo complementario al del nodo central, solucionando así los problemas de escalabilidad de la solución anterior. Esta ventaja tiene asociado un inconveniente debido a que el ratio global en claves por usuario del nodo central se mantiene siempre constante, en consecuencia, cuantos más nodos haya en nuestro sistema, menor será el número de claves por nodo.

- En el supuesto de que exista una diferencia considerable en precio de

los extremos del sistema QKD[13], podemos adoptar una configuración que adopte la mayor rentabilidad económica posible, utilizando el extremo más caro como nodo central, y conectando a su alrededor los equipos de menor precio.

Centralización

El resultado final de esta topología es una estrategia centralizada, con las ventajas e inconvenientes que esto conlleva. A la ventaja económica que acabamos de ver, tenemos que añadir la facultad de control que adquiere el nodo central, que será de especial interés en la gestión de los registros de certificación que veremos en el capítulo siguiente. Evidentemente, esa dependencia del nodo central es también un riesgo estructural, ya que la red depende al completo de su correcto funcionamiento.

Pero el principal inconveniente que se nos presenta es la pérdida de eficiencia. Como muestra la figura 7.10, sólo uno de los extremos de la red puede estar conectado con el nodo central[14], y en consecuencia, el resto de los equipos debe esperar a que el nodo central esté disponible para realizar un intercambio. Una vez más, esta limitación es sólo en el nivel cuántico, ya que el resto de niveles se pueden ejecutar de forma asíncrona, y por lo tanto simultanea. Si tenemos en cuenta que la principal limitación de un intercambio de claves se encuentra en el proceso de destilación clásico, podemos pensar en esta estrategia como una alternativa aceptable, ya que se puede mantener un intercambio "lógico" de claves de forma simultanea en todos los nodos de la red.

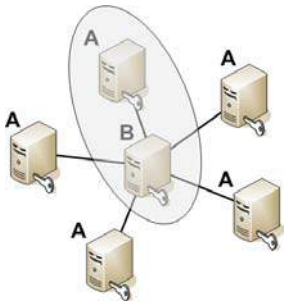


Figura 7.10: Intercambios en una topología en estrella.

Eso sí, las limitación en la sincronización de los canales vuelve a ser clave. Cada reconexión del nodo central con un extremo distinto de la red, requiere de un proceso de medida y sincronización del canal de comunicación cuántico. Ese proceso, como ya hemos comentado en innumerables casos, es costoso en cuanto a tiempo. Por lo tanto, debemos buscar un punto de equilibrio en el número de intercambios a nivel físico que se realizan durante cada conexión, de tal forma que:

[13]Este escenario será habitual cuando los equipos asignados a cada nodo se conectan utilizando una estrategia de doble dirección.

[14]Recordemos que es una limitación estructural de los sistemas QKD.

- El tiempo empleado en la sincronización no sea un factor decisivo en el rendimiento final de los intercambios.
- Al mismo tiempo que la latencia entre cada ciclo de intercambios no supere el tiempo de destilación de las claves intercambiadas en cada nodo. En cierto modo, una forma de limitar o reducir la latencia de una red en estrella es reduciendo el número de brazos de esa red, es decir, el número de nodos que se conectan al nodo central.

Podemos calcular los tiempos de intercambio por nodo, t_{node}, y el tiempo de latencia[15] entre los intercambios, t_{lat}, de la siguiente forma:

$$t_{node} = t_{sync} + n \cdot t_{exch}$$

$$t_{lat} = N \cdot t_{node}$$

Donde t_{sync} es el tiempo de sincronización de la línea, n es el número de intercambios realizados, t_{exch} es el tiempo de intercambio de cada clave, y N es el número de nodos. Además, sabemos que el tiempo de destilación de una clave a nivel clásico es proporcional al tiempo de intercambio a nivel físico en función de los parámetros de caracterización y eficiencia del sistema QKD empleado:

$$t_{dist} = F_{dist} \cdot K_{raw}$$

$$K_{raw} = p_{det} \cdot f \cdot t_{exch}$$

Donde t_{dist} es el tiempo de destilación de una clave, F_{dist} es el factor de destilación de la clave, K_{raw} es el tamaño de la clave intercambiada, p_{det} es la probabilidad de detección, y f es la frecuencia de emisión de los pulsos intercambiados.

Finalmente, podemos acotar el tiempo de latencia en función del tiempo empleado para la destilación de todas las claves de un nodo.

$$t_{lat} \leq n \cdot t_{dist}$$

7.5.3. Topologías híbridas

Como siempre, la mejor forma de sustentar un equilibrio entre las distintas topologías disponibles es utilizar una estrategia híbrida, como muestra la siguiente figura 7.11.

Existirán situaciones en las que requerimos un nivel de eficiencia superior, o con una carga de trabajo elevada, para las que sería interesante utilizar una configuración en anillo, con uno o dos equipos por nodo, según se de el caso. Mientras que podemos ampliar ese anillo con extensiones en estrella, considerablemente más económicas, sin dependencias estructurales, y fácilmente adaptables e interesantes para una futura gestión centralizada de las claves intercambiadas.

[15]Tiempo empleado para el intercambio de claves en todos los nodos de la red.

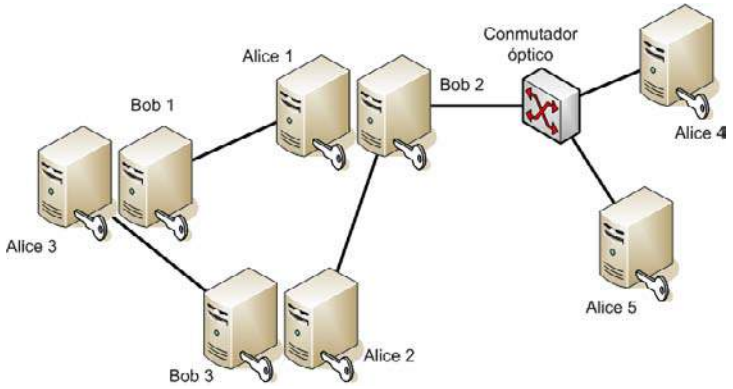


Figura 7.11: Topología híbrida. Redes en anillo y estrella.

7.6. Intercambio a 3 bandas

Imaginemos una situación como muestra la siguiente figura (7.12), donde intervienen tres interlocutores, Alice, Bob y Charlie. Debido a la distancia entre Alice y Charlie, o por alguna otra razón, no existe un canal privado que comunique directamente a ambos interlocutores. Pero sí están unidos a por medio de un canal privado con un nodo adyacente compartido, Bob.

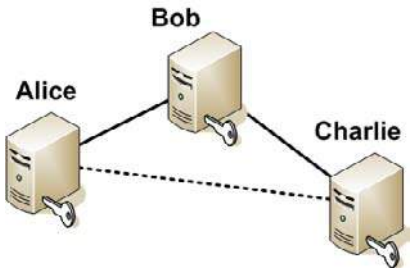


Figura 7.12: Distribución de claves a 3 bandas.

Parece evidente que la forma más sencilla para compartir una clave entre Alice y Charlie es que ambos realicen antes un intercambio con Bob. Pero vamos a ver, que existe otra estrategia que permite compartir una clave entre Alice y Charlie con tan sólo una intervención parcial de Bob. En el siguiente algoritmo (9) definimos esa estrategia.

En primer lugar, debemos tener presente que esta estrategia reduce el número de destilaciones realizadas, pero esto es efectivo sólo en el caso de que las destilaciones no se puedan ejecutar de forma concurrente. Es decir, si Alice y Bob pueden realizar el proceso de destilación al mismo tiempo que Bob y Charlie realizan la destilación de otra clave, este procedimiento no aporta ninguna mejora al transporte *one-time-pad*. La razón por la que proponemos esta estrategia es que puede ser interesante de cara al transporte de una clave a lo largo de distancias importantes, cuando la carga de información a través del canal público es muy elevada, e intentamos reducirla de alguna manera.

Otros inconvenientes de esta estrategia son:

Algoritmo 9 Intercambio a 3 bandas.

1. Alice intercambia una clave en bruto con Bob.
2. Bob intercambia con Charlie la clave en bruto recibida de Alice. Este intercambio se completa sin realizar ningún proceso de reconciliación de bases.
3. Bob envía a Alice las bases utilizadas para la decodificación de la clave intercambiada.
4. Charlie envía a Alice las bases utilizadas para la decodificación de la clave intercambiada con Bob.
5. Alice responde a Charlie con las posiciones donde la bases elegidas por Bob y Charlie han sido correctas.
6. Alice y Charlie destilan la clave intercambiada.

- Bob conoce la clave intercambiada por Alice y Charlie. Escuchando la información compartida por Alice y Charlie a través del canal público, Bob puede destilar una clave idéntica a la compartida por ambos extremos. Esto puede ser un riesgo se seguridad, y obliga a que Bob se encuentre autenticado.
- Otro inconveniente de esta estrategia es que Bob debe poder comportarse como un Alice de cara a la comunicación entre Bob y Charlie. Lo cual sólo ocurrirá en configuraciones de anillos donde cada nodo contiene los dos equipos, Alice y Bob, o configuraciones como la mostrada a continuación (7.13).
- Además, en las estrategias donde el número de detecciones es considerablemente inferior al número de pulsos intercambiados, la eficiencia del protocolo se ve reducida de forma exponencial. Esta eficiencia se puede mejorar incrementando el número de intercambios a nivel físico, siempre que el tiempo empleado en esos intercambios sea inferior al tiempo necesario para la destilación de dos claves.
- También nos encontramos con la propagación de los errores a través de cada comunicación. Luego la tasa de error en Charlie será el doble que la esperada por Bob.
- Finalmente, hay que autenticar dos transmisiones, lo que vuelve a disminuir el ratio de clave final.

Para poder determinar si esta estrategia es efectiva, es decir, si reducimos el tiempo total empleado en el intercambio de una clave entre Alice y Charlie, necesitamos conocer: el tiempo de intercambio de una clave en bruto entre dos extremos, t_{raw}, el número de intercambios que necesitamos hacer para obtener una clave en Charlie, n_{raw}, y el tiempo empleado en el proceso de destilación, t_{dist}.

El primero de los parámetros que necesitamos, el tiempo de intercambio de una clave en bruto, lo podemos calcular a partir el tiempo que emplea cada pulso en recorrer la fibra, t_{line}, el número de pulsos emitidos, n, y la frecuencia con la que se emiten dichos pulsos, f[16].

$$t_{raw} = t_{line} + \frac{n}{f} = \frac{\ell_{line}}{c_{fiber}} + \frac{n}{f}$$

Si realizamos este cálculo para los sistemas id-3000 que estamos utilizando, con el intercambio de un 1 GB de pulsos, bajo una frecuencia de emisión de 5 MHz, y una distancia de 10 Km. El tiempo que obtenemos para el intercambio de una clave es el mostrado a continuación:

$$t_{raw} = 2 \cdot \frac{10Km}{2 \times 10^8 Km/s} + \frac{10^6}{5MHz} = 10^{-7}s + 0{,}2s \approx 0{,}2s$$

Como vemos, la distancia que separa a ambos extremos de la comunicación no es significativa de cara al tiempo empleado en cada intercambio.

Ahora bien, el tamaño de la clave obtenida en Bob es inferior al número de pulsos emitidos, dependiendo de la probabilidad de detección final del sistema. Por lo tanto, puesto que queremos que Bob reenvíe una secuencia de valores similar en tamaño a la enviada por Alice, tenemos que realizar un número determinado de intercambios, n_{raw}, que dependerá de la probabilidad de detección, p_{det}.

$$n_{raw} = \frac{1}{p_{det}}$$

En nuestro caso, puesto que la probabilidad de detección es del 1.2 %, $n_{raw} = 1/0{,}012 \approx 83{,}3$, el número de intercambios que debemos realizar para obtener un mega de clave en Bob es tan elevado, que el tiempo necesario para realizar dichos intercambios supera los 15 segundos.

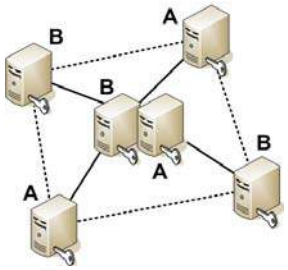


Figura 7.13: Anillo con nodos mixtos.

Como vemos en la figura 7.13, la topología de la conexión de los equipos es en estrella, pero los intercambios se pueden realizar en forma de anillo, ya que, utilizando la estrategia que acabamos de describir, todos los extremos de la red pueden intercambiar claves con sus adyacentes.

[16]Para la medición de este tiempo de intercambio no hemos tenido en cuenta otros factores externos a los sistemas QKD, como es tiempo necesario para generar una secuencia aleatoria de valores, o el tiempo empleado en transmitir esa secuencia desde un ordenador hasta el dispositivo físico encargado de codificarla.

7.7. Niveles de una red QKD

Para completar este capítulo, hemos querido mostrar una representación del acoplamiento de una red QKD a distintos niveles:

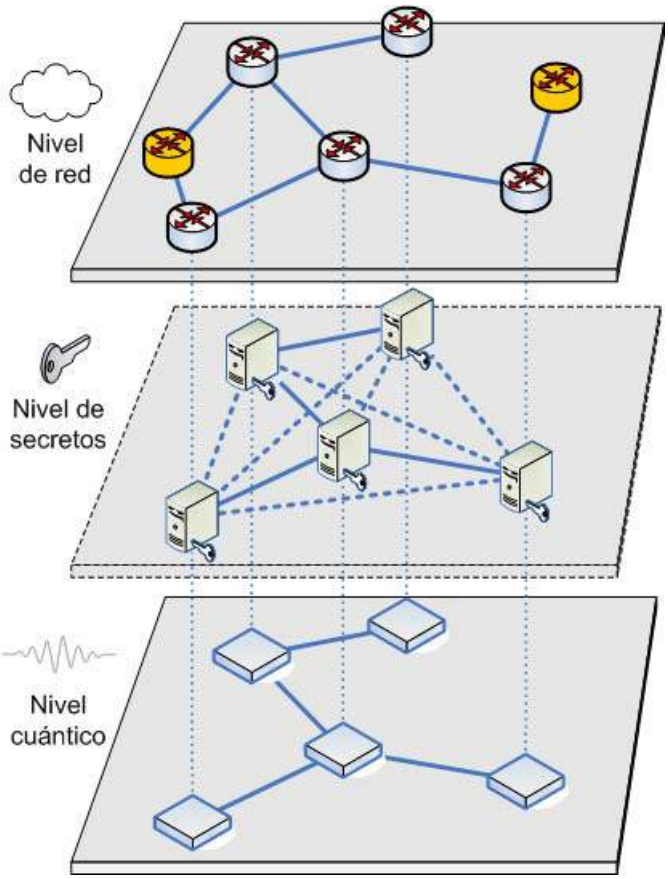


Figura 7.14: Niveles de una red con distribución cuántica de claves.

1. **Nivel cuántico**. El nivel inferior de cualquier red de distribución cuántica de claves es la capa de enlaces cuánticos para la distribución segura de claves. Tanto si son compartidos como si no los enlaces del nivel cuántico, la red resultante será como mucho igual a la red de conexión convencional. Recordemos que es imprescindible la disposición de un canal de comunicación público.

   En principio, los enlaces utilizados a este nivel deben ser dedicados, pero como ya hemos visto, podemos compartir estos canales con la red de distribución convencional.

2. **Nivel de secretos**. El siguiente nivel que obtenemos en nuestra red QKD, es el nivel de secretos. Este nivel no es más que una extensión del nivel cuántico. Extensión que obtenemos de la interconexión, dos a dos, de todos los nodos de la red. A partir de esta definición, podemos obviar la existencia de este nivel, pero es importante tener en cuenta las implicaciones

de cada conexión indirecta. Sabemos que si dos nodos comparten una línea de comunicación dedicada, ambos extremos pueden intercambiar una clave segura. También sabemos que dos extremos que no disponen de una línea dedicada pueden compartir una clave a partir de un tercer nodo, con el coste asociado de la necesidad de compartir inicialmente dos claves, una por cada camino del nivel cuántico.

Este nivel de secretos nos hace ver el nivel anterior como un camino de claves, que se van consumiendo a lo largo de la trayectoria que une dos equipos que intentan compartir una clave.

3. **Nivel de red**. Finalmente, llegamos al nivel clásico de cualquier red. En este nivel se muestran los nodos y sus conexiones, sean o no compartidas por el nivel cuántico. Por lo tanto, en este nivel descubriremos nodos que no poseen una identificación dentro del nivel cuántico, y caminos que tampoco aparecen en dicho nivel.

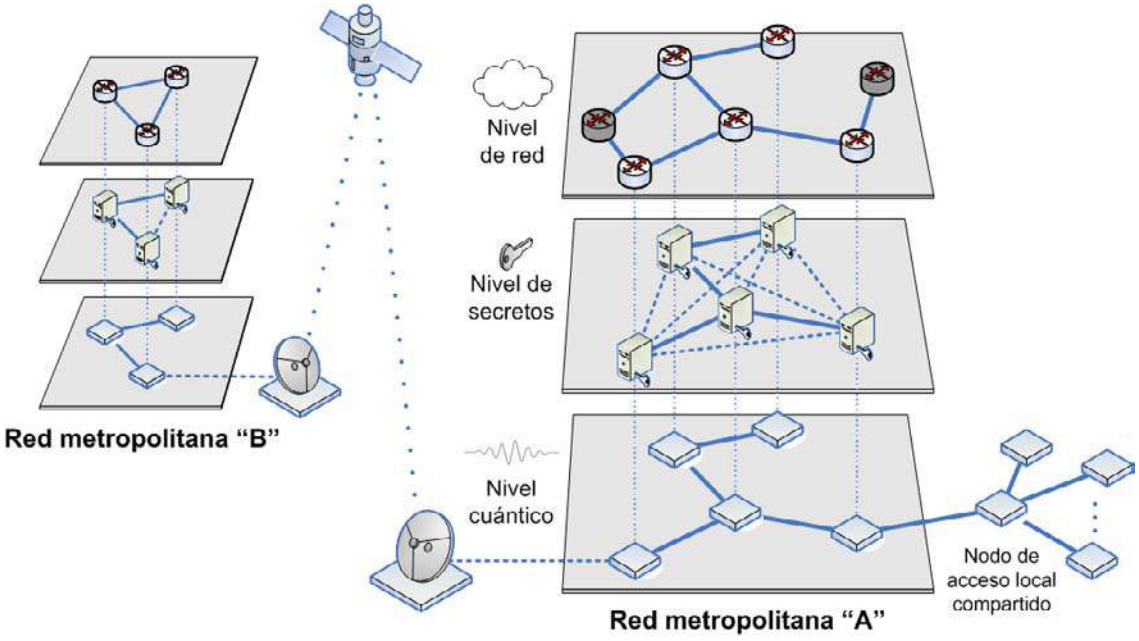


Figura 7.15: Niveles de una red con distribución cuántica de claves (versión ampliada).

7.8. Componentes en una red QKD funcional

7.8.1. Red principal. Backbone

Nos planteamos un objetivo específico, nada descabellado aunque si futurista, como puede ser proporcionar una red de distribución cuántica de claves que de servicio a una ciudad completa, por ejemplo: Madrid. La razón principal por la que reducimos el alcance de nuestro objetivo a una única ciudad es la distancia. Sabemos que la separación entre los extremos de una comunicación es una

de las restricciones más importantes de los sistemas QKD, por lo que evitamos el planteamiento de un problema donde esa limitación sea un cuello de botella. Aún así, en una ciudad como Madrid bajo esta restricción (la distancia) va a ser imposible conectar todos los puntos de la ciudad de forma directa.

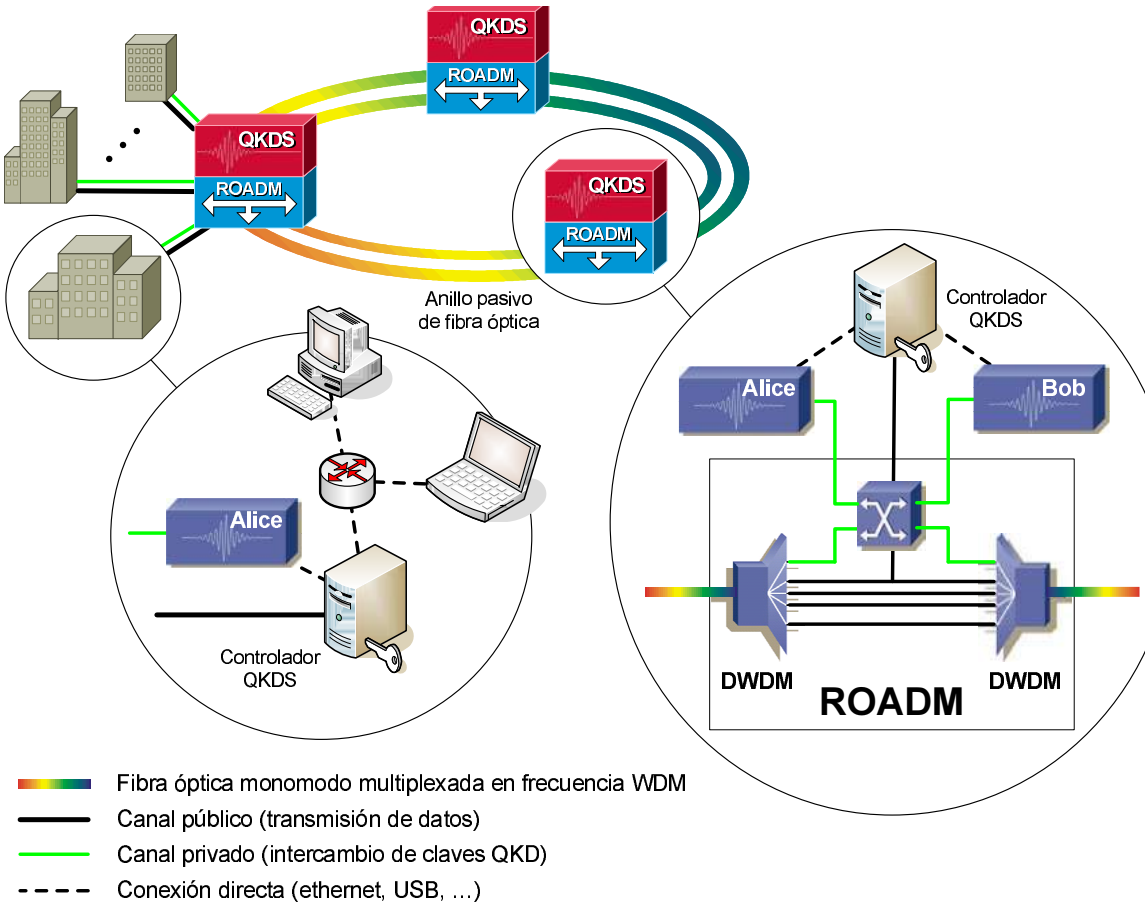


Figura 7.16: Instalación de un anillo QKD en el área metropolitana de Madrid.

Parece inevitable la necesidad de diseñar un entramado de nodos que permita la máxima difusión de claves entre todos los usuarios conectados a la red. Todo lo visto hasta el momento nos lleva a pensar en una arquitectura mixta, donde subredes en forma anillo y otras en forma de estrella se comunican para formar un entramado de nodos con acceso a claves cuánticas. Si intentamos plantear el escenario básico donde dos redes, una en anillo y otra en estrella, se conectan, llegamos a un esquema como el presentado en la figura 7.16. Debido a que las redes en anillo tienen un mejor rendimiento frente a la carga de trabajo, a pesar de tener también un mayor coste, parece evidente que esta topología formará lo que conocemos como *backbone*, la columna vertebral de nuestra red QKD. De igual forma, las topologías en anillo aparecerán en lo que denominamos redes de acceso (preferiblemente de menor coste y con exigencias de rendimiento inferiores). En consecuencia, la figura 7.16 parece un fiel reflejo de lo que será

una red metropolitana.

7.8.2. Redes de acceso

Tecnologia xPON

En los últimos años las empresas operadoras de telecomunicaciones están implantando un nuevo tipo de redes conocidas bajo el término PON[17]. Como su nombre indica se trata de redes basadas en fibra óptica con un comportamiento pasivo, es decir, no modifican la señal que se transmite a través de la red, y por lo tanto: no amplifican la señal, ni transforman la misma (como ocurre con los conversores electro-ópticos que encontramos frecuentemente en los equipos terminales de redes ópticas comerciales). Son redes punto-a-multipunto donde por medio de un "divisor" una única fibra óptica se divide para llegar a distintos nodos (en la práctica no más de 32). Los componentes que encontramos en este tipo de redes son tres:

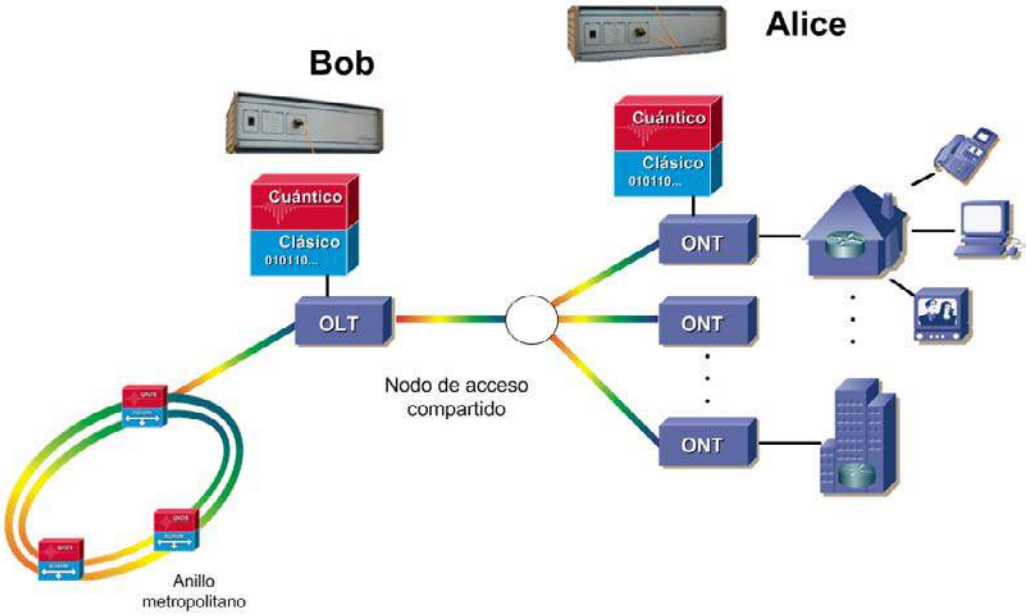


Figura 7.17: Esquema simple de una red PON.

- **OLT**[18]. Es el extremo de la red que actúa como nodo central para dar servicio al resto de nodos (los extremos situados después de la división de la fibra). Sólo existirá un nodo OLT en la red.
- **ONT**[19] (o ONU[20]). Es el equipo que se sitúa en el extremo de la red que queda detrás del divisor, o splitter. En una red PON entontraremos trantas ONTs como divisiones en la fibra.

[17] *Passive Optical Network.*
[18] *Optical Line Termination.*
[19] *Optical Network Terminal.*
[20] *Optical Network Unit.*

- *Splitter*, o divisor óptico. Es el divisor de la fibra.

Por sus características, las redes PON son un buen mecanismo para llevar la fibra óptica hasta el usuario final, y matener múltiples nodos conectados con la menor cantidad de fibra necesaria. Son redes de corto alcance, diseñadas para "el último kilómetro", por lo que deben combinarse con otras estrategias de comunicación para la construcción de redes de área metropolitana. La figura 7.17 muestra un ejemplo de red PON donde aparecen los tres componentes comentados: OLT, ONT y splitter; además de los sistemas QKD necesarios para construir lo que denominamos como **red de acceso compartido**. La integración de los sistemas QKD en redes PON es un avance aparentemente prometedor para el desarrollo de ambas tecnologías, y es que tanto los sistemas QKD como las redes PON comparten dos requisitos de trabajo fundamentales: redes pasivas y de corto alcance. Pero además, la topología de una red PON como la descrita es ideal para un tipo muy concreto de sistemas QKD, los de doble dirección (o *plug and play*), donde un único Bob situado en la OLT es compartido por varios Alice, uno por cada ONT.

Como se observa en las figuras 7.17 y 7.18 todas las ONTs deben compartir el uso una única fibra que une el divisor óptico con la OLT. La forma habitual en la que una red PON comparte en canal de comunicación que une las ONTs con la OLT es mediante multiplexación del tiempo, TDMA[21]. Esta multiplexación define intervalos de tiempo, o *slots*, durante los cuales sólo una de las ONTs se comunica con la OLT, y viceversa (en cada intervalo de tiempo la OLT se transmite hacia una única ONT).

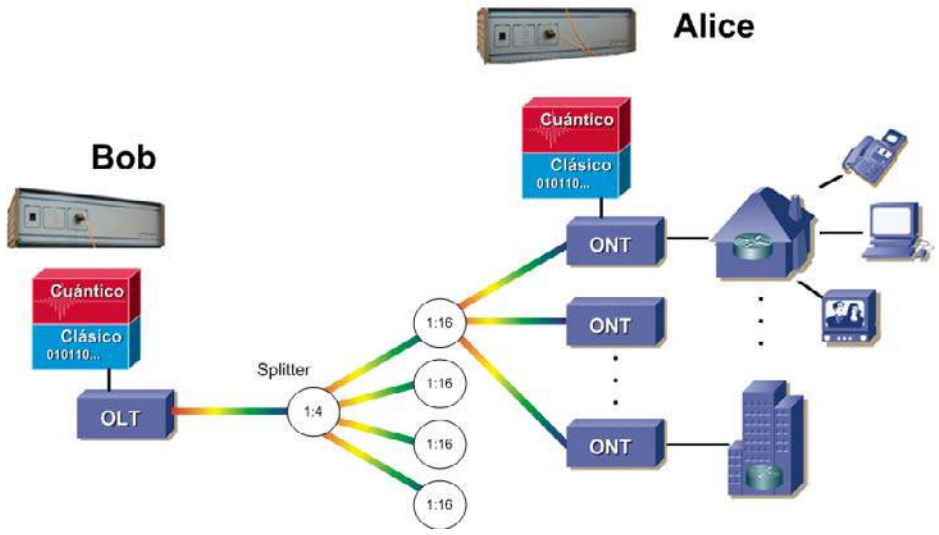


Figura 7.18: Esquema avanzado de una red PON con distintos niveles de splitters.

[21] *Time Division Multiple Access*: acceso múltiple por división en el tiempo.

En la actualidad existen múltiples tipos de redes PON (gestionadas mediante multiplexación del tiempo, TDMA): APON[22], BPON[23], EPON[24] y GPON[25] son algunos de los modelos de redes PON estandarizados. Y se trabaja en el desarrollo de nuevas especificaciones como la futura GEPON.

Tecnología WDM-PON

Si añadidos en una red PON la multiplexación en frecuencia, WDM, de las señales que se transmiten entre los extremos de la red, el resultado más inmediato es un incremento del ancho de banda disponible por cada usuario de la red. Este aumento en el ancho de banda disponible se debe a que los usuarios ya no deben compartir un único canal a través de la fibra que los une con el nodo central, OLT. Cada usuario, ONT, puede utilizar una frecuencia distinta para comunicarse con la OLT sin interferir con el resto de comunicaciones.

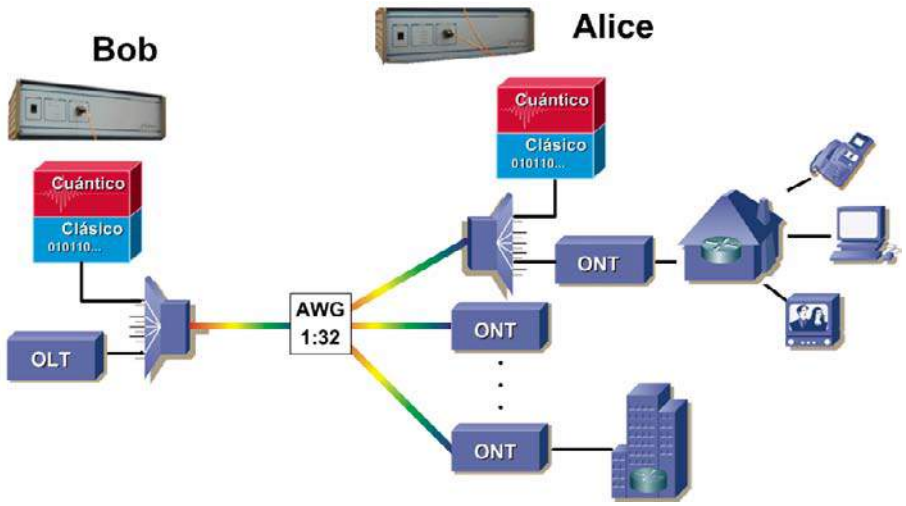


Figura 7.19: Esquema WDM-PON.

Para adaptar la tecnología PON a WDM-PON necesitamos dos nuevos componentes:

- Parece evidente que necesitamos un multiplexor en cada extremo de la red, ONT u OLT, aunque esto no es del todo cierto ya que, por ejemplo, los equipos que se sitúan detrás de cada ONT pueden estar preparados para trabajar con una frecuencia concreta (y sólo una), en cuyo caso no se requiere la extración de otras frecuencias incorporadas en la misma fibra.
- El otro componente que necesitamos es un sustituto del *splitter*, algo más avanzado, de tal forma que podamos realizar una división física de la fibra en función de la frecuencia. Este componente es el AWG[26]. La característica

[22] ATM PON, *Asynchronous Transfer Mode* PON.
[23] *Broadband* PON: red óptica pasiva de banda ancha.
[24] Ethernet PON.
[25] Gigabit PON.
[26] *Array Waveguide Grating.*

que lo diferencia de un splitter común aporta una de las mayores ventajas de esta nueva tecnología. Al no tener que dividir la señal que procede de la OLT para enviarla a todas las ONTs, la pérdida de potencia en un AWG es considerablemente menor a la del splitter, con lo que se incrementa sustancialmente la distancia máxima que puede separar a los extremos de una red WDM-PON.

Capítulo 8

Autenticación y certificación

A primera vista, la mecánica cuántica no aporta ningún mecanismo nuevo para la autenticación de los interlocutores de una comunicación, lo que en principio nos obliga a utilizar las estrategias de la criptografía convencional. Pero estas estrategias se basan en la emisión de certificados de clave pública, que es el mismo mecanismo que hemos sustituido con la criptografía cuántica. Esto nos lleva a pensar en nuevo concepto de certificación, basado en la distribución cuántica de claves, y es lo que estudiaremos en este capítulo.

8.1. Entidad certificadora

Imaginemos la configuración más sencilla de una red con tres nodos, donde por razones de economía los equipos se conectan siguiendo una topología en estrella. En cada intercambio de claves, el nodo central comparte una clave con alguno de los nodos exteriores, y por lo tanto, dicho nodo central es conocedor de todas las claves distribuidas en la red. La figura 8.1 muestra de forma gráfica el entorno que acabamos de describir.

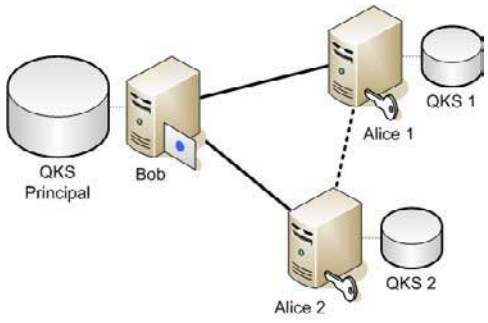


Figura 8.1: Red de distribución cuántica de claves con servidor de certificados.

Puesto que el nodo central posee cada una de las claves intercambiadas, puede autenticarse con cualquier extremo de la red. Si ampliamos esta autenticación a dos nodos cualesquiera, lo que estamos realizando es una certificación de dichos extremos. Por lo tanto, la distribución cuántica de claves nos permite implementar un nuevo mecanismo de autenticación basado en el registro de las

claves intercambiadas, para lo cual introducimos un nuevo concepto, el almacén cuántico de claves.

8.1.1. Almacén cuántico de claves

Un nuevo componente que pasa a formar parte de nuestro sistema QKD es el *almacén de claves*, QKS[1]. Comenzamos a estudiarlo en este apartado por su aplicación directa como registro, para el uso de certificados. Pero, como veremos a continuación, es un recurso con múltiples aplicaciones:

- Ya en los capítulos anteriores, debimos identificar la utilización de este tipo de almacenes a modo de *buffer*, o registro temporal, de claves de sesión. La seguridad que ofrece una clave está ligada a varios factores, pero dos de los más importantes son: el uso que se ha realizado de esa clave, y la aleatoriedad del mensaje cifrado. El hecho es que el primero de los factores (el uso) se ve afectado tanto por el tiempo de utilización de una clave, como por el volumen de los datos cifrados. Por esta razón, resulta interesante disponer de un buffer de claves que nos permita cambiar de modo ágil y efectivo entre distintas claves de sesión. En resumen, un buffer nos permite una gestión dinámica de las claves en función de los parámetros de ejecución del sistema.

- De forma similar, cada pareja de nodos que comparte un canal cuántico debe almacenar un número determinado de claves, que les permita autenticar el canal público en futuros intercambios de clave. Este registro es fundamental para poder completar un nuevo proceso de distribución. Visto de otra forma, si el almacén de claves de sesión está vacío, los extremos de la comunicación no pueden intercambiar mensajes cifrados. Pero si el almacén vacío es el utilizado para la autenticación del canal público, ambos extremos no pueden volver a compartir una clave. Recordemos que un requisito indispensable para el intercambio de una clave segura mediante un protocolo QKD es que la comunicación a través del canal público se realice de forma autenticada. Es decir, ambos extremos de la comunicación deben poseer una clave privada que les permita autenticarse mientras completan el intercambio de una nueva clave. Esta es la razón por la que muchos autores consideran que la distribución cuántica de claves debería emplear un término distinto, quizá más apropiado, como es aumento o *crecimiento cuántico de clave*s, QKG[2].

- Finalmente, un almacén cuántico de claves puede funcionar como registro de certificación. Como veremos en el apartado siguiente, dos interlocutores que intentan autenticarse pueden hacerlo utilizando parte de la clave compartida con una autoridad de certificación, utilizando la distribución cuántica de claves.

[1] *Quantum Key Store*: almacén cuántico de claves.
[2] *Quantum Key Growing*.

8.2. Certificación

A continuación, intentamos resumir de forma gráfica el proceso completo de certificación que han de seguir dos interlocutores, que no comparten ningún tipo de información previa, para poder establecer una comunicación privada entre ambos. El enfoque que hemos realizado desde este proceso de certificación se fundamenta en la utilización de los sistemas de criptografía cuántica para la distribución de claves. El escenario que presentamos será el mismo durante todo el proceso de certificación, y es el formado por los siguientes actores:

- Una **entidad de certificación**, capaz de intercambiar claves a nivel cuántico. Esta entidad debe disponer de un almacén de claves, QKS, con el objetivo de mantener en todo momento un registro de las claves distribuidas. Ese almacén o registro de claves le permitirá autenticar y cifrar una conversación con cualquier usuario que haya intercambiado previamente una clave, lo que será la esencia del proceso de certificación que pretendemos construir.
- Una red privada, donde sus nodos se conectan utilizando un canal de comunicación cuántico, a través del cual se distribuyen claves de forma segura mediante un sistema QKD como los estudiados.
- Una red pública, como puede ser Internet, donde dos interlocutores que se ponen en contacto desean autenticarse y comenzar una comunicación segura.

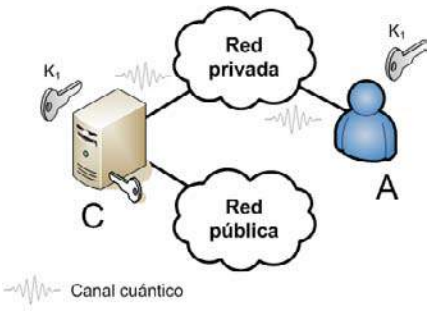


Figura 8.2: Autenticación. Paso 1.

Hemos visto que una forma bastante segura de distribuir claves es utilizar un canal de comunicación cuántico. La figura 8.2 ilustra ese hecho, el intercambio de una clave, K_1, a través de un canal cuántico entre un usuario, A, y una entidad de certificación, C. Este será el primer paso de nuestro proceso de certificación. Como hemos visto en los capítulos anteriores, ese canal cuántico no tiene por qué ser exclusivo, ni punto a punto, pudiento estar formado por una red de nodos conectados entre sí (evidentemente a través de canales cuánticos). Por esta razón, la conexión entre el usuario, A, y la entidad de certificación, C, se identifica en la figura por una red a la que denominamos privada[3].

[3]Consideramos que la red es privada por el hecho de permitir el intercambio de "secretos" entre los nodos de la red.

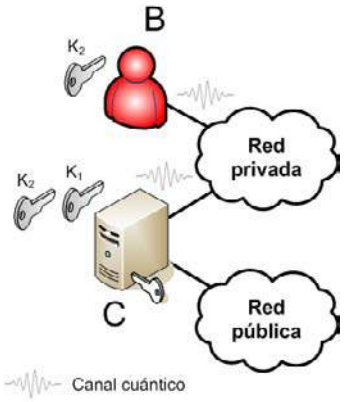


Figura 8.3: Autenticación. Paso 2.

En la siguiente figura, 8.3, la misma entidad de certificación, C, intercambia otra clave de sesión, K_2, con un nuevo usuario, B. En ese momento, la entidad de certificación mantiene un registro de las dos claves intercambiadas, K_1 y K_2, por lo que en cualquier instante puede autenticar a cada uno de los usuarios, A o B, utilizando una parte de la clave intercambiada. Es decir, la entidad de certificación está en disposición de establecer una comunicación segura con ambos usuarios, A y B.

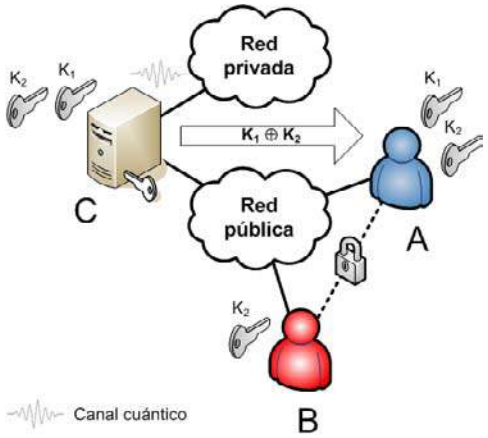


Figura 8.4: Autenticación. Paso 3.

Para acabar el proceso, la figura 8.4 muestra como ambos usuarios, A y B, pueden ahora autenticarse con la entidad de certificación, C, utilizando el canal de comunicación público, y solicitar a dicha entidad de certificación que proporcione la clave de uno de ellos (o parte de la clave) al otro usuario. De esta forma, los dos usuarios podrían comenzar una conversación segura a través de una red pública.

El algoritmo 10 muesta una lista de los pasos que acabamos de describir para completar un proceso de certificación entre dos usuarios utilizando un sistema QKD.

Algoritmo 10 Proceso de certificación utilizando un sistema QKD.

1. Un usuario, A, intercambia una clave, K_1, con una entidad certificadora, C, a través de un canal cuántico.
2. Otro usuario, B, intercambia una clave, K_2, con la misma entidad certificadora, C, a través del canal cuántico.
3. El usuario B se pone en contacto con la entidad certificadora, C, a través un canal público, autenticándose y cifrando la comunicación utilizando una parte de la clave K_2, para pedir a la entidad que envíe otra parte de esa clave, K_2, al usuario A.
4. La entidad certificadora C se pone en contacto con el usuario A, a través de un canal público, autenticando y cifrando la comunicación con el mismo utilizando una parte de la clave K_1, y le envía parte de la clave K_2 como solicitó el usuario B.
5. Los usuarios A y B pueden establecer una comunicación segura, a través de un canal público, autenticando y cifrando la conexión utilizando la parte de la clave K_2 que la entidad certificadora, C, puso en común.

Observación: La distribución cuántica de claves se presenta como una alternativa al cifrado asimétrico, donde se evita uno de los riesgos más importantes del este último como es la ruptura, mediante fuerza bruta, del cifrado. La seguridad del cifrado asimétrico se basa en la dificultad de cómputo que presentan ciertas operaciones, lo que supone que la seguridad de una clave asimétrica se reduzca con el tiempo. Esta reducción de la seguridad es aún más crítica cuando la clave es almacenada, como ocurre en las entidades de certificación. Por lo tanto, la ventaja en seguridad que aporta la criptografía cuántica es todavía mayor en los sistemas de certificación.

8.3. Un escenario futuro/real

En un futuro no muy lejano la tecnología de los sistemas QKD será adaptada a las redes de telecomunicaciones actuales, construyendo lo que hemos denominado como redes de distribución cuántica de claves, QKDN. Esas redes estarán formadas por nodos conectados a través de canales de comunicación cuánticos, que permitirán el intercambio continuo de claves entre cada par de nodos adyacentes de la red[4]. Como vimos en el capítulo anterior, a través de las redes pasívas del "último kilómetro" (redes PON) los nodos de una red QKDN pueden llegar hasta el usuario final por medio de lo que conocemos como: nodos

[4]El intercambio de claves entre nodos adyacentes será posible siempre y cuando los nodos sean compatibles, es decir, siempre y cuando los equipos QKD situados en cada extremo sean de distinto tipo. Además, mediante mecanismos como la multiplexación y el enrutado de ciertas longitudes de onda conseguiremos comunicar de forma directa dos nodos no adyacentes, pero la limitación en distancia estará siempre presente por lo que existe un problema pendiente: el transporte de claves entre nodos.

de acceso. El resultado de completar este proceso de integración es una red de distribución cuántica de claves universal.

Si a este escenario añadimos el concepto innovador de los almacenes de claves cuánticas, QKS, y las entidades de certificación descritas en el apartado anterior, ya no sólo estaremos llevando las claves proporcionadas por los sistemas QKD al usuario final a través de una red de uso común, sino que además tendremos un nuevo mecanismo de certificación universal. Claves más seguras, al alcance de todo el mundo, en un futuro donde la comunicación aparenta romper cualquier límite de acceso.

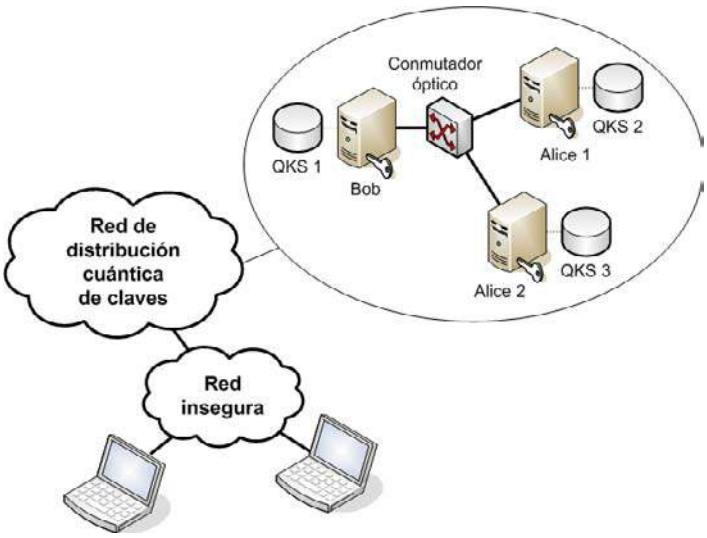


Figura 8.5: Aplicación práctica de una red de distribución cuántica de claves.

La figura 8.5 muestra un ejemplo práctico de lo que puede ser una aplicación futura de los sistemas de distribución cuántica de claves.

Parte IV

Ataques y vulnerabilidades

Capítulo 9

Ataques

9.1. Condiciones

Antes de proceder con el estudio de las distintas estrategias de ataque para un sistema QKD, debemos definir cuales son las condiciones de funcionamiento de dicho sistema, bajo las cuales debe acogerse cualquier hipotético espía. Esas condiciones son:

- Los extremos del sistema están conectados a través de dos canales de comunicación: uno cuántico o privado, y otro público o convencional. Puede darse el caso en el que la disposición física de ambos canales sea la misma[1], pero su utilización será distinta. Por ejemplo, podemos utilizar una única línea de fibra óptica para simultanear las comunicaciones cuántica y pública, bien sea dividiendo la fibra en canales de distinta frecuencia, o por medio de una modulación temporal; pero las condiciones de trabajo de ambos canales será distinta. Las condiciones de acceso a los canales cuántico y convencional son las siguientes:

  - El acceso al canal cuántico será total, teniendo la capacidad de hacer cualquier cosa que no esté prohibida por las reglas de la mecánica cuántica, y por lo tanto le fuerzan a:
    - No poder copiar o duplicar la información transmitida a través del canal.
    - La lectura de cualquier etado provocará la modificación del mismo.
  - El espía o atacante podrá leer toda la información transmitida a través del canal convencional, pero nunca podrá modificarla, ya que este se enceuntra autenticado.

- También, debemos estudiar la seguridad de un sistema QKD bajo el supuesto de que un atacante dispone de toda la tecnología "posible" a su favor.

[1] Por ejemplo, cuando utilizamos la tecnología WDM para la división en frecuencia de la fibra.

9.2. Estrategias para el ataque

9.2.1. Intercepta y reenvía. (Dropping, Man-in-the-middle)

La primera estrategia que nos viene a la mente a la hora de pensar en posibles ataques a nuestro sistema es la de interceptación y reenvío de la información intercambiada a través del canal cuántico. Bajo esta estrategia, un potencial atacante se colocaría entre los interlocutores de nuestro sistema, Alice y Bob, interceptando la información emitida por uno de los extremos y reenviando la información detectada al otro extremo.

A continuación veremos si esta estrategia es factible, y cómo puede llevarse a cabo de forma óptima para el atacante.

Sincronización

El primer obstáculo que ha de superar nuestro espía es la sincronización. La comunicación deber ser interrumpida por el atacante para poder interpretar la información transmitida, lo que provoca un lapso de tiempo medible (el tiempo necesario para la intercepción), y por lo tanto, dicha interrupción puede ser detectada por el extremo que recibe la información.

Pero a pesar de que las limitaciones en la sincronización pueden ser lo suficientemente importantes como para considerar el ataque como impracticable, debemos aferrarnos al hecho de que el atacante dispone de tecnología ilimitada. Y dentro de esa tecnología cabe la posibilidad de que el atacante disponga de un canal alternativo que optimice el tiempo necesario para que los pulsos lleguen hasta el extremo receptor, camuflando así el tiempo necesario por el espía para interpretar y reenviar la información transmitida. Debemos considerar por lo tanto este tipo de ataques como factibles.

Estrategias de medida. La base de Breidbart

Una vez que damos por factible la posibilidad de realizar este tipo de ataques, debemos estudiar cual es su implementación óptima para la obtención de la mayor información posible de la clave intercambiada. Para la ejecución de este ataque un espía puede utilizar dos estrategias distintas en función del tipo de medida realizada en la intercepción de los pulsos. Estudiaremos ambas estrategias desde el punto de vista de los protocolos basados en la utilización de cuatro estados no ortogonales, como ocurre en el protocolo BB84.

1. A primera vista, la forma más sencilla de implementar el ataque es utilizar de forma aleatoria las mismas bases empleadas por Alice y Bob para codificación y decodificación de la información. En este caso, y bajo el supuesto de que atacamos un sistema basado en la utilización de cuatro estados no ortogonales, el espía acertará con la base utilizada en el 50 % de las mediciones, donde la información interpretada será correcta al 100 %; mientras que en la otra mitad de los casos, donde la base utilizada no es la correcta, el resultado será aleatorio. En resumen, el porcentaje de información

disponible por un espía con este ataque es del 75 %, y por lo tanto, el porcentaje de error introducido por mala interpretación de la información es del 25 %.

2. Pero el espía no tiene por qué utilizar las mismas bases que Alice y Bob, pudiendo buscar una estrategia que optimice la probabilidad de acierto en cada medida. Esta probabilidad de acierto se maximiza cuando utilizamos una base intermedia, con un ángulo de $22\frac{1}{2}$ conocida como la **base de Breidbart**, [14] [30]:

$$\begin{aligned} |B_+\rangle &= \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}|1\rangle \\ |B_\times\rangle &= -\sin\frac{\pi}{8}|0\rangle + \cos\frac{\pi}{8}|1\rangle \end{aligned}$$

Con esta base la probabilidad de acierto aumenta aproximadamente hasta el 85,35 % ($p = \cos(\pi/8)^2 = (2+\sqrt{2})/4 \approx 0{,}8536$ [14]), mientras que el error introducido se sigue manteniendo en el 25 %. La diferencia con respecto a la estrategia anterior es que toda la información obtenida utilizando una base intermedia es no determinista, mientras que en el caso anterior 2/3 de la información conseguida era exacta.

Tasa de error

El principal inconveniente de este ataque es el alto porcentaje de error introducido en la comunicación. Independientemente de la estrategia de medida elegida para el ataque, la tasa de error provocada es del 25 %, una tasa de error demasiado elevada para pasar desapercibida, siendo incluso irresoluble para las estrategias de corrección de errores utilizadas[2].

Ahora bien, el atacante puede ajustar la cantidad de pulsos a interceptar con el objetivo de reducir la tasa de error a un margen que pueda inducir al atacado a continuar con la transmisión en lugar de descartar la posibilidad de transmitir una clave secreta. Así por ejemplo, si un atacante desea que la tasa de error introducida sea inferior al 8 %, tan sólo debe reducir el número de pulsos interceptados a 1/3 del total, que sigue siendo un porcentaje elevado de la transmisión. Como consecuencia, estamos siempre obligados a aplicar algún mecanismo de amplificación de la privacidad que tenga en cuenta este tipo de ataques, que pueden estar camuflados en la tasa de error, o QBER obtenido.

9.2.2. División del número de fotones. (Photon Number Splitting)

Como ya vimos en los capítulos iniciales de este proyecto, no existen fuera de prototipos experimentales fuentes de emisión de fotones individuales y en su lugar utilizamos pulsos láser atenuados. El inconveniente de esta alternativa es su falta de precisión, por la cual obtenemos pulsos con un número de fotones variable. Esta imprecisión también está presente en algunos sistemas QKD basados en pares entrelazados, siempre en función del tipo de fuente de pares

[2] Recordemos que el procedimiento utilizado actualmente para la corrección de errores, Cascade, sólo puede corregir una tasa máxima del 15 % de errores en la comunicación.

EPR utilizada. El número de fotones que encontramos a la salida del emisor, Alice, sigue una distribución estadística de *Poisson* de media μ, como muestra la expresión:

$$P(n,\mu) = \frac{\mu^n}{n!}e^{-\mu}$$

Donde n es el número de fotones que contiene el pulso atenuado, y μ es el número promedio de fotones esperado a la salida del atenuador.

A modo de ejemplo hemos incluido una tabla, 9.1, donde mostramos los valores de las probabilidades según esta distribución para un número promedio de fotones inferior a 1, concretamente para $\mu = 0{,}5$ y $\mu = 0{,}1$. En el primero de los casos, $\mu = 0{,}5$, podemos ver como la probabilidad de enviar más de un fotón es superior al 9 %, lo que podemos considerar como un riesgo límite o excesivo puesto que, el espía puede obtener una parte considerable de la clave a partir de la medición de los fotones adicionales en cada pulso, sin que los interlocutores tengan modo alguno de detectarlo.

μ	p_0	p_1	p_2	p_3	p_4	p_5
0,5	60,65 %	30,33 %	7,58 %	1,26 %	0,16 %	0,02 %
0,1	90,48 %	9,05 %	0,45 %	0,01 %		

Tabla 9.1: Distribución de fotones de un pulso atenuado para $\mu = 0{,}5$ y $\mu = 0{,}1$.

Este tipo de ataques se conocen bajo los nombres de *beamsplitting*, o *photon number splitting*, PNS. Y aunque tecnológicamente se encuentran lejos de poder ser implementados, debemos tenerlos en cuenta a la hora de implementar cualquier mecanismo de amplificación de la privacidad (siempre y cuando utilicemos fuentes de fotones con este riesgo estructural).

Para intentar reducir el riesgo de este tipo de ataques podemos adoptar varias estrategias:

- La estrategia más sencilla que podemos emplear es la reducción del número promedio de fotones a la salida de la fuente (del atenuador), con el objetivo de reducir al mismo tiempo el número de pulsos que contienen más de un fotón. El inconveniente de aplicar esta solución es que reducimos drásticamente la eficiencia del sistema, puesto que se reduce de forma exponencial el número de fotones que alcanza el extremo final de la comunicación. En el siguiente artículo [50] D. Pearson y C. Elliot exponen un estudio del cálculo óptimo del número promedio de fotones para un sistema QKD.

- Otra forma de paliar los efectos de los ataques PNS es utilizar un protocolo específico, como el SARG04, que reduce la cantidad de información que puede obtener un espía de la interpretación de los pulsos con más de un fotón. El inconveniente asociado a este protocolo reside en el hecho de que el porcentaje de bits reconciliados es tan sólo del 25 % sobre el tamaño de la clave en bruto (número de bits detectados), en lugar del 50 % habitual.

- Seguramente la estrategia más apropiada para incrementar la seguridad de nuestra fuente de fotones es la que utiliza los estados trampa, o *decoy states*. La ventaja principal que aporta esta estrategia es la capacidad para

detectar la presencia de un intruso, sin reducir la eficiencia del sistema. Estos sistemas se basan en usar más de un número promedio de fotones por pulso. Valores típicos son $\mu = 0{,}8$ y $\mu = 0{,}2$. Durante el proceso de reconciliación uno de los dos conjuntos de pulsos (típicamente el menos intenso) es usado para detectar el espía. Ya que el espía tiene que ajustar su modo de proceder para un μ determinado, el no saber a cual de los dos conjuntos pertenece el pulso hace que sea detectable [34].

Finalmente, queremos comentar tan sólo la existencia de otras estrategias de ataque basadas en este riesgo estructural de las fuentes de fotones individuales, que podemos estudiar en los artículos [11].

9.3. Ataques experimentales

En la sección anterior hemos visto las distintas estrategias de ataque desde el punto de vista teórico. Ahora veremos otra serie de estrategias de ataque basadas en los defectos de los componentes utilizados en los sistemas QKD actuales. Se trata, por lo tanto, de ataques desde el punto de vista experimental.

Anotamos sólo como referencia a Vadim Makarov, autor de la mayoría de los estudios realizados recientemente acerca de posibles ataques experimentales sobre sistemas QKD [60, 61].

La primera demostración práctica de un ataque exitoso sobre un sistema QKD real es la realizada por Hoi-Kwong Lo et al. [63]. La estrategia utilizada para el ataque se basa en un desplazamiento en el tiempo, *time-shift*, de los pulsos intercambiados entre Alice y Bob. Utilizando la tecnología actual, el atacante alarga o acorta la fibra utilizada como canal cuántico entre los extremos de la comunicación. Este desplazamiento de la fibra se va a traducir en una modificación de la distribución de probabilidad de activar uno u otro detector, es decir, de obtener un 1 o un 0 en Bob (ver figura 4.2). Debido a un mal ajuste o emparejamiento de los detectores, *detector mismatch*, los sistemas con dos o más detectores pueden ser sensibles a este tipo de ataques, que incrementan considerablemente la información que un espía puede obtener de una clave intercambiada mediante QKD.

9.4. Otras estrategias para la realización de ataques

Hasta el momento hemos estudiado distintas estrategias para el ataque de un protocolo QKD desde los puntos de vista teórico y práctico. En estas estrategias nos hemos centrado en el análisis de la información que puede obtener un espía de su actuación sobre el canal cuántico (durante el intercambio de fotones), y de la lectura del canal público (durante el proceso de reconciliación de bases y corrección de errores). Ahora bien, si regresamos al capítulo donde estudiamos la arquitectura de un sistema QKD, comprobaremos que nos hemos dejado un segmento de esa arquitectura donde no hemos tenido en cuenta las vulnerabilidades del sistema: se trata del proceso de **autenticación**.

Al describir los protocolos QKD expusimos una serie de condiciones que deben cumplirse en la ejecución de dichos protocolos, y sin las cuales no está

garantizada la seguridad de los mismos. Una de esas condiciones es que la comunicación a través del canal público debe estar en todo momento autenticada. Para realizar esa autenticación necesitamos una clave idéntica en ambos extremos de la comunicación, lo que sugiere que un sistema QKD debe utilizar parte de la clave intercambiada en un proceso anterior como llave de autenticación en los siguientes intercambios. Esta dependencia de claves para la autenticación hace que un sistema QKD no sea tan sólo un sistema generador de claves, puesto que también es consumidor de las mismas. En consecuencia, un sistema QKD sólo será rentable si la clave generada en un proceso de intercambio es mayor que la clave utilizada para la autenticación del siguiente intercambio. Además, esa realimentación de claves nos muestra a la distribución cuántica de claves desde una perspectiva distinta, más parecida a un proceso de regeneración, al que hacemos referencia con un nuevo término: **crecimiento cuántico de claves**, QKG[3] —seguramente más apropiado que el término QKD—.

Jörgen Cederlöf estudia en su tésis [56] los aspectos de seguridad relacionados con la autenticación utilizada en los sistemas QKD. Las conclusiones a las que llega Cederlöf vienen resumidas en un artículo [64] donde plantea un posible ataque a los sistemas QKD a través de la autenticación. Como comenta el autor, en la actualidad los sistemas QKD utilizan como mecanismo de autenticación estándar el propuesto por Wegman y Carter [3, 4], donde la seguridad está íntimamente ligada a la aleatoriedad de los bits utilizados en la clave de autenticación. Esa aleatoriedad no está garantizada en los sistemas QKD puesto que un atacante puede actuar sobre el canal cuántico, ya no para obtener directamente información del mismo, sino para modificar la distribución probabilística de los valores intercambiados que formarán parte de la clave. Una pequeña variación en la distribución de las detecciones en Bob proporciona al espía un cierto conocimiento de la clave intercambiada, inicialmente mínimo, pero que puede incrementar con el tiempo. El incremento en la información que obtiene el espía de la clave se debe fundamentalmente a que parte de la clave intercambiada se utiliza en el proceso de autenticación del siguiente intercambio de clave. Si en los siguientes intercambios el espía vuelve a realizar el mismo ataque, la información que el atacante posee de la etiqueta utilizada para la autenticación puede crecer de igual forma que lo hacen las claves intercambiadas (las claves realmente crecen como comentamos al referir el término QKG), pudiendo así comprometer la seguridad de las claves compartidas mediante QKD.

El mismo autor, Jörgen Cederlöf, propone dos opciones como mecanismo de prevención: por un lado, el aumento de las etiquetas utilizadas para la autenticación, y por otro, la reducción de la clave final en el proceso de amplificación de la privacidad. Ambos mecanismos supondrían un incremento del tiempo de vida “segura” de los protocolos QKG, pero no serían la solución definitiva a este tipo de ataques.

[3] Quantum Key Growing.

Capítulo 10

Vulnerabilidades

10.1. Aleatoriedad

Son muchos los detalles que debemos cuidar en el diseño e implementación de un sistema QKD, pero sobre todos esos factores destaca uno, la *aleatoriedad*. Es inmediato comprender que la seguridad de un secreto reside en nuestra capacidad para mantenerlo oculto (para mantenerlo realmente *secreto*), o dicho de otra forma, la seguridad de una clave reside en su desconocimiento, para lo cual debemos vigilar todos los estados por los que pasa dicha clave a lo largo de su vida: la generación de esa clave, su distribución y almacenamiento. Pues bien, dejando aparte el almacenamiento, los otros dos procesos (la generación de claves y su distribución[1]) dependen directamente de la generación de secuencias de valores aleatorios, lo que convierte a estas secuencias, o más concretamente al generador de dichas secuencias, en un punto clave de la seguridad de nuestro sistema.

Antes de continuar, debemos aclarar qué es lo que entendemos por vulnerable, no en su definición sino en el contexto donde lo aplicamos. Porque un sistema QKD no es más fácil de atacar por medio de su generador de números aleatorios, pero cualquier debilidad en dicho generador compromete la seguridad de todo el sistema.

En este proyecto trabajamos con los dos estados: la generación de claves y su distribución, que encontramos en la entrada y la salida de datos del sistema.

10.1.1. El generador de números aleatorios

En la actualidad, la tendencia nos lleva al uso de componentes hardware para la generación de secuencias de valores aleatorios, que podemos encontrar también en su versión cuántica, como muestra la figura 10.1[2]. La razón por la que se utilizan elementos externos para la generación de números aleatorios se fundamenta en que la naturaleza de un ordenador es, de por sí, determinista.

[1] En este contexto, cuando hablamos de distribución de claves nos referimos exclusivamente a la distribución cuántica de claves, QKD.

[2] Imagen extraída de la página web de *id Quantique*. http://www.idquantique.com/ `http://www.idquantique.com/`

Bajo esta misma justificación, la naturaleza probabilística de la mecánica cuántica propone a este nivel físico como el recurso genuino para la producción de secuencias aleatorias.

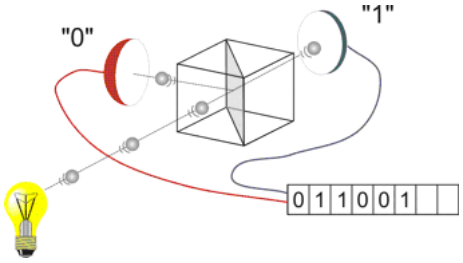


Figura 10.1: Esquema conceptual de un generador cuántico de números aleatorios. Esta es la idea usada en el sistema "*Quantis*" de id Quantique.

En el presente proyecto no hemos utilizado ningún equipamiento especial para la generación de números aleatorios, pero con el objetivo de validar los resultados obtenidos hemos cuidado la elección de cada secuencia aleatoria, utilizando registros de valores perfectamente aleatorios previamente generados y testeados. Estos registros aleatorios se han obtenido a través de fuentes de públicas y gratuitas [5].

10.1.2. Distribución de las detecciones

En el apartado anterior estudiamos el primer impacto de la aleatoriedad en nuestro sistema, que se obtiene a partir del generador de números aleatorios, pero además, debemos controlar otro segmento del sistema relacionado directamente con la aleatoriedad, y son los resultados. Bajo el supuesto de que los datos de partida son perfectamente aleatorios, debemos suponer que el resultado de los protocolos ejecutados para la distribución de claves es también perfectamente aleatorio.

El resultado que debemos estudiar ahora son las detecciones, y más concretamente su distribución. En este punto debemos interpretar el concepto de distribución como reparto. Con el objetivo de proporcionar la menor información posible a un espía, nuestro sistema debe repartir las detección en pulsos perfectamente aleatorios, de tal forma que la probabilidad de obtener una detección sea la misma en cualquier instante de la ejecución.

Para comprobar la aleatoriedad de los resultados de nuestro sistema, lo primero que hicimos fue reunir una cantidad de pruebas lo suficientemente considerable como para que la distribución de resultados fuera constante. Mostramos esos resultados en la siguiente gráfica (10.2)[3], en la que descubrimos una curiosa irregularidad al comienzo de las detecciones.

Evidentemente, esa irregularidad supone un riesgo estructural, pero ante todo nos interesa conocer cual es la causa de esa extraña distribución, y la encontramos en el *dead time*. Si recordamos el funcionamiento de los detectores de fotones, nos encontraremos con una característica común a todos, que es

[3]En la gráfica se muestra el número de detecciones obtenidas en cada uno de los puntos de envío de un tren de pulsos, donde el tamaño utilizado para el tren se pulsos es de 624 fotones.

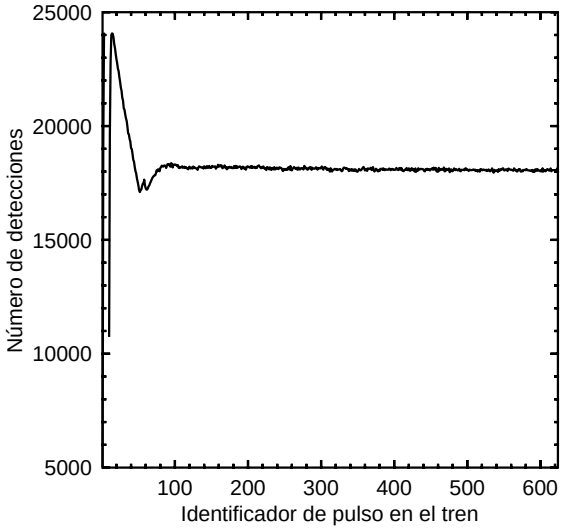


Figura 10.2: Distribución de las detecciones en función de los pulsos por tren.

el tiempo de bloqueo, o dead time, que sufren dichos componentes debido a su inestabilidad temporal después de haberse activado por una detección. Ese tiempo de bloqueo es el que provoca la inestabilidad de las detecciones al comienzo de una transmisión debido a que la probabilidad de obtener una detección al inicio de una secuencia de transmisión, es superior a causa de que inicialmente el sistema parte de un estado de reposo, donde las primeras detecciones no se ven afectados por un dead time anterior. Este comportamiento hace que el QBER efectivo sea menor al principio del tren y mayor al final, lo que contrasta con la idea de QBER constante que se ha aplicado siempre en criptografía cuántica.

10.2. Pulsos fantasma

Puede considerarse habitual, o bien es cierto que no suele ser extraño, encontrar en toda fiesta algún invitado que no es bienvenido. La molestia no va más allá de movernos en un entorno en el que no pueda afectarnos, para lo cual sólo debemos estar prevenidos. Y esta experiencia —que puede parecernos exclusiva de la vida real— formará parte de nuestros sistemas, lo que nos obligará a tomar las medidas pertinentes, para lo cual debemos conocer en primer lugar quién es el invitado.

Una vez más tenemos que viajar al mundo de las comunicaciones por fibra óptica para comprender la situación que se nos presenta, centrándonos en un parámetro característico de la fibra: la *atenuación*. Existen dos razones fundamentales por las que un pulso de luz transmitido por medio de una fibra óptica puede atenuarse:

1. Debido a la absorción de la luz por parte del material sobre el que está construida la fibra.

2. A causa de la presencia de impurezas.

El primer factor es el más común, y no depende exclusivamente del material que compone la fibra, ya que la longitud de onda es otro coeficiente que influye directamente en el nivel de absorción de la fibra. Pero el factor de atenuación que nos interesa estudiar en este momento es el segundo, la presencia de impurezas. Estas impurezas, o irregularidades en general, producen una desviación del haz de luz que puede provocar la atenuación del pulso, y que tiene un efecto muy particular cuando esta desviación se convierte en reflexión debido a que el ángulo de incidencia supera los 180º. El resultado es un pulso atenuado que sigue un cambio de sentido, y que conocemos como **pulso reflejo**. Un agente externo que actúa de forma similar a una impureza es un conector.

Resulta como poco curioso el efecto que pueden tener estos pulsos reflejo sobre el funcionamiento de nuestro sistema, principalmente por su baja intensidad, pero además, porque su influencia requiere de la coincidencia de un conjunto de condiciones muy precisas. Pero antes de estudiar estas condiciones debemos ser conscientes de los parámetros que estamos manejando, y es que un pulso reflejo posee una relación directa con el pulso original que equivale a una atenuación de 30 dB, lo que en condiciones normales (con una longitud de onda de 1550 nm) es la pérdida correspondiente a 120 Km de fibra[4].

Ya conocemos todos los elementos que van a afectar a nuestro sistema y su grado de influencia. Ahora los reunimos todos en un diagrama de ejecución como el mostrado en la figura 10.3, y comprobamos cual es comportamiento final del sistema.

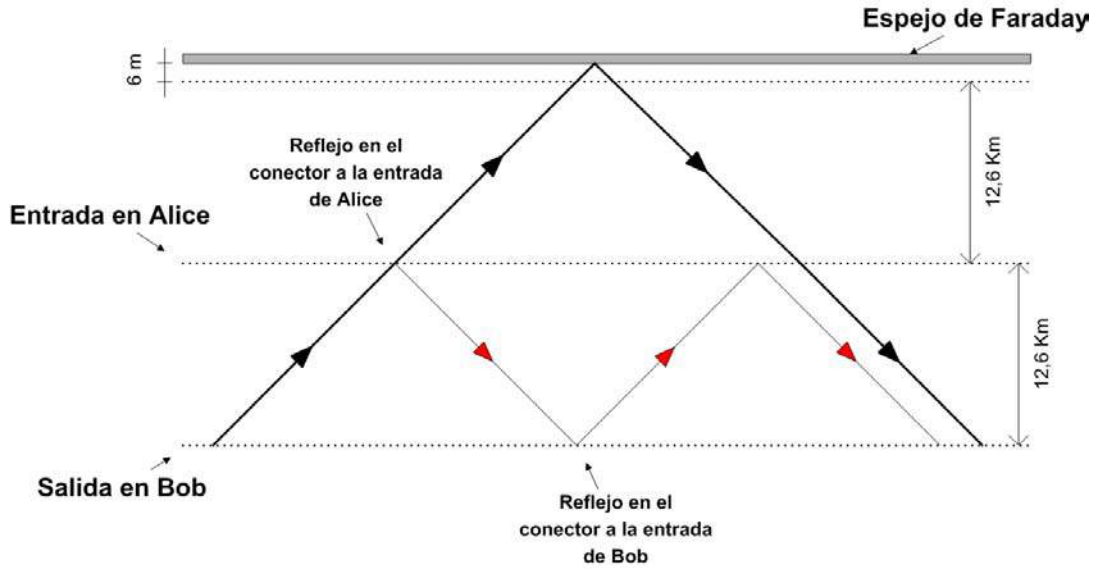


Figura 10.3: Recorrido en el tiempo de un pulso láser emitido desde Bob, junto a los pulsos reflejo que va generando.

Sorprendentemente, podemos comprobar a primera vista que la utilización de líneas de comunicación, de longitud proporcional a la bobina de fibra incluida dentro de Alice, puede provocar simetrías en el diagrama de generación de pulsos

[4]El cálculo para la atenuación se ha realizado suponiendo una pérdida de 0.25 dB/Km.

reflejo, que concluyen en la presencia de dos pulsos muy cercanos en el tiempo (y distancia), a la vuelta del haz de luz originado en Bob.

Algunos detalles importantes que vemos a primera vista son los descritos a continuación. En primer lugar, vemos que los pulsos reflejo sólo nos pueden afectar en los sistemas de doble dirección, o *plug and play*, en los que el haz de luz inicial, que va desde Bob hasta Alice, es capaz de generar un pulso reflejo que interfiera con el fotón que regresa finalmente a Bob. En segundo lugar, es imprescindible que el pulso reflejo, generado inicialmente a la entrada de Alice, rebote un número determinado de veces. El número de veces que deberá rebotar el pulso para aproximarse al fotón de vuelta, dependerá de la configuración interna en Alice, como muestran las siguientes figuras (10.3, 10.4 y 10.5). Cuando la longitud del recorrido de un pulso dentro de Alice es similar a la distancia que separa a los extremos de la comunicación, Alice y Bob, el pulso fantasma que intercepta al fotón de vuelta necesita reflejarse hasta 3 veces.

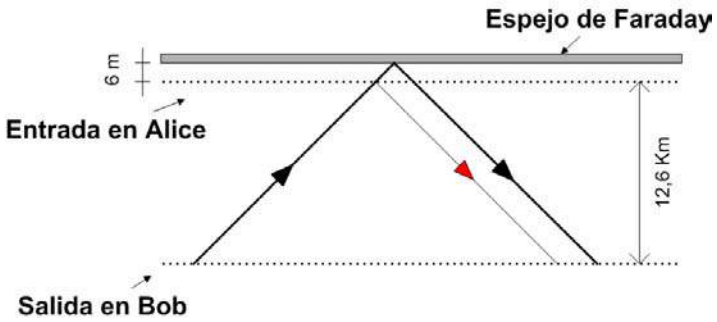


Figura 10.4: Actuación de un pulso reflejo con un recorrido interno, en Alice, de corta longitud.

Por otro lado, cuando la distancia que recorre un pulso dentro de Alice es muy pequeña, el primer pulso reflejado a la entrada en Alice ya puede interferir en los resultados. Mientras que el caso opuesto, cuando la distancia recorrida dentro de Alice es muy superior a la que separa a Alice de Bob, el número de pulsos reflejo aumenta.

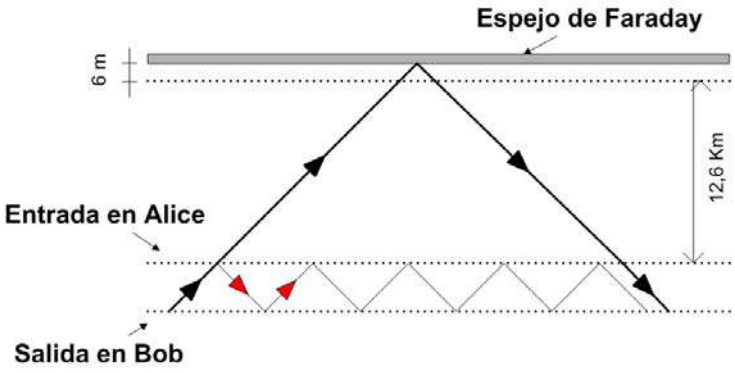


Figura 10.5: Interferencia de un pulso reflejo con una separación entre Alice y Bob inferior a la longitud del recorrido dentro de Alice.

En conclusión, las condiciones necesarias para que nuestro sistema se vea afectado por la presencia de pulsos reflejo son las siguientes:

- El sistema debe ser de doble dirección (o plug & play). Como se muestra en el diagrama anterior (10.3), los pulsos reflejo se crean en el trayecto de ida del haz de luz original.
- La longitud de la línea que comunica a los extremos del sistema debe ser un número entero de veces la longitud de la línea incluida dentro de Alice.

10.2.1. Una situación real

Hasta el momento hemos hablado tan sólo de la existencia de pulsos reflejo. Pulsos que se pueden localizar muy cerca de los fotones que vuelven desde Alice (ya que sólo nos interesan los reflejos ocurridos en los sistemas de doble dirección), pero que en muy rara ocasión van a interferir directamente con el fotón de vuelta. Nos preguntarnos entonces, ¿cual es el impacto real de estos pulsos reflejo?

El hecho es que en un sistema real ejecutamos una secuencia de envíos continuada, con una separación constante (que en nuestro caso es de 200 ns, equivalente a 40 m), lo que multiplica las probabilidades de que un pulso reflejo pueda interferir con un fotón de vuelta, quizá no del pulso original, pero sí de un pulso correspondiente a otro envío.

Pero además, tenemos un problema añadido y es la diferenciación entre un pulso reflejo y el pulso correcto. En nuestro caso, ese es el problema principal que se nos presenta, ya que el software proporcionado por id Quantique para la medición de la línea que comunica a Alice y Bob no tiene en cuenta este problema. Por esta razón, si ejecutamos el procedimiento de medición de la línea con los parámetros deseados de ejecución, es decir, utilizando la atenuación correspondiente para el envío de un único fotón, entonces el proceso de medición nos detectará el pulso fantasma en vez del pulso correcto debido a la intensidad del primero. Por lo tanto, nos vemos obligados a cambiar los parámetros de ejecución en el proceso de medición, para que el pulso detectado sea el correcto. Es decir, debemos bajar la atenuación hasta que la intensidad del pulso generado sea superior a la del pulso fantasma.

Las capturas 10.6 y 10.7 muestran los resultados del proceso de medición de la línea utilizando el software de id Quantique. En la primer de las capturas, el pulso detectado corresponde con el pulso reflejo, por lo que los valores de calibrado son inferiores a los esperados (un *Waiting Period* de 5109, como el que aparece en la captura indicada, no es el valor esperado para la longitud de línea utilizada). Mientras que la segunda captura muestra los valores correctos.

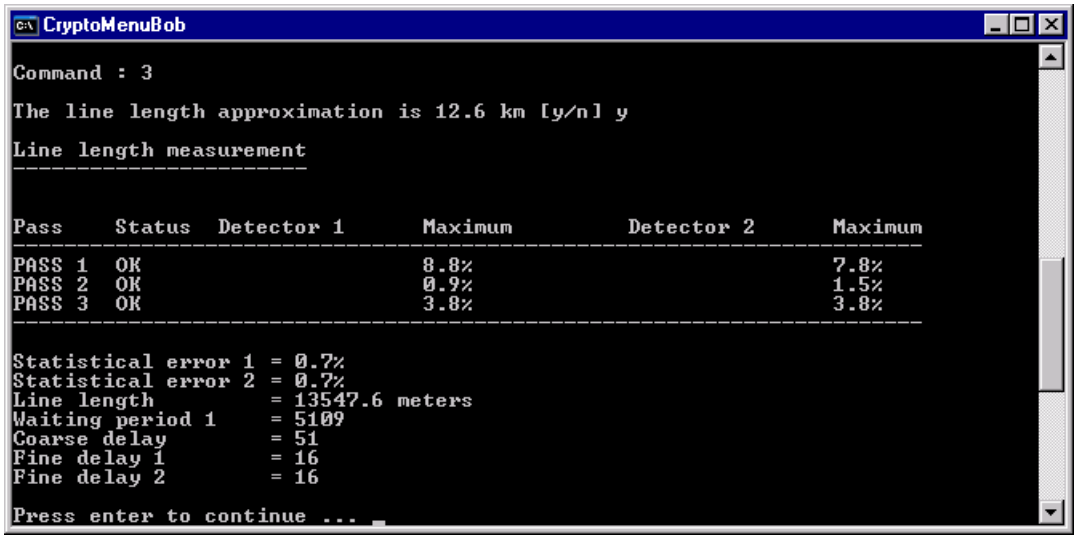


Figura 10.6: Resultado de una medición de la línea con detección del intervalo de llegada del pulso fantasma.

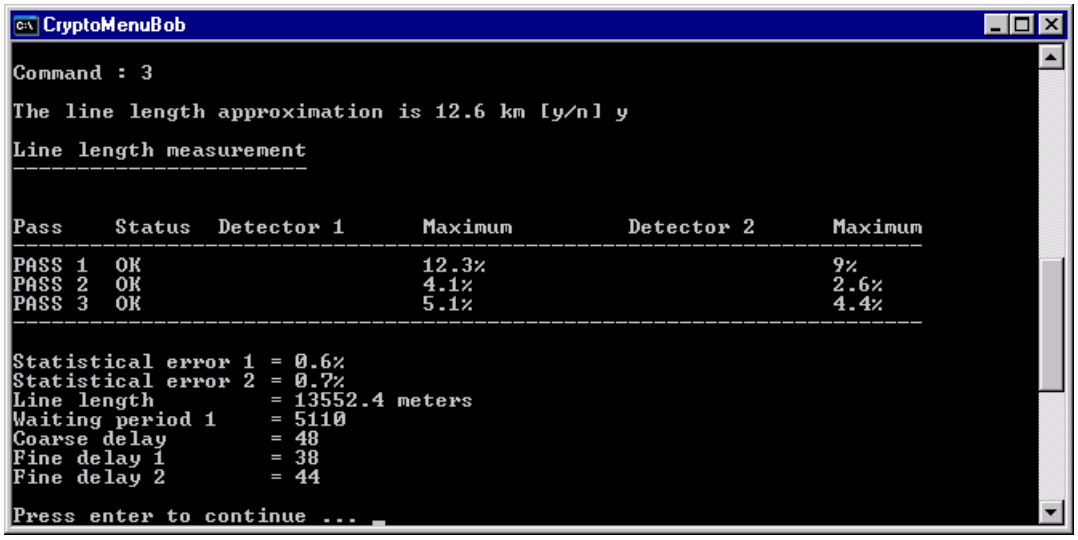


Figura 10.7: Resultado de una medición de la línea con detección del intervalo de llegada de los fotones correctos.

Parte V

Apéndice

Capítulo 11

Perspectivas de futuro

Resulta complicado aventurar el futuro de los sistemas de distribución cuántica de claves. Aunque más que complicado es arriesgado intentar predecir su impacto en el mercado, ¿será la criptografía cuántica un desarrollo para todas las masas o sólo en ciertos nichos de mercado? Parece evidente que uno de los principales factores para el desarrollo de esta tecnología es el económico, pero existen ejemplos de otras tecnologías que se han hecho un hueco en la sociedad sobrepasando esa barrera financiera. Sin ir más lejos, la telefonía móvil es un ejemplo de lo que hablamos, puesto que hoy día, un mercado tan complicado como puede ser el del hogar ha creado esa necesidad de comunicación en todo momento a pesar del coste que lleva asociada, y de existir otra tecnología similar, ya implantada, y de coste mucho más reducido (como es la telefonía fija). La pregunta que debemos hacernos puede ser: ¿será la seguridad una necesidad del futuro? Parece que sí, si interpretamos la seguridad como privacidad en un entorno futuro donde todo estará interconectado.

11.1. Focos de investigación

El desconocimiento de estos sistemas o la tecnología que llevan implícita puede hacer pensar a muchas personas que este tipo de desarrollos está muy lejos de ser comercial, o creer que se trata una tecnología joven. Pero esto no es así. Bien es cierto que la investigación sobre este tipo de sistemas no ha acabado, al igual que ocurre con muchos otros sistemas de comunicación, donde la investigación es una rama que siempre estará presente intentando mejorar la tecnología. Quizá el ejemplo más claro de que esta tecnología ha dado ya el gran paso hacia el mercado es el hecho de que, son varias las disciplinas que trabajan ahora mismo para mejorar los sistemas QKD: la física teórica y experimental, ciencias matemáticas, telecomunicaciones e informática, son cuatro disciplinas que estudian hoy día el presente y futuro de esta tecnología. Veremos de forma individual donde se centran los trabajos de cada una de estas disciplinas.

Físico/experimental

La línea de investigación experimental de los sistemas QKD se centra principalmente en los dos de sus componentes clave: las fuentes de fotones y los

detectores. Al mismo tiempo que estudia otras estrategias para el transporte o codificación de la información.

- Fuentes de fotones. Las fuentes de fotones individuales siguen siendo un mito tecnológico, y su alternativa, las fuentes de fotones atenuados, presentan uno de los riesgos más importantes para la seguridad de los sistemas QKD: los ataques por PNS. Como punto crítico para la seguridad de los sistemas QKD, las fuentes de fotones son un objetivo muy importante en la investigación actual.
- Detectores. La tecnología de detección es una de las secciones de la criptografía cuántica que más a progresado en los últimos años, y sobre la que más esperanzas hay depositadas. Además, el impacto de esta tecnología sobre el rendimiento de un sistema QKD es directo: un incremento en la eficiencia de los detectores de un sistema QKD se traduce en un incremento equivalente en la eficiencia final del sistema.
- Nuevos mecanismos para el transporte de la información: variables continuas, sistemas coherentes (COW[1]), o entrelazamiento, son algunas de las alternativas. La progresión de algunos de estos mecanismos, como el entrelazamiento, puede ser clave para el desarrollo de otras tecnologías como los repetidores cuánticos.

Teórico

Desde el punto de vista teórico, centrado en los sistemas de información, los objetivos principales de investigación corresponden con los niveles finales en la pila de protocolos QKD, donde encontramos:

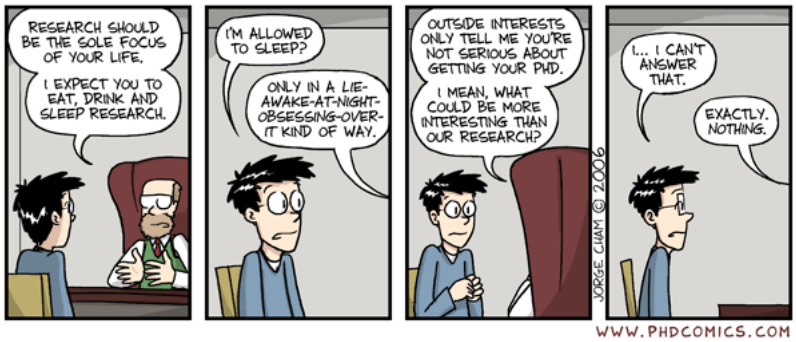
- Nuevos procedimientos para la corrección de errores, que permitan corregir una alta tasa de error proporcionando la mínima información posible. De forma adicional, a los futuros procedimientos de corrección de errores de los sistemas QKD se les pide una nueva característica: un consumo del ancho de banda más compacto. En la actualidad, procedimientos como el *Cascade* requieren de un tráfico de red escaso en volumen, aunque elevado en tiempo de latencia debido al gran número de transmisiones que deben realizarse entre los extremos de la comunicación.
- Amplificación de la privacidad y estudios analíticos de la seguridad, dependientes o no de los dispositivos utilizados.
- Autenticación y transporte de claves.

Telecomunicaciones

El salto a la rama de la ingeniería se está produciendo en este momento. Como consecuencia de esto no existe una estrategia definida en cuanto a los objetivos principales de investigación, pero el trabajo actual parece mostrar que la mayor parte del esfuerzo se centrará en:

[1] Coherent One-Way.

- Diseño y optimización de redes de distribución cuántica de claves, distinguiendo dos estrategias: redes de acceso y *backbones*.
- Integración a nivel hardware con las tecnologías de redes ópticas comerciales, en especial con las redes ópticas pasivas, PON[2].
- Integración a nivel software con los protocolos actuales para la distribución de claves y cifrado de las comunicaciones.



[2]Passive Optical Network.

Apéndice A

Definiciones y demostraciones

A.1. Resumen de algunas definiciones importantes

- Teorema de no-clonación:

$$\psi \otimes |b\rangle \otimes |A_i\rangle \rightarrow \psi \otimes \psi \otimes |A_j\rangle$$

- Representación de los cuatro estados utilizados en los protocolos BB84 y SARG:

$$|\psi\rangle = \bigotimes_{i=1}^{n} |\psi_{a_i b_i}\rangle \tag{A.1}$$

$$|\psi_{10}\rangle = \cos\frac{\pi}{4}|\psi_{00}\rangle + \cos\frac{\pi}{4}|\psi_{01}\rangle$$

$$|\psi_{11}\rangle = \cos\frac{\pi}{4}|\psi_{00}\rangle - \cos\frac{\pi}{4}|\psi_{01}\rangle$$

- Distribución binomial:

$$Bin(n,p) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k} \tag{A.2}$$

- Distribución de Poisson:

$$P(n,\mu) = \frac{\mu^n}{n!} e^{-\mu} \tag{A.3}$$

Cuando el número esperado de ocurrencias es muy bajo, $\mu \leq 0{,}1$, utilizamos la expresión simplificada de la distribución de Poisson $P(n,\mu) = \mu^n/n!$ (puesto que $e^{-\mu} \approx 1$).

- Energía de una partícula:

$$E = \hbar\omega = h\nu = h\frac{c}{\lambda} \tag{A.4}$$

Donde h es la constante de Planck, $h = 6{,}626 \times 10^{-34} J \cdot s$, c es la velocidad de la luz, $c = 300000 Km/s$, y λ es la longitud de onda de la partícula.

- QBER:

$$QBER = QBER_{opt} + QBER_{det} \tag{A.5}$$

  - QBER óptico

$$QBER_{opt} = \frac{1 - V}{2}$$

    Donde V es la visibilidad de la fibra.
  - QBER de detección:

$$QBER_{det} = \frac{p_{dark}}{p_{det} + 2p_{dark}}$$

    Donde p_{dark} es la probabilidad de obtener un conteo oscuro, dark count, y p_{det} es la probabilidad de obtener una detección.

- Probabilidad de detección (cálculo simplificado[1]):

$$R_{raw}(\delta) = \sum_n p_n \left[1 - (1 - \eta_{det}\eta_\delta)^n\right] \simeq \eta_{det}\eta_\delta\mu$$

Donde η_{det} es la eficiencia de los detectores, η_δ es la atenuación debida a las pérdidas de una fibra de longitud ℓ, $\eta_\delta = 10^{-\delta/10}$, $\delta = \alpha\ell$ [dB], α es el valor de la atenuación por kilómetro de fibra óptica (típicamente oscila entre los 0,25 dB y los 0,3 dB), y μ es el número medio de fotones esperado a la salida de Alice.

- Entropía de Shannon:

$$\begin{aligned} H(X) = H(p_1, p_2, \ldots, p_n) &= \sum_{i=1}^{n} p(x_i) \log\left(\frac{1}{p(x_i)}\right) \\ &= -\sum_{i=1}^{n} p(x_i) \log\left(p(x_i)\right) \end{aligned} \tag{A.6}$$

- Entropía binaria:

$$H(p, 1-p) = -p \log p - (1-p)\log(1-p)$$

- Entropía condicionada:

$$\begin{aligned} H(X|Y) &= \sum_y p(y) H(X|Y=y) \\ &= -\sum_y p(y) \sum_x p(x|y) \log(p(x|y)) \end{aligned}$$

- Entropía de Rényi:

[1] En el artículo de Gilbert y Hamrick [29] encontramos una explicación más detallada de las operaciones necesarias para calcular de forma precisa la probabilidad de detección de una partícula.

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \sum_{i=1}^{n} p(x_i)^\alpha$$

$$H_2(X) = -\log_2 \sum_{i=1}^{n} p(x_i)^2 = -\log_2 P(X = Y)$$

- Entropía de Von Neumann:

$$S(\rho) = -\sum_{j=1}^{k} \lambda_j \log_2 (\lambda_j)$$

- Información mutua:

$$I(X,Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \\ &= I(Y,X) \end{aligned}$$

- Quantum Bit Error, QBER:

$$QBER = \frac{nerrores}{nerrores + n\neg error} \tag{A.7}$$

A.2. Demostraciones

Teorema de no-clonación

Podemos realizar una demostración simple del teorema de no-clonación probando que no existe ninguna operación unitaria, U, que permita la evolución: $\psi \otimes |0\rangle \rightarrow \psi \otimes \psi$, para cualquier estado, ψ. Para ello, partimos del supuesto de que dicha operación unitaria existe, e intentamos clonar los estados $|0\rangle$ y $|1\rangle$, luego existe:

$$U\,|0\rangle \otimes |0\rangle = |0\rangle \otimes |0\rangle$$

$$U\,|1\rangle \otimes |0\rangle = |1\rangle \otimes |1\rangle$$

Y puesto que trabajamos sobre un sistema lineal, tenemos que:

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

Que no coincide con el resultado esperado de una clonación, $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

Apéndice B

Referencias

B.1. Referencias externas

B.1.1. IPsec

Primera generación

RFCs 1825-1829, published in 1995.

RFC-1825 Security Architecture for the Internet Protocol (Obsoleto por: RFC2401)

Segunda generación

In 1998, these documents were obsoleted by RFCs 2401-2412. 2401?2412 are not compatible with 1825-1829, although they are conceptually identical.

RFC-2401 Security Architecture for the Internet Protocol (Obsoleto por: RFC4301)

RFC-2402 IP Authentication Header (AH)

RFC-2406 IP Encapsulating Security Payload (ESP)

RFC-2407 The Internet IP Security Domain of Interpretation for ISAKMP (Obsoleto por: RFC4306)

RFC-2408 Internet Security Association and Key Management Protocol (ISAKMP)

RFC-2409 The Internet Key Exchange (IKE)

RFC-2411 IP Security Document Roadmap

Tercera generación

In December 2005, third-generation documents, RFCs 4301-4309.

RFC-4301 Security Architecture for the Internet Protocol

RFC-4308 Cryptography Suites for IPsec

DRAFT IKEv2 Protocol (Obsoletes: 2407, 2408 & 2409)

B.2. Aclaraciones sobre la distribución cuántica de claves

- La utilización de un término tan impreciso como el de "*criptografía cuántica*", puede llevarnos a pensar en su aplicación como un sustituto de la criptografía convencional. Esto es un error habitual, sobre todo en aquellos entornos que no están familiarizados con los temas aquí tratados, y es la razón por la que justificamos la utilización de otro tipo de denominaciones, como puede ser la "*distribución cuántica de claves*".
- La distribución cuántica de claves, como su nombre indica, es un sistema de distribución de claves basado en los principios de una disciplina de la física conocida como mecánica cuántica. Pero, no es un sistema de distribución o cifrado de la información.
- La distribución cuántica de claves no es una solución final para los sistemas de cifrado, pero sí es seguramente la mejor opción para la distribución segura de claves.

B.3. ITU-T Recommendation G.694.1

Frecuencias recomendadas para los canales de un sistema DWDM con una separación por canal de 100 GHz y 50 GHz, en las bandas L, C y S. Información extraida de: International Telecommunication Union, ITU `http://www.itu.int/`.

	L-Band			
	100 GHz Grid		50 GHz Offset	
	THz	nm	THz	nm
1	186.00	1611.79	186.05	1611.35
2	186.10	1610.92	186.15	1610.49
3	186.20	1610.06	186.25	1609.62
4	186.30	1609.19	186.35	1608.76
5	186.40	1608.33	186.45	1607.90
6	186.50	1607.47	186.55	1607.04
7	186.60	1606.60	186.65	1606.17
8	186.70	1605.74	186.75	1605.31
9	186.80	1604.88	186.85	1604.46
10	186.90	1604.03	186.95	1603.60
11	187.00	1603.17	187.05	1602.74
12	187.10	1602.31	187.15	1601.88
13	187.20	1601.46	187.25	1601.03
14	187.30	1600.60	187.35	1600.17
15	187.40	1599.75	187.45	1599.32
16	187.50	1598.89	187.55	1598.47
17	187.60	1598.04	187.65	1597.62
18	187.70	1597.19	187.75	1596.76
19	187.80	1596.34	187.85	1595.91
20	187.90	1595.49	187.95	1595.06
21	188.00	1594.64	188.05	1594.22
22	188.10	1593.79	188.15	1593.37
23	188.20	1592.95	188.25	1592.52
24	188.30	1592.10	188.35	1591.68
25	188.40	1591.26	188.45	1590.83

Tabla B.1: L-Band 1-25.

	L-Band			
	100 GHz Grid		50 GHz Offset	
	THz	nm	THz	nm
26	188.50	1590.41	188.55	1589.99
27	188.60	1589.57	188.65	1589.15
28	188.70	1588.73	188.75	1588.30
29	188.80	1587.88	188.85	1587.46
30	188.90	1587.04	188.95	1586.62
31	189.00	1586.20	189.05	1585.78
32	189.10	1585.36	189.15	1584.95
33	189.20	1584.53	189.25	1584.11
34	189.30	1583.69	189.35	1583.27
35	189.40	1582.85	189.45	1582.44
36	189.50	1582.02	189.55	1581.60
37	189.60	1581.18	189.65	1580.77
38	189.70	1580.35	189.75	1579.93
39	189.80	1579.52	189.85	1579.10
40	189.90	1578.69	189.95	1578.27
41	190.00	1577.86	190.05	1577.44
42	190.10	1577.03	190.15	1576.61
43	190.20	1576.20	190.25	1575.78
44	190.30	1575.37	190.35	1574.95
45	190.40	1574.54	190.45	1574.13
46	190.50	1573.71	190.55	1573.30
47	190.60	1572.89	190.65	1572.48
48	190.70	1572.06	190.75	1571.65
49	190.80	1571.24	190.85	1570.83
50	190.90	1570.42	190.95	1570.01

Tabla B.2: L-Band 26-50.

	C-Band			
	100 GHz Grid		50 Grid Offset	
	THz	nm	THz	nm
1	191.00	1569.59	191.05	1569.18
2	191.10	1568.77	191.15	1568.36
3	191.20	1567.95	191.25	1567.54
4	191.30	1567.13	191.35	1566.72
5	191.40	1566.31	191.45	1565.90
6	191.50	1565.50	191.55	1565.09
7	191.60	1564.68	191.65	1564.27
8	191.70	1563.86	191.75	1563.45
9	191.80	1563.05	191.85	1562.64
10	191.90	1562.23	191.95	1561.83
11	192.00	1561.42	192.05	1561.01
12	192.10	1560.61	192.15	1560.20
13	192.20	1559.79	192.25	1559.39
14	192.30	1558.98	192.35	1558.58
15	192.40	1558.17	192.45	1557.77
16	192.50	1557.36	192.55	1556.96
17	192.60	1556.55	192.65	1556.15
18	192.70	1555.75	192.75	1555.34
19	192.80	1554.94	192.85	1554.54
20	192.90	1554.13	192.95	1553.73
21	193.00	1553.33	193.05	1552.93
22	193.10	1552.52	193.15	1552.12
23	193.20	1551.72	193.25	1551.32
24	193.30	1550.92	193.35	1550.52
25	193.40	1550.12	193.45	1549.72

Tabla B.3: C-Band 1-25.

	C-Band			
	100 GHz Grid		50 Grid Offset	
	THz	nm	THz	nm
26	193.50	1549.32	193.55	1548.91
27	193.60	1548.51	193.65	1548.11
28	193.70	1547.72	193.75	1547.32
29	193.80	1546.92	193.85	1546.52
30	193.90	1546.12	193.95	1545.72
31	194.00	1545.32	194.05	1544.92
32	194.10	1544.53	194.15	1544.13
33	194.20	1543.73	194.25	1543.33
34	194.30	1542.94	194.35	1542.54
35	194.40	1542.14	194.45	1541.75
36	194.50	1541.35	194.55	1540.95
37	194.60	1540.56	194.65	1540.16
38	194.70	1539.77	194.75	1539.37
39	194.80	1538.98	194.85	1538.58
40	194.90	1538.19	194.95	1537.79
41	195.00	1537.40	195.05	1537.00
42	195.10	1536.61	195.15	1536.22
43	195.20	1535.82	195.25	1535.43
44	195.30	1535.04	195.35	1534.64
45	195.40	1534.25	195.45	1533.86
46	195.50	1533.47	195.55	1533.07
47	195.60	1532.68	195.65	1532.29
48	195.70	1531.90	195.75	1531.51
49	195.80	1531.12	195.85	1530.72
50	195.90	1530.33	195.95	1529.94

Tabla B.4: C-Band 26-50.

	S-Band			
	100 GHz Grid		50 GHz Offset	
	THz	nm	THz	nm
1	196.00	1529.55	196.05	1529.16
2	196.10	1528.77	196.15	1528.38
3	196.20	1527.99	196.25	1527.60
4	196.30	1527.22	196.35	1526.83
5	196.40	1526.44	196.45	1526.05
6	196.50	1525.66	196.55	1525.27
7	196.60	1524.89	196.65	1524.50
8	196.70	1524.11	196.75	1523.72
9	196.80	1523.34	196.85	1522.95
10	196.90	1522.56	196.95	1522.18
11	197.00	1521.79	197.05	1521.40
12	197.10	1521.02	197.15	1520.63
13	197.20	1520.25	197.25	1519.86
14	197.30	1519.48	197.35	1519.09
15	197.40	1518.71	197.45	1518.32
16	197.50	1517.94	197.55	1517.55
17	197.60	1517.17	197.65	1516.78
18	197.70	1516.40	197.75	1516.02
19	197.80	1515.63	197.85	1515.25
20	197.90	1514.87	197.95	1514.49
21	198.00	1514.10	198.05	1513.72
22	198.10	1513.34	198.15	1512.96
23	198.20	1512.58	198.25	1512.19
24	198.30	1511.81	198.35	1511.43
25	198.40	1511.05	198.45	1510.67

Tabla B.5: S-Band 1-25.

	S-Band			
	100 GHz Grid		50 GHz Offset	
	THz	nm	THz	nm
26	198.50	1510.29	198.55	1509.91
27	198.60	1509.53	198.65	1509.15
28	198.70	1508.77	198.75	1508.39
29	198.80	1508.01	198.85	1507.63
30	198.90	1507.25	198.95	1506.87
31	199.00	1506.49	199.05	1506.12
32	199.10	1505.74	199.15	1505.36
33	199.20	1504.98	199.25	1504.60
34	199.30	1504.23	199.35	1503.85
35	199.40	1503.47	199.45	1503.10
36	199.50	1502.72	199.55	1502.34
37	199.60	1501.97	199.65	1501.59
38	199.70	1501.21	199.75	1500.84
39	199.80	1500.46	199.85	1500.09
40	199.90	1499.71	199.95	1499.34
41	200.00	1498.96	200.05	1498.59
42	200.10	1498.21	200.15	1497.84
43	200.20	1497.46	200.25	1497.09
44	200.30	1496.72	200.35	1496.34
45	200.40	1495.97	200.45	1495.60
46	200.50	1495.22	200.55	1494.85
47	200.60	1494.48	200.65	1494.11
48	200.70	1493.73	200.75	1493.36
49	200.80	1492.99	200.85	1492.62
50	200.90	1492.25	200.95	1491.88

Tabla B.6: S-Band 26-50.

Bibliografía

[1] C. E. Shannon, *"Communication theory of secrecy systems"*, Bell System Technical Journal, Vol. 28, 1949.

[2] A. Rényi, *"On measures of entropy and information"*, Proceedings of 4th Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, 1961.

[3] J. L. Carter and M. N. Wegman, *"Universal Classes of Hash Functions (extended abstract)"*, Proceedings of the 9th Annual ACM Symposium of Computing, pp. 106-112, 1977.

[4] M. N. Wegman and J. L. Carter *"New hash functions and their use in authentication and set equality"*, Journal of Computer and System Sciences, Vol. 22, 1981.

[5] S. Wiesner, *"Conjugate Coding"*, Sigact News, Vol. 15, No. 1, 1983.

[6] C. H. Bennett and G. Brassard, *"Quantum cryptography: plubic key distribution and coin tossing"*, Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, IEEE Press., pp. 175-179, 1984.

[7] C. H. Bennett, G. Brassard and J.-M. Robert *"How to Reduce your Enemy's Information (extended abstract)"*, Lecture Notes in Computer Science, Vol. 218, pp. 468-476, 1985.

[8] D. Zuckerman, *"How to recycle random bits"*, Proceedings of 30th IEEE Symposium on Foundations of Computer Science, 1989.

[9] C. H. Bennett, G. Brassard and J.-M. Robert, *"Privacy Amplification by Public Discussion"*, SIAM Journal on Computing, Vol. 17, No. 2, 1988.

[10] A. K. Ekert, *"Quantum Cryptography Based on Bell's Theorem"*, Phys. Rev. Lett., Vol. 67, Is. 6, pp. 661-663, 1991.

[11] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail and J. Smolin, *"Experimental Quantum Cryptography"*, Journal of Cryptography, Vol. 5, Is. 1, Eurocrypt '90, 1992.

[12] C. H. Bennett, G. Brassard and N. D. Mermin, *"Quantum Cryptography without Bell's Theorem"*, Phys. Rev. Lett., Vol. 68, pp. 557, 1992.

[13] C. H. Bennett, *"Quantum Cryptography Using any Two Nonorthogonal States"*, Phys. Rev. Lett., Vol. 68, No. 21, pp. 3121, 1992.

[14] C. H. Bennett, G. Brassard, C. Crépeau and M. H. Skubiszewska, *"Practical Quantum Oblivious Transfer"*, Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '91), Lecture Notes in Computer Science, Vol. 576, pp. 351-366, 1992.

[15] G. Brassard, *"Cryptology column – Quantum cryptography: A bibliography"*, Sigact News, Vol. 24, No. 3, pp. 16-20, 1993.

[16] G. Brassard, *"A bibliography of quantum cryptography"*, 1993.

[17] H. Krawczyk, *"LFSR-based Hashing and Authentication"*, IBM T.J. Watson Research Center, 1994.

[18] G. Brassard and L. Salvail, *"Secret-Key Reconciliation by Public Discussion"*, Lecture Notes in Computer Science, Vol. 765, pp. 411-423, 1994.

[19] C. H. Bennett, G. Brassard, C. Crépeau and U. M. Maurer, *"Generalized Privacy Amplification"*, IEEE Transactions on Information Theory, Vol. 41, No. 6, 1995.

[20] G. Brassard and C. Crepéau, *"Cryptology Column – 25 Years of Quantum Cryptography"*, Sigact News, Vol. 27, No. 3, pp. 13-24, 1996.

[21] C. A. Fuchs and A. Peres, *"Quantum-state disturbance versus information gain: Uncertainty relations for quantum information"*, Phys. Rev. A, Vol. 53, pp. 2038-2045, 1996.

[22] C. A. Fuchs, N. Gisin, R. B. Griffiths, C.-S. Niu and A. Peres, *"Optimal eavesdropping in quantum cryptography. I. Information bound and optimal strategy"*, Phys. Rev. A, Vol. 56, pp. 1163-1172, arXiv:quant-ph/9701039 `http://arxiv.org/pdf/quant-ph/9701039`, 1997.

[23] J. I. Cirac and N. Gisin, *"Coherent eavesdropping strategies for the four state quantum cryptography protocol"*, Phys. Lett. A, Vol. 229, 1997.

[24] C. Cachin and U. M. Maurer *"Linking Information Reconciliation and Privacy Amplification"*, Journal of Cryptology, 1997.

[25] B. A. Slutsky, R. Rao, P.-C. Sun, L. Tancevski and S. Fainman, *"Defense frontier analysis of quantum cryptographic systems"*, Appl. Opp., Vol. 37, No. 14, pp. 2869-2878, 1998.

[26] B. A. Slustky, R. Rao, P.-C. Sun and Y. Fainman, *"Security of quantum cryptography against individual attacks"*, Phys. Rev. A, Vol. 57, No. 4, 1998.

[27] P. W. Shor and J. Preskill, *"Simple proof of security of the BB84 Quantum Key Distribution Protocol"*, Phys. Rev. Lett., Vol. 85, No. 2, pp. 441-444, 2000.

[28] K. Yamazaki and T. Sugimoto, *"A Study on Secret Key Reconciliation Protocol 'Cascade"'*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E83-A, No. 10, pp. 1987-1991, 2000.

[29] G. Gilbert and M. Hamrick, *"Practical Quantum Cryptography: A Comprehensive Analisys (Part One)"*, arXiv:quant-ph/0009027 `http://arxiv.org/pdf/quant-ph/0009027`, 2000.

[30] N. Gisin, G. Ribordy, W. Tittel and H. Zbinden, *"Quantum Cryptography"*, Rev. Mod. Phys., Vol. 74, pp. 145, arXiv:quant-ph/0101098 `http://arxiv.org/pdf/quant-ph/0101098`, 2001.

[31] P. M. Nielsen, C. Schori, J. L. Sorensen, L. Salvail, I. Damgard and E. Polzik, *"Experimental quantum key distribution with proven security against realistic attacks"*, Journal of Modern Optics, Vol. 48, No. 13, 2001.

[32] C. Elliott, *"Building the quantum network"*, New J. Phys., Vol. 4, 2002.

[33] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel and C. G. Peterson. *"Fast, efficient error reconciliation for quantum cryptography"*, Phys. Rev. A, Vol. 67, Is. 5, arXiv:quant-ph/0203096 `http://arxiv.org/pdf/quant-ph/0203096`, 2002.

[34] W.-Y. Hwang, *"Quantum Key Distribution with High Loss: Toward Global Secure Communication"*, Phys. Rev. Lett., Vol. 91, No. 5, 2003.

[35] Shengli Liu, Henk C. A. Van Tilborg and Marten Van Dijk, *"A Practical Protocol for Advantage Distillation and Information Reconciliation"*, Designs, Codes and Cryptography, Vol. 30, Is. 1, pp. 39-62, 2003.

[36] C. Elliott, D. Pearson and G. Toxel, *"Quantum Cryptography in Practice"*, SIGCOMM'03, arXiv:quant-ph/0307049 `http://arxiv.org/pdf/quant-ph/0307049`, 2003.

[37] U. Maurer and S. Wolf, *"Secret-Key Agreement Over Unauthenticated Public Channles - Part I: Definitions and a Completeness Result"*, IEEE Transactions on Information Theory, Vol. 49, No. 4, 2003.

[38] U. Maurer and S. Wolf, *"Secret-Key Agreement Over Unauthenticated Public Channles - Part II: The Simulatability Condition"*, IEEE Transactions on Information Theory, Vol. 49, No. 4, 2003.

[39] U. Maurer and S. Wolf, *"Secret-Key Agreement Over Unauthenticated Public Channles - Part III: Privacy Amplification"*, IEEE Transactions on Information Theory, Vol. 49, No. 4, 2003.

[40] N. Doraswamy and D. Harkins, "*IPsec: The New Security Standard for the Internet, Intranets and Virtual Private Networks (second edition)*", Prentice Hall, ISBN 0-13-046189-X, 2003.

[41] A. Nakassis, J. C. Bienfang and C. J. Williams, *"Expeditious Reconciliation for Practical Quantum Key Distribution"*, Proceedings of the SPIE, Vol. 5436, pp. 28-35, Quantum information and computation II, 2004.

[42] V. Scarani, A. Acín, G. Ribordy and N. Gisin, *"Quantum Cryptography Protocols Robust against Photon Number Splitting Attacks for Weak Laser Pulse Implementations"*, Phys. Rev. Lett, Vol. 92, arXiv:quant-ph/0211131 `http://arxiv.org/pdf/quant-ph/0211131`, 2004.

[43] J. M. Myers, T. T. Wu and D. S. Pearson, *"Entropy estimates for individual attacks on the BB84 protocol for quantum key distribution"*, Proceedings of SPIE, Vol. 5436, Quantum Information and Computation II, pp. 36-47, 2004.

[44] D. Gottesman, Hoi-Kwong Lo, N. Lütkenhaus and J. Preskill, *"Security of Quantum Key Distribution with Imperfect Devices"*, IEEE International Symposium on Information Theory, ISIT 2004, pp. 136, arXiv:quant-ph/0212066 `http://arxiv.org/pdf/quant-ph/0212066`, 2004.

[45] C. Elliott, *"The DARPA Quantum Network"*, BBN Technologies, arXiv:quant-ph/0412029 `http://arxiv.org/pdf/quant-ph/0412029`, 2004.

[46] C. Elliott, *"Quantum Cryptography in Practice"*, Building the DARPA Quantum Network, BBN Technologies, 2004.

[47] R. Alléaume et al., *"Experimental open air quantum key distribution with a single photon source"*, New J. Phys., arXiv:quant-ph/0402110 `http://arxiv.org/pdf/quant-ph/0402110`, 2004.

[48] C. Elliot, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer and H. Yeh, *"Current status of the DARPA Quantum Network"*, BBN Technologies, arXiv:quant-ph/0503058 `http://arxiv.org/pdf/quant-ph/0503058`, 2005.

[49] D. Pearson, *"Building a QKD Network out of Theories and Devices"*, Building the DARPA Quantum Network, BBN Technologies, 2005.

[50] D. Pearson and C. Elliot, *"On the Optimal Mean Photon Number for Quantum Cryptography"*, arXiv:quant-ph/0403065 `http://arxiv.org/pdf/quant-ph/0403065`, 2005.

[51] C. Branciard, N. Gisin, B. Kraus and V. Scarani, *"Security of two quantum cryptography protocols using the same four qubit states"*, Phys. Rev. A, Vol. 72, Is. 3, Id. 032301, arXiv:quant-ph/0505035 `http://arxiv.org/pdf/quant-ph/0505035`, 2005.

[52] M. A. Sfaxi, S. Ghernaouti-Hélie, G. Ribordy and O. Gay, *"Enhancing IP security by integrating Quantum Key Distribution into communication processes"*, 8th International Conference on Telecommunications (ConTEL 2005), Vol. 1, pp. 35-40, 2005.

[53] M. A. Sfaxi, S. Ghernaouti-Hélie, G. Ribordy and O. Gay ,*"Using Quantum Key Distribution within IPSEC to secure MAN communications"*, IfIP-MAN 2005 Conference Proceeding, 2005.

[54] R. Alléaume et al., *"Topology, architecture and protocols for a Quantum Key Distribution network"*, Workshop on classical and quantum information security (Secoqc), 2005.

[55] Chi-Hang Fred Fung, Kiyoshi Tamaki and Hoi-Kwong Lo, *"On the performance of two protocols: SARG04 and BB84"*, Phys. Rev. A, Vol. 73, Id. 012337, arXiv:quant-ph/0510025 http://arxiv.org/pdf/quant-ph/0510025, 2005.

[56] Jörgen Cederlöf, "Authentication in quantum key growing", Thesis http://www.lysator.liu.se/~jc/mthesis/mthesis.pdf, 2005.

[57] M. Dianati and R. Alléaume, *"Architecture of the Secoqc Quantum Key Distribution network"*, First International Conference on Quantum, Nano, and Micro Technologies (ICQNM'07), arXiv:quant-ph/0610202 http://arxiv.org/pdf/quant-ph/0610202, 2006.

[58] M. Eguchi, M. Hagiwara and H. Imai, *"A Quantum Key Distribution Protocol with Selecting Announced States, Robust against Photon Number Splitting Attacks"*, arXiv:quant-ph/0603066 http://arxiv.org/pdf/quant-ph/0603066, 2006.

[59] P. A. Hiskett, D. Rosenberg, C. G. Peterson, R. J. Hughes, S. Nam, A. E. Lita, A. J. Miller and J. E. Nordholt, *"Long-distance quantum key distribution in optical fibre"*, New J. Phys., Vol. 8, Is. 9, pp. 193, arXiv:quant-ph/0607177 http://arxiv.org/pdf/quant-ph/0607177, 2006.

[60] V. Makarov, A. Anisimov and J. Skaar, "Effects of detector efficiency mismatch on security of quantum cryptosystems", Phys. Rev. A, Vol. 74, Id. 022313, arXiv:quant-ph/0511032v3 http://arxiv.org/pdf/quant-ph/0511032v3, 2006.

[61] V. Makarov and J. Skaar, "Faked states attack using detector efficiency mismatch on SARG04, phase-time, DPSK, and Ekert protocols", arXiv:quant-ph/0702262v3 http://arxiv.org/pdf/quant-ph/0702262v3, 2007.

[62] R, Alléaume et al., *"SECOQC White Paper on Quantum Key Distribution and Cryptography"*, arXiv:quant-ph/0701168, 2007.

[63] Yi Zhao, Chi-Hang Fred Fung, Bing Qi, Christine Chen, Hoi-Kwong Lo, "Quantum hacking: experimental demostration of time-shift attack against practical quantum key distribution systems", arXiv:0704.3253v2 http://arxiv.org/pdf/0704.3253v2, 2008.

[64] J. Cederlöf and J.-A. Larsson, "Security aspects of the Authentication used in Quantum Cryptography", IEEE Transactions on Information Theory, Vol. 54, No. 4, arXiv:quant-ph/0611009v3 http://arxiv.org/pdf/quant-ph/0611009v3, 2008.

[65] RANDOM.ORG The Random Number Service http://www.random.org/files/. *Pregenerated Random Numbers. RANDOM.ORG is a true random*

number service that generates randomness via atmospheric noise. This page contains daily true random numbers generated by RANDOM.ORG in the past.

[66] International Telecommunication Union `http://www.itu.int/`. ITU-T Recommendation G.694.1 (2002). Spectral grids for WDM applications: DWDM frequency grid.

[67] International Telecommunication Union `http://www.itu.int/`. ITU-T Recommendation G.694.2 (2003). Spectral grids for WDM applications: CWDM wavelength grid.

Índice alfabético