VNIVERSIDAD NACIONAL
AVTONOMA DE
MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

IMPLEMENTACIÓN DEL PROTOCOLO DE DISTRIBUCIÓN CUÁNTICA DE CLAVES, PROTOCOLO BB84

TESIS

QUE PARA OPTAR POR EL GRADO DE:

MAESTRA EN CIENCIAS (COMPUTACIÓN)

PRESENTA:
VERÓNICA DÍAZ SÁNCHEZ

TUTOR:
ING. MARIO RODRÍGUEZ MANZANERA
IIMAS

MÉXICO, D. F. MAYO 2015

Universidad Nacional Autónoma de México
UNAM
Biblioteca Central
Dirección General de Bibliotecas de la UNAM

UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

En una meta, en un sueño, en un objetivo, se encuentran personas, momentos, experiencias, lugares, que juntos son la estructura donde en la cima se encuentra ese logro. Quiero agradecer por este logro a esa bonita estructura que la vida me permitió vivir.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por sus programas para el apoyo en el inicio y la conclusión de mis estudios de posgrado.
A la Universidad Nacional Autónoma de México (UNAM), porque me abrió sus puertas y me acogió como su hija académica.
Al Posgrado de Ciencia e Ingeniería de la Computación por la experiencia de formar parte de su programa de estudio, porque me permitió descubrir más de mi persona.
A mis maestros por su motivación y tolerancia. Al personal administrativo por su atención y cuidado.
A mi tutor el Ing. Mario Rodríguez Manzanera por la admirable persona que es. Por motivarme con sus enseñanzas, sus experiencias, sus conocimientos.
A mis sinodales por su tiempo e interés en la mejora de mi trabajo de tesis.
Al Dr. Víctor Manuel Velázquez, coordinador de óptica de la Facultad de Ciencias y encargado del Laboratorio de Óptica Avanzada por su apoyo en permitirme presenciar la demostración de la existencia del fotón como parte de una petición personal para el enriquecimiento de este trabajo de tesis.

A mis compañeros del posgrado porque aprendí mucho de ellos, por las risas, las decepciones, las experiencias, por su amistad, en especial de JPV, gracias por llevarme a los límites y rebasarlos, gracias por su compañía y por las pruebas que pasamos juntos en este posgrado.

A mis padres porque me dieron la vida y me dieron los mejores valores que me permiten ser la persona que soy, por su cariño, esto es para ustedes.
A mi madre Isabel que me acompañó en mis derrotas y que las sentía como si fueran de ella, y a pesar de ello siempre estuvo ahí con las palabras precisas, enseñándome siempre a compartir y ayudar. Aunque no estuvo en esta fase de mi vida, sé que ella me acompañaba. A mi padre que en esta fase de mi vida partió, gracias por su fortaleza y ejemplos, por enseñarme a ser justa y valiente.

A mis hermanos Jorge, Juan Carlos y María Eugenia. Gracias hermanos por ayudarme. Gracias hermana por estar ahí, acompañándome, guiándome, gracias por escucharme y soportarme.
A mis amigos Erasmo M. y Ofelia G., porque a través de la distancia siempre estuvimos juntos, gracias por estar orgullosos de mí.

A Dios, porque a pesar de las situaciones difíciles que he vivido siempre me da fortaleza y salud.

Resumen

El presente trabajo muestra uno de los primeros avances de la criptografía cuántica y es el protocolo BB84. Este protocolo es propuesto para el intercambio seguro de claves entre dos puntos, un emisor y un receptor. La manera de distribuir la clave está fundamentada en las propiedades de la mecánica cuántica y no en funciones matemáticas o algoritmos propuestos como se hace en la criptografía convencional.

El funcionamiento del protocolo BB84 es desarrollado en este trabajo en dos panoramas, el primero es sin la presencia de algún espía y el segundo es con la presencia de algún espía. Dicho espía tiene la finalidad de conseguir la clave que se está distribuyendo, al espía se le considera un intruso en el canal de comunicación.

Para mostrar el funcionamiento del protocolo BB84 este ha sido implementado por medio de código, es decir, en un lenguaje de programación. Esta implementación está dada con las especificaciones de dicho protocolo, puede interpretarse como una simulación ya que el protocolo está funcionando sin el componente cuántico.

El primer desarrollo es con el lenguaje de programación C con una con una librería cuántica, es decir con el uso de LibQuantum. El segundo desarrollo del protocolo BB84 es sin el uso de librerías cuánticas, este desarrollo se hace en C#.

Aunado al funcionamiento del protocolo BB84, se muestran varios resultados, estos resultados se van mostrando conforme el protocolo inicia y proporcionan información al lector de lo que va sucediendo al ir generando una clave. Otro resultado a mostrar es el cálculo de la tasa de error cuántico como una forma de detectar intrusos en la comunicación.

Índice

Agradecimientos ..... **iii**

Resumen ..... **v**

Índice ..... **vii**

Índice de figuras ..... **xi**

Índice de tablas ..... **xii**

1. Introducción **1**

1.1 Antecedentes 1

1.2 Motivación 3

1.3 Objetivo 4

1.4 Metodología 5

1.5 Metas 6

1.6 Relevancia de la Investigación 7

1.7 Estructura de la Tesis 8

2. Estado del Arte **10**

2.1 Historia 10

2.2 Código y Cifrado 13

2.3 La Criptografía como una Ciencia 13

2.4 Cifrado Vernam 14

2.5 Criptografía 16

2.6 Criptoanálisis 16

2.7 Criptología 17

2.8 Esquema de Cifrado 17

2.9 Algoritmos Criptográficos 17
2.10 Criptografía Simétrica 18
2.11 Criptografía Asimétrica 19
2.12 Clasificación de los algoritmos Criptográficos 21
2.13 Canal de Comunicación 22
2.14 Sistema Criptográfico 24
2.15 Sistemas Criptográficos Clásicos 26
2.16 Sistemas Criptográficos Modernos 27
2.17 Protocolo 28
2.18 Protocolo Criptográfico 28
3. Criptografía Cuántica (Protocolos) 30
3.1 Información Clásica 30
3.2 Información Contenida en una Señal 31
3.3 Computación Clásica 31
3.4 Algoritmos Cuánticos 36
3.5 Mecánica Cuántica 37
3.6 La Computadora Cuántica 38
3.7 Información Cuántica 39
3.8 El Qubit 41
3.9 Múltiples Qubits 43
3.10 ¿Cómo utilizar el Qubit? 43
3.11 Puerta Cuántica Hadamard H 46
3.12 Criptografía Cuántica 47
3.13 Detalles sobre la QKD 48
3.14 Codificación de la Polarización 49

3.15 Protocolos de QKD 51

4. Protocolo BB84 53

4.1 Canal cuántico del Protocolo BB84 53

4.2 Canal público del Protocolo BB84 54

4.3 Seguridad del protocolo BB84 55

4.4 Protocolo BB84 transmisión sin Espías 56

4.5 Protocolo BB84 transmisión con Espías 57

4.6 Canal Cuántico con Ruido 59

4.7 Reconciliación de las Bases 60

4.8 ¿Qué sucede con el Protocolo BB84 al detectar un espía? 61

4.9 Estimación de Error 62

4.10 Estrategias de Ataques sobre los Protocolo QKD 63

4.11 Clave Secreta generada por el Protocolo BB84 64

5. Resultados de la Implementación del Protocolo BB84 66

5.1 Ejecución del Protocolo BB84 sin Espías con LibQuantum en C 66

5.1.1 Resultados de la Ejecución del Protocolo BB84 sin espías con LibQuantum en C 69

5.2 Ejecución del Protocolo BB84 sin Espías y con Espías en C# 73

5.2.1 Ejecución del Protocolo BB84 sin Espías en C# 74

5.2.1.1 Resultados de la Ejecución del Protocolo BB84 sin espías en C# 76

5.2.2 Ejecución del Protocolo BB84 con Espías en C# 78

5.2.2.1 Resultados de la Ejecución del Protocolo BB84 con espías en C# 80

5.2.3 Ejecución del Protocolo BB84 con espías en C#, estimación del Error 83

5.2.3.1 Resultados de la Ejecución del Protocolo BB84 espías en C#, estimación del Error 85

6. Resultados Generales 89

Conclusiones **91**

Bibliografía **93**

Acrónimos **97**

Índice de figuras

Figura 2. 1: La escítala Espartana. 11
Figura 2. 2: Esquema de cifrado con clave secreta con un canal seguro de comunicación para intercambiar la clave. 19
Figura 2. 3: Esquema de cifrado con clave pública. 20
Figura 2. 4: Canal clásico de información. 23
Figura 2. 5: Clasificación de los sistemas criptográficos moderno. 27

Figura 3. 1: Transformada de Hadamard $H^{\otimes 2}$ operando con dos qubits en paralelo. 46
Figura 3. 2: Representación del sistema QKD. 49
Figura 3. 3: Representación de la Polarización de fotones. 50
Figura 3. 4: Grados de polarización de los fotones, interpretación bit lógico. 51

Figura 4. 1: Representación del protocolo de distribución de claves cuánticas BB84 56
Figura 4. 2: Escenario para un sistema QKD (Quantum Key Distribution) con Eve interceptado en el canal de transmisión cuántico. 60
Figura 4. 3: Resultado de la reconciliación de las bases (sifting stage) en el protocolo BB84 sin espías. 61
Figura 4. 4: Tamaños de clave mínimos recomendados por varias instituciones como la NIST en el año 2014. 65

Figura 5. 1: Envío de 8 qubits, protocolo BB84 con LibQuantum en C. 68
Figura 5. 2: Clave generada del envío de 8 qubits, protocolo BB84 con LibQuantum en C. 69
Figura 5. 3: Gráfica de porcentaje útil de la clave inicial para obtener la clave final con el protocolo BB84 comparando la implementación en LibQuantum y C# (sin espías). 73
Figura 5. 4: Envío de 128 qubits, protocolo BB84 sin espías, implementación en C#. 75
Figura 5. 5: Reconciliación de las bases, protocolo BB84 sin espías, implementación en C#. 75
Figura 5. 6: Clave generada con el Protocolo BB84 sin espías, implementación en C#. .. 76
Figura 5. 7: Clave generada con el Protocolo BB84 con espías, implementación en C#. . 79

Figura 5. 8: Reconciliación de las bases donde con bases iguales se tienen estados diferentes, estas mediciones no son parte de la clave final del envío de 128 qubits, implementación del protocolo BB84 sin espías en C#..... 80
Figura 5. 9: Resultado de la reconciliación de las bases (*sifting stage*) en el protocolo BB84 con espías. 82
Figura 5. 10: Resultado de los qubits perdidos, implementación del protocolo BB84 en C# con espías y sin espías..... 83
Figura 5. 11: El valor de la porción revelada igual a cero para el cálculo de la tasa de error cuántico (QBER), ejecución con 16 qubits de clave inicial, implementación en C#. 84
Figura 5. 12: Tamaño del bloque revelado para el cálculo de la tasa de error cuántico (QBER), ejecución con 64 qubits de clave inicial, implementación en C#. 85
Figura 5. 13: La tasa de error cuántico (QBER) con valor 0 %, una simulación con 64 qubits, implementación en C#. 87

Índice de tablas

Tabla 2. 1: Código Boudot (cifrado de Vernam). 14
Tabla 2. 2: Función XOR y Pad. 15
Tabla 2. 3: Personajes en los protocolos criptográficos. 29

Tabla 4. 1: Nomenclatura del estado del qubit con la base y polarización de un fotón. 54

Tabla 5. 1: Algunos simuladores de Computación Cuántica. 66
Tabla 5. 2: Resultado del promedio para diez simulaciones de una clave inicial de 8 qubits, con el protocolo BB84 sin espías implementado con LibQuantum en C..... 70
Tabla 5. 3: Resultado de las longitudes de las claves finales de una simulación con el protocolo BB84 sin espías implementado con LibQuantum en C. 70
Tabla 5. 4: Resultado del promedio para diez simulaciones de una clave inicial de 256 qubits, con el protocolo BB84 sin espías implementado con LibQuantum en C. 71
Tabla 5. 5: Resultado del promedio para diez simulaciones de una clave inicial de 4096 qubits, con el protocolo BB84 sin espías implementado con LibQuantum en C. 72

Tabla 5. 6: Nomenclatura del estado del qubit --base y polarización de un fotón-- para la ejecución del protocolo BB84 sin espías y con espías implementado en C#. 74
Tabla 5. 7: Resultado de las claves generadas (diez simulaciones) de una clave inicial de 128 qubits con el protocolo BB84 sin espías, implementación en C#. 77
Tabla 5. 8: Resultado de Claves generadas iguales de una clave inicial de 8 qubits con el protocolo BB84 sin espías, implementación en C#. 78
Tabla 5. 9: Resultados de los promedios del porcentaje de clave útil (con espías y sin espías) con el protocolo BB84 con una clave inicial de 128 qubits, implementación en C#. 81
Tabla 5. 10: Resultado del promedio de los qubits perdidos con una clave inicial de 128 qubits en la simulación del protocolo BB84, implementación en C# sin espías y con espías. 83
Tabla 5. 11: Resultado de las longitudes de las claves finales calculando QBER, de una simulación con el protocolo BB84 con espías implementado con C#. 85
Tabla 5. 12: Resultado de la comparación de las longitudes de las claves finales sin espías y calculando QBER con una clave inicial de 512 qubits, de una simulación con el protocolo BB84 con espías implementado con C#. 86
Tabla 5. 13: Valor de la tasa de error cuántico (QBER) para diferentes claves iniciales, implementación en C#. 88

Capítulo 1

Introducción

La información puede decirse que representa poder, la información hay que compartirla con más individuos si de lo que se trata es de aprovechar al máximo su uso. El autor de cierta información (emisor) debe transferirla a algún soporte por ejemplo correo electrónico y hacer llegar ese soporte a su destino (receptor) mediante algún canal de comunicación (mensajería, correo electrónico etc.). En dicho canal pueden estar a la espera terceras personas con intención de interceptar la información. Es imposible asegurar que nunca conseguirán dicha información, la esperanza es que, aunque tengan la información y puedan leerla, no entiendan nada porque el contenido estará cifrado, aquí es donde la criptografía ayuda.

Hoy en día las conversaciones utilizan redes compartidas, para las empresas ya no es suficiente con que las operadoras de telecomunicaciones ofrezcan confianza con protocolos seguros, es por eso que se aplica cifrado en todas partes.

La criptografía consiste tomar un documento original y aplicarle un algoritmo cuyo resultado es un nuevo documento, ese documento está cifrado. Enteramente la seguridad de dicho cifrado no se basa en el propio algoritmo, porque este puede ser de conocimiento público o robado, además el algoritmo puede ser modificado. La seguridad del cifrado se basa enteramente en la clave. Ahora bien en la actualidad se tiene la criptografía simétrica y asimétrica que son parte de la criptografía moderna.

1.1 Antecedentes

El problema principal de la criptografía simétrica es la distribución de las claves, el emisor y el receptor deben tener la misma clave para cifrar y descifrar, entonces

deben asegurarse que tengan la clave correcta. De esto resulta que por el canal donde se envía información no se pueden distribuir las claves, ya que es un canal inseguro. Entonces hay que utilizar un segundo canal de comunicación que tendría que protegerse. Otro problema de la criptografía simétrica es la gestión de las claves almacenadas, asumiendo que una clave separada es usada por cada par de usuarios en la red, el número total de claves crece rápidamente como crece el número de usuarios. Una red de n usuarios requiere $n(n-1)/2$ claves, además si un usuario utiliza t claves necesita entonces t canales seguros para actualizarlas. Este problema puede minimizarse si se mantiene un número reducido de usuarios, pero se sabe que esto no es posible porque una red de usuarios siempre está creciendo.

La criptografía asimétrica cifra utilizando dos claves, lo que se cifra con una se descifra con la otra, entonces comparando con la criptografía simétrica el receptor tendrá dos claves, una clave que es publica y conocida para cifrar la información y una clave privada para descifrar la información, entonces ya no es necesario buscar canales protegidos para enviarla porque la clave privada no es divulgada y no se comparte.

Ahora bien, por lo anterior, ¿Qué métodos de cifrado se deben utilizar, simétrico o asimétrico?, ¿porque? De momento pudiera considerarse que el cifrado asimétrico es una mejor alternativa frente al cifrado simétrico debido al mejor soporte en el intercambio de claves. Con el uso del cifrado asimétrico un usuario solo tiene que proporcionar su clave pública al otro usuario que desea enviarle información. Entonces espera el mensaje cifrado con su clave pública y lo descifra con su clave privada.

El cifrado asimétrico tiene un inconveniente, los algoritmos asimétricos son poco eficientes en tiempo ya que tardan bastante en aplicar las claves para generar información cifrada y descifrada, esto es porque las claves deben ser largas para asegurar la independencia matemática entre ellas. Esta limitación del cifrado asimétrico lo convierte en una estrategia no muy viable en el cifrado de grandes volúmenes de información.

En la actualidad un uso muy común es utilizar el cifrado asimétrico para intercambiar una clave privada entre dos usuarios. Es común nombrar a esa clave como clave de sesión y es usada para cifrar toda la comunicación por medio de un cifrado simétrico, mejorando así el rendimiento por cifrado asimétrico para grandes volúmenes de información. Un ejemplo de esta práctica con cifrado simétrico y asimétrico se menciona a continuación.

Hay que proteger la clave privada, cualquiera con permisos de administrador puede llegar a ella si se encuentra almacenada por ejemplo en un archivo, es entonces que archivos que guardan claves privadas se les denomina *keyring* [1] o archivo llavero que está protegido mediante cifrado simétrico, entonces se repite la historia, para utilizar la clave privada se debe introducir la clave que descifra el llavero y esto implica que se distribuya una clave de cifrado simétrico. También se debe transportar la clave privada, en el cifrado simétrico es sencillo recordar la clave que cifra y descifra, pero en la clave privada hay que recordar cientos de símbolos entonces si el archivo llavero se pierde se puede dar un ataque de fuerza bruta contra el cifrado simétrico.

La criptografía moderna con referencia a la criptografía cuántica, se basa en especular acerca de la imposibilidad de descomponer eficientemente números enteros grandes en sus factores primos utilizando un algoritmo clásico. El tamaño de los números que se usan en las claves de seguridad han ido aumentando para contrarrestar el creciente poder de cálculo de las computadoras actuales; no obstante los expertos tienen claro que llegará un día en el que se consiga superar esa barrera.

Como resultado al problema de la distribución de claves en la criptografía simétrica surge una alternativa y es la distribución cuántica de claves o QKD (*Quantum Key Distribution*), que es usada en criptografía cuántica.

1.2 Motivación

La realización de este trabajo inicia con la inquietud de conocer si existen nuevas formas o métodos de cifrado de información. Una pregunta formulada que acompañó dicha inquietud fue ¿Además de utilizar algoritmos de cifrado, existirán

algunos otros métodos de cifrado que no empleen cálculos numéricos?, ¿Qué otro método de cifrado puede interpretar información?

1.3 Objetivo

La problemática con el cifrado simétrico es encontrar un método eficiente el cual permita acordar el intercambio de claves de forma seguro [2]. Aunque un canal se defina como seguro éste es susceptible a que un intruso lo intercepte y obtenga la clave. Para después descifrar la información que este en curso por el canal inseguro.

La criptografía cuántica es un conjunto de protocolos de comunicación cuántica y un protocolo importante es el QKD. La distribución cuántica de claves (QKD) se propuso como una solución a los problemas de distribución de claves en los sistemas criptográficos clásicos. La QKD permite que las partes involucradas en una comunicación puedan intercambiar las claves de forma segura, además las claves se pueden utilizar en la criptografía convencional [2].

El principal objetivo de este trabajo es implementar por medio de un lenguaje de programación el funcionamiento de uno de los protocolos QKD que es el protocolo BB84, con la finalidad de exponer como funciona este protocolo de criptografía cuántica conforme a las leyes de la mecánica cuántica y el uso de los *qubits*. Además mostrar como el protocolo BB84 provee un mecanismo que permite a las partes que se están comunicando entre si detectar automáticamente escuchas o intrusos en la comunicación.

No obstante, se pretende también mostrar su funcionamiento en el ámbito de generar y distribuir una clave con la presencia de espías y sin la presencia de espías, de igual forma que sucede con esta clave. De tal manera que en la implementación con código se muestran más resultados del comportamiento de este protocolo. Dichos resultados están relacionados por ejemplo con la longitud de la clave final obtenida en comparación con la clave inicial a distribuir.

En el caso de distribuir una clave con presencia de espías, este trabajo expone la manera de la detección de dicho espía en base al cálculo de la tasa de error cuántico del cual se explica un detalle encontrado.

Otro objetivo en la implementación del protocolo BB84, es introducir al lector en el conocimiento de nuevas formas de cifrado, es decir, la distribución cuántica de claves (QKD) que es un método en el cual estados cuánticos son usados para cifrar, ya no se hace uso del concepto *bit*, ahora se hace uso del concepto del *qubit*. La fortaleza de la criptografía cuántica es que los cifrados que son generados obedecen las leyes de la mecánica cuántica, mas no son implementados con algoritmos numéricos para generar la clave secreta, así como lo hace la criptografía convencional (criptografía moderna).

1.4 Metodología

Como primer paso se inicia la investigación de antecedentes que fundamenten la aparición de una criptografía alterna o paralela a la criptografía moderna que es nombrada como criptografía convencional. A principios de la investigación un concepto se destapa, este concepto está relacionado con la representación de la información y es el concepto del *qubit* (*quantum bit*) o *bit* cuántico. Este concepto contribuyó a formular más preguntas, ¿En qué casos participa un *qubit*?, o mejor aún, involucra a pensar que el *bit* es una representación binaria de 0 y 1 y que se puede implementar de manera eléctrica en un circuito para cero voltios o tierra (0 binario), o $+5$ voltios (1 binario), entonces ¿Qué sucede con el *qubit*? Conforme se avanza en la investigación se arroja un resultado, y el resultado es sobre los temas de la criptografía cuántica y los protocolos para la criptografía cuántica, dichos temas se encontraron en las publicaciones de las referencias [3] y [4]. Estableciendo que cuando se menciona criptografía cuántica de lo que se refiere en realidad es de la distribución cuántica de claves. El primer protocolo de criptografía cuántica es el protocolo BB84 y es un protocolo de distribución cuántica de claves que hace uso del término *qubit*. Abiertas las puertas del conocimiento se continúa el enriquecimiento del tema y se indaga más sobre estos protocolos y su funcionamiento. Es aquí el punto de partida del desarrollo de este trabajo. El protocolo BB84 es el primer protocolo de criptografía cuántica y es la base del surgimiento de nuevas versiones. Otros protocolos fueron consultados y en los

casos encontrados el protocolo BB84 se distingue por ser un protocolo completo en cuanto a que la cifra de información se hace con dos bases, es decir, se tienen cuatros estados posibles de representación de información en comparación con dos estados que presentaron versiones propuestas del protocolo. Además es el primer protocolo de criptografía cuántica.

El inicio del conocimiento de cómo funciona el protocolo BB84, no solo se resume a su funcionamiento, pues varios nuevos conceptos emergen del tema. Conceptos como teoría de la información cuántica, *qubits*, complejidad computacional cuántica, mecánica cuántica, fotones, computación cuántica entre otras. Dichos conceptos fueron abordados con la ayuda del área de la Física.

Como siguiente paso se inicia el estudio para entender cómo funciona este protocolo. Algunos inicios se resumían a corridas escritas en papel, dibujos, videos explicativos, etc.

Madurados los conceptos nuevos, se lleva a cabo la implementación del protocolo BB84 esperando mostrar más resultados que simplemente mostrar su funcionamiento, a continuación se mencionan los resultados a mostrar:

La clave final con cierta longitud, la clave final en binario, la cantidad de *qubits* perdidos y el porcentaje de clave útil y el porcentaje de clave no útil (sin espías).

Así como también para el caso con presencia de espías también es espera mostrar: El porcentaje de estados alterados de la clave inicial al medir el espía (llamado Eve), el número de estados alterados de la clave inicial al medir Eve, la clave final con cierta longitud, cantidad de *qubits* perdidos donde el estado difiere. Con la presencia de espías la clave final se reduce a una longitud, se muestra dicha longitud y el porcentaje de clave útil con espías. Otro resultado a mostrar es el cálculo de la tasa de error cuántico como una forma de detectar intrusos en la comunicación.

1.5 Metas

En el transcurso de este trabajo la primera meta que surge es entender el concepto del *qubit* en contraposición con el *bit*, al cual estamos familiarizados.

De igual forma adentrarse a nuevos conocimientos y conceptos sobre la física moderna y como estos conceptos y conocimientos están siendo usados y aplicados a la criptografía.

Con más formalismo sobre este trabajo de tesis la meta principal es mostrar la implementación del protocolo BB84 por medio de un lenguaje de programación.

Dos propuestas de implementación se contemplan. La primera propuesta es con en el lenguaje de programación C con una librería cuántica, es decir, con el uso de LibQuantum que simula el comportamiento de una computadora cuántica, en concreto el uso de registros cuánticos. Esta primera fase involucra el funcionamiento del protocolo BB84 solamente sin presencia de espías, pero aunado a esta fase se mostraran algunos resultados. La meta en esta fase es la utilización de LibQuantum como primer propuesta de implementación. La segunda propuesta es sin el uso de librerías cuánticas. La meta en esta segunda fase, es la construcción de todo el protocolo con presencia de espía y sin presencia de espías. La construcción de todo el protocolo se hace en C#.

La siguiente meta es originar varios resultados como producto de la implementación del protocolo BB84.

Llevar a cabo el cálculo de más resultados arroja más detalles para la comprensión de dicho protocolo. Con el cálculo de dichos resultados se pretende concluir que sucede con este protocolo.

1.6 Relevancia de la Investigación

El conocimiento de nuevas formas de cifrado abre las puertas a preguntarse ¿Qué está sucediendo en la actualidad con estas nuevas formas de cifrado?

Este trabajo de tesis propone acercar al lector y al público en general al conocimiento de un protocolo propuesto para la criptografía cuántica.

Como contribución al conocimiento de este protocolo se presenta la implementación del mismo.

En la actualidad el implantar los principios asociados a la mecánica cuántica en nuevas formas de cifrado, ha rebasado la idea de teorizar solo en papel. Han surgido empresas y grupos de investigación que están estudiando tan novedosa tecnología

que es la criptografía cuántica. Algunas empresas en este campo son: Toshiba, Hewlett Packard, IBM, Mitsubishi, NEC, NTT, Google.

Ya existen empresas que se dedican a la venta de claves de cifrado cuánticas, como son:

Quintessence Labs su página:http: http://quintessencelabs.com/products-3/ , donde ofrece el producto qOptica^{TM}, que permite seguridad en la distribución de la clave punto a punto basado en la distribución cuántica de claves (*Quantum Key Distribution*).

MagiQ su página: http://www.magiqtech.com/Products.html , donde se puede apreciar por ejemplo el siguiente producto, el sistema de distribución de clave cuántica Q-Box, que hace uso del protocolo BB84.

Id Quantique su página: http://www.idquantique.com/photon-counting/clavis.html, donde ofrece el producto CLAVIS^2, una de sus funciones es intercambiar de forma segura una clave con una distancia de hasta 100 km usando los protocolos BB84 y SARG.

1.7 Estructura de la Tesis

Este trabajo de tesis consta de 5 capítulos. El capítulo 1 introduce al lector de manera general en el tema de la criptografía convencional y la problemática que esta enfrenta en la actualidad. Introduce al lector en una forma de cifrado y distribución de clave alterna a la que es aplicada hoy en día. El capítulo 2 describe los conceptos básicos del cual la criptografía convencional se fundamenta, estos conceptos ayudan al lector a comprender como es que funciona un esquema de cifrado. También se hace una breve mención de los indicios históricos. El capítulo 3 presenta la criptografía cuántica, sus fundamentos y los conceptos que la conforman para terminar con la presentación de un protocolo de distribución de claves cuánticas (QKD). El capítulo 4 presenta el protocolo BB84 como un protocolo QKD (*Quantum Key Distribution*) y como es que el protocolo funciona para su implementación en este trabajo de tesis. El protocolo es presentado en dos escenarios con espías y sin espías espiando en el canal de comunicación. El capítulo 5 y 6 exponen los resultados obtenidos de la implementación del protocolo

BB84 en este trabajo de tesis, así como el análisis de dichos resultados en el comportamiento del protocolo. Por último se presentan las conclusiones a las que llega este trabajo.

Capítulo 2

Estado del Arte

En este capítulo se expone los inicios de la criptografía como un medio de seguridad nacional en acontecimientos importantes como la Primera y Segunda Guerra Mundial. Se mencionan documentos, resultado del desarrollo y avances de nuevas formas de cifrado, así como del registro de su historia.
Se presentan conceptos y esquemas de la criptografía actual, criptografía que conocemos con el uso de las computadoras de hoy en día.

2.1 Historia

La criptografía se considera un arte muy antigua, desde los egipcios se conoce el uso de los jeroglíficos y formas de escribir en códigos secretos.
El primer uso claro de métodos criptográficos se dio durante la guerra entre Atenas y Esparta (siglo V a.C.). Plutarco historiador griego describe la escítala como un palo o bastón en el cual se enrollaba una tira de cuero en espiral, al estar enrollada la tira, se escribía un mensaje de forma longitudinal o a lo largo del bastón. La tira desenrollada mostraba un mensaje sin relación al mensaje original escrito. Este procedimiento supone que tanto el emisor como el receptor del mensaje disponen de un palo o bastón con el mismo diámetro y longitud.
La escítala espartana [5] es el primer cifrado de transposición, pues las letras simplemente se cambian de sitio o se transponen. La Figura 2.1 muestra la forma de la escítala.

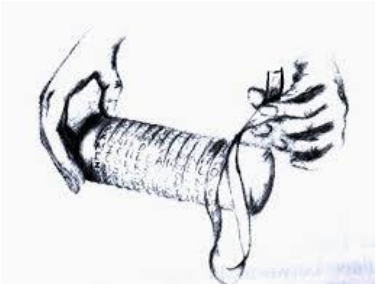


Figura 2. 1: La escítala Espartana.

Importantes desarrollos y avances surgieron en la Primera y Segunda Guerra Mundial. La criptografía fue usada como una herramienta para proteger secretos nacionales y con ella, se abrió la posibilidad de difundir escritos y publicaciones acerca del tema [5].

Con el paso del tiempo y con el fin de la Primera Guerra Mundial, estos escritos públicos también llegaron a su fin. Durante los años 1930 y 1940 aparecieron algunos documentos como literatura abierta y varios tratados acerca del tema fueron publicados. Algunos escritos prevalecieron como el de Claude Shannon, "*The Communication Theory of Secrecy Systems*", publicado en el *Bell System Technical Journal* en 1949 [6].

Después de la Segunda Guerra Mundial, la armada y la naval de Estados Unidos comenzaron a trabajar de manera secreta, empezaron a realizar avances sobre la criptografía.

Desde 1949 hasta 1967 la literatura criptográfica decayó, no fue de gran impacto. Una contribución a la historia de la criptografía fue la Historia de David Khan, "*The Codebreakers: The Story of Secret Writing*" [7]*,* no es una aportación de nuevas técnicas, sino más bien, contiene la más completa historia de la criptografía desde sus inicios, incluyendo información considerada de tipo confidencial para el gobierno.

Por el año de 1974, el *National Bureau of Standards* (NBS) hoy NIST (*National Institute of Standards and Technology*) realizó una convocatoria pública en el Registro Federal de los Estados Unidos para el desarrollo de sistemas criptográficos [8], [10], como resultado de la convocatoria en 1975 IBM presentó un sistema

llamado Lucifer, este sistema fue modificado dando origen a el sistema de cifrado de clave secreta DES (*Data Encryption Standard*).

El 17 de marzo de 1975 fue publicado el algoritmo DES en el registro Federal y en noviembre de 1976 fue aprobado como un estándar federal en términos de seguridad. En enero de 1977 fue publicado como FIPS PUB 46 [9], [10], posteriormente fue confirmado en 1998 con el FIPS 46-3 este último definiendo el Triple DES.

El DES se había revisado cada cinco años, hasta 1998, año en que fue dado de baja como estándar en servicios de seguridad.

En 1998 *Electronic Frontier Foundation* (EFF), ganó la competencia RSA DES Challengue II-2, rompiendo el DES en menos de tres días. La computadora de la EFF fue llamada DES *Cracker*. Al siguiente año en forma conjunta *Deep Crack* y *distributed.net* lograron ganar el concurso RSA DES *Challengue* III, rompiendo el DES en 22 horas y 15 minutos.

En mayo del 2002, DES fue reemplazado finalmente por AES (*Advanced Encryption Standard*), DES fue confirmado como FIPS 46-3 que especifica la preferencia del uso del Triple DES. En el 2001 el algoritmo AES se publica como FIPS 197 [9].

En el año 1976 la criptografía dio un giro. En la Universidad de Stanford se publica el artículo *"New Directions in Criptography"* [10], [11]. Los autores de éste artículo son Whitfield Diffie y Martin E. Hellman. Ellos proponen una nueva clase de sistema criptográfico, en el que las claves para cifrar y descifrar son diferentes y la última no se deriva de la anterior, además de cumplir con ciertos requisitos.

El artículo propone minimizar la necesidad de los canales seguros para distribuir las claves, esto dio paso a los sistemas criptográficos de clave pública. Ya se habla de distribución de claves.

En el año de 1977 un grupo del MIT (*Massachusetts Institute of Technology*) dio a conocer un algoritmo criptográfico llamado RSA. RSA de las siglas de R.L. Rivest, A. Shamir y L. Adleman. El RSA se considera muy robusto, se han hecho intentos por descifrarlo.

Este algoritmo fue el primer sistema criptográfico de clave pública, es uno de los más utilizados tanto para cifrar como para firmar digitalmente y también es usado

para la transmisión segura de datos a través de canales inseguros. Este algoritmo fue patentando por el MIT en 1983 en los Estados Unidos con el número 4.405.829, cuya patente ya expiró en el año 2000.
Si se está interesado en la historia completa de la criptografía se recomienda el libro de David Khan, *"The Codebreakers: The Story of the Secret Writing"* [7].

La organización científica nacional IACR (*Internal Association for Cryptologic Research*) mantiene la investigación en el área de la criptografía, cuya finalidad básica es la seguridad de la información.

2.2 Código y Cifrado

La palabra código alude a un tipo muy particular de comunicación secreta que ha ido cayendo en desuso a lo largo de los siglos. En un código, una palabra o una frase es reemplazada por otra palabra, un símbolo o algún número, debe entenderse que código y cifrado tienen una distinción. El cifrado es una transformación o reemplazo de cada elemento del mensaje original (letra por letra, *bit* por *bit*) no importa su estructura lingüística. En la actualidad, los códigos ya no se utilizan, pero en su época marcaron líneas de gloria como el código en navajo [5].

2.3 La Criptografía como una Ciencia

Fue que gracias a los estudios de Claude Shannon sobre Teoría de la Información en el año de1948 donde sugiere utilizar operaciones que mesclaran sustituciones y transposiciones. La criptografía no solo se consideraba un arte sino que fue reconocida como una ciencia.
La criptografía se divide en criptografía clásica y criptografía moderna.
La criptografía clásica alude a métodos de cifrado antiguos como el cifrado de Vernam [1].

2.4 Cifrado Vernam

En 1917 Gilbert Vernam desarrolló un algoritmo de cifrado que lleva su nombre (cifrado Vernam), pero no fue hasta 1926 donde da a conocer este resultado.

En 1917, durante la Primera Guerra Mundial Gilbert Vernam ingeniero de AT&T, se le encomendó la tarea de inventar un método de cifrado del cual los alemanes no pudieran descifrar [10].

Su cifrado es un cifrado por sustitución binaria basado en el código de Boudot de 5 *bits*. En este cifrado se tiene una única operación con la función XOR cuya notación matemática es $\oplus$. Esta función u operación se usa tanto para cifrar como para descifrar el mensaje.

La Tabla 2.1 donde muestra el cifrado de Vernam con código Boudot. Pueden verse otras formas del cifrado de Vernam en [8] de los cuales se emplea el código ASCII.

Código Binario	Carácter	Código Binario	Carácter
00000	Blanco	10000	T
00001	E	10001	Z
00010	=	10010	L
00011	A	10011	W
00100	Espacio	10100	H
00101	S	10101	Y
00110	I	10110	P
00111	U	10111	Q
01000	<	11000	O
01001	D	11001	B
01010	R	11010	G
01011	J	11011	↑
01100	N	11100	M
01101	F	11101	X
01110	C	11110	V
01111	K	11111	↓

Tabla 2. 1: Código Boudot (cifrado de Vernam).

Clave: ***by*** *11001 10101*
Mensaje: ***VE*** $\oplus$ *11110 00001*

UH → *00111 10100* → *Mensaje cifrado*

Un factor importante considerado en este algoritmo es su simplicidad [13].

El método consiste de un alfabeto binario de ceros y unos, se tiene el mensaje original y se tiene una clave. El cifrado de Vernam se considera un cifrado de Vigenère [5], [10], pero en este caso el mensaje original y la clave es de tipo binario y tienen una longitud similar.

Si la clave es escogida aleatoriamente y no se usa nuevamente el cifrado de Vernam es llamado *one-time system* o *one-time pad* [10], [12]. La clave se actualiza para cada mensaje a cifrar y dicha clave es de igual longitud a la del mensaje.

Claude Shannon en su trabajo, "*The Communication Theory of Secrecy Systems*" [6], demostró que las características anteriores hacen que este cifrado sea perfectamente seguro, dado que no hay forma de efectuar un criptoanálisis [8].

Cuando se habla de la palabra *pad*, se está refiriendo a la clave, *one-time pad* se usa en pares, una copia del *pad* se conserva por cada usuario, en este caso para un emisor y un receptor. La Tabla 2.2 muestra el cifrado Vernam, donde la función XOR realiza una sustitución de 00 y 11 a 0, y de 01 y 10 a 1.

Mensaje original	Pad (clave)	Mensaje cifrado XOR
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 2. 2: Función XOR y Pad.

Los principios esenciales en los que se basa la criptografía clásica son en base a dos tipos de cifrado, la técnica de transposición y la técnica de sustitución.

Los tipos de cifrado fueron divididos así por su historia y son importantes porque son el antecedente de la criptografía moderna.

Algunas opiniones argumentan que la criptografía apareció de manera espontánea después de que la escritura fue inventada más las vastas aplicaciones en los tiempos de guerra.

En el año 1970 nuevas formas de criptografía llegan con el desarrollo generalizado de las computadoras y los sistemas de información, representando para las empresas del sector privado una demanda para proteger la información en medios de formato digital y proporcionar servicios de seguridad.

2.5 Criptografía

La criptografía, es el arte y la ciencia de mantener los mensajes seguros y es practicada por criptógrafos. Una tarea clásica y fundamental de la criptografía es proveer confidencialidad por medio de métodos de cifrado. Entre algunas disciplinas que engloba se encuentran la Teoría de la Información, la Teoría de números o Matemáticas Discretas y la Complejidad Algorítmica.

Una definición formal de criptografía es la siguiente:

La criptografía es el estudio de las técnicas matemáticas relacionados con los aspectos de seguridad de la información, como la confidencialidad, integridad de datos, autenticación de la entidad, y autenticación del origen de datos.

La criptografía no es el único medio para proporcionar seguridad de la información, sino más bien un conjunto de técnicas [13].

Dos trabajos fundamentales sobre los que se apoya prácticamente toda la teoría criptográfica actual (criptografía moderna) son los desarrollados por Claude Shannon en sus artículos *"A Mathematical Theory of Communication"* [14] y *"Communication Theory of Secrecy Systems"* [6]. Shannon sienta las bases de la Teoría de la Información y de la Criptografía moderna. Y el publicado por Whitfield Diffie y Martin Hellman en 1976, se titula *"New directions in Cryptography"* [10], [11], que introduce el concepto de Criptografía de clave Pública, abriendo enormemente el uso de esta disciplina.

2.6 Criptoanálisis

El criptoanálisis busca romper los métodos de cifrado de la criptografía. Un criptoanalista es la persona que practica el criptoanálisis y trata de descifrar los mensajes cifrados para recuperar el texto plano por medio de técnicas matemáticas sin el conocimiento de alguna clave.

2.7 Criptología

La rama de las matemáticas que estudia la criptografía y el criptoanálisis en conjunto es la criptología. Los buenos criptólogos tienen que entrenarse en la teoría matemática.

Los principios de Kerckhoffs [13], deben su nombre al criptógrafo militar holandés Dutchman Auguste Kerckhoffs. Fue educado en Francia, estudió distintos sistemas criptográficos cuyos resultados de su investigación fueron publicados en el artículo titulado *"La Cryptographie Militaire"*. En su artículo menciona 6 reglas que los sistemas criptográficos deben cumplir o que un criptógrafo debe de tener en cuenta para basar la seguridad del mismo [13]. Una suposición fundamental del criptoanálisis primero anunciada por Kerckhoffs en el siglo XIX, es que el secreto debe residir enteramente en la clave [10].

2.8 Esquema de Cifrado

La criptografía involucra dos procesos: el cifrado y el descifrado.

Cifrado: transformación del texto plano (el mensaje) en un texto cifrado o criptograma.

Descifrado: es la transformación del criptograma al texto plano.

Se llama K al conjunto de claves. Un elemento de K es llamado una clave.

Llamemos E_k a la función de cifrado o transformación de cifrado. Si E_k es una correspondencia uno a uno, su inversa debe existir y es D_k. D_k es la función de descifrado o transformación de descifrado.

Al proceso de aplicar la transformción E_k a un mensaje m, se le refiere como cifrado de m o del mensaje.

Al proceso de aplicar la transformacion D_k al texto cifrado c, se le refiere como descrifrado de c o del criptograma [13], [15].

2.9 Algoritmos Criptográficos

Los algoritmos matemáticos son una forma simple de representar un amplio conjunto de transformaciones para cifrar y descifrar.

Un algoritmo criptográfico también es llamado cifrador y es una función matemática, generalmente son dos funciones relacionadas (transformación para cifrar y descifrar un mensaje), se tiene entonces que E y D son los algoritmos de cifrado y descifrado respectivamente.

Un algoritmo puede ser restringido si su seguridad se basa en mantener oculto su funcionamiento, esta practica se considera histórica. Si alguien revela el funcionamiento, los demás tendrán que cambiar su algoritmo, además los algoritmos de tipo restringido son populares en aplicaciones de baja seguridad.

La criptografía moderna resuelve este problema de seguridad con el uso de claves K.

Generalmente los algoritmos criptográficos funcionan empleando claves como un parámetro de los mismos.

2.10 Criptografía Simétrica

Los algoritmos de criptografía simétrica utilizan la misma clave para el proceso de cifrado y descifrado, entonces las funciones quedan como:

$$E_k(m) = c$$

$$D_k(c) = m$$

Donde si $e \in K$ y $d \in K$, sucede que:

e = clave de cifrado d = clave de descifrado

$$e = d = k \text{ y } k \in K.$$

Ambas funciones tienen la propiedad $D_k(E_k(m)) = m$, en la Figura 2.2 se muestra el esquema de cifrado y descifrado donde participa una clave.

La seguridad de los algoritmos simétricos reside en la clave, divulgar la clave significa que cualquiera puede cifrar y descifrar los mensajes [10].

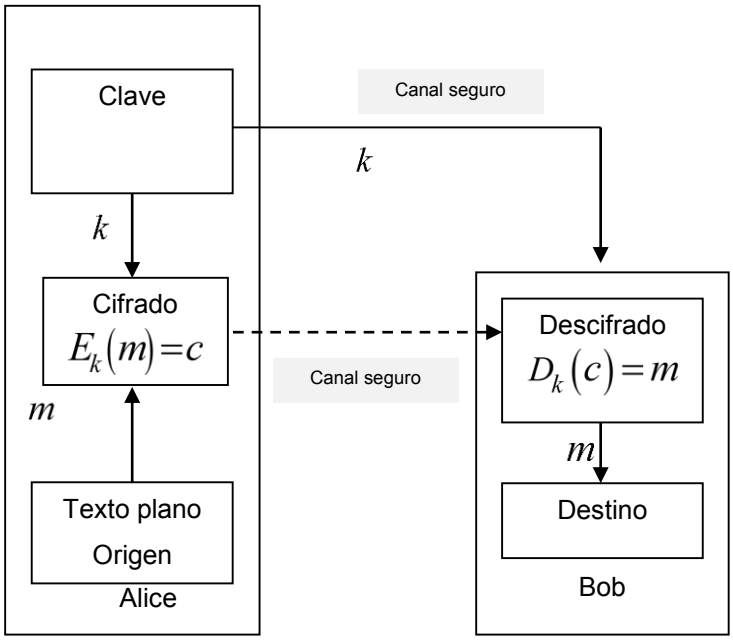


Figura 2. 2: Esquema de cifrado con clave secreta con un canal seguro de comunicación para intercambiar la clave.

Estos algoritmos simétricos son sencillos de utilizar y, en general resultan bastante eficientes (tardan poco tiempo en cifrar y descifrar). Un ejemplo de criptografía simétrica es la autenticación de un móvil GSM [1]. También se pueden cifrar archivos en Windows con la herramienta IZArc, IZArc es un compresor de archivos pero además tiene la opción de cifrar [1].

2.11 Criptografía Asimétrica

Algunos algoritmos usan diferentes claves de cifrado y de descifrado, es decir:

$$E_{k_1}(m)=c$$
$$D_{k_2}(c)=m$$
$$D_{k_2}\left(E_{k_1}(m)\right)=m$$

Donde si $e \in K$ y $d \in K$, sucede que:

e = clave de cifrado $\qquad$ d = clave de descifrado

$$e \neq d$$

La Figura 2.3 muestra el esquema de cifrado asimétrico.

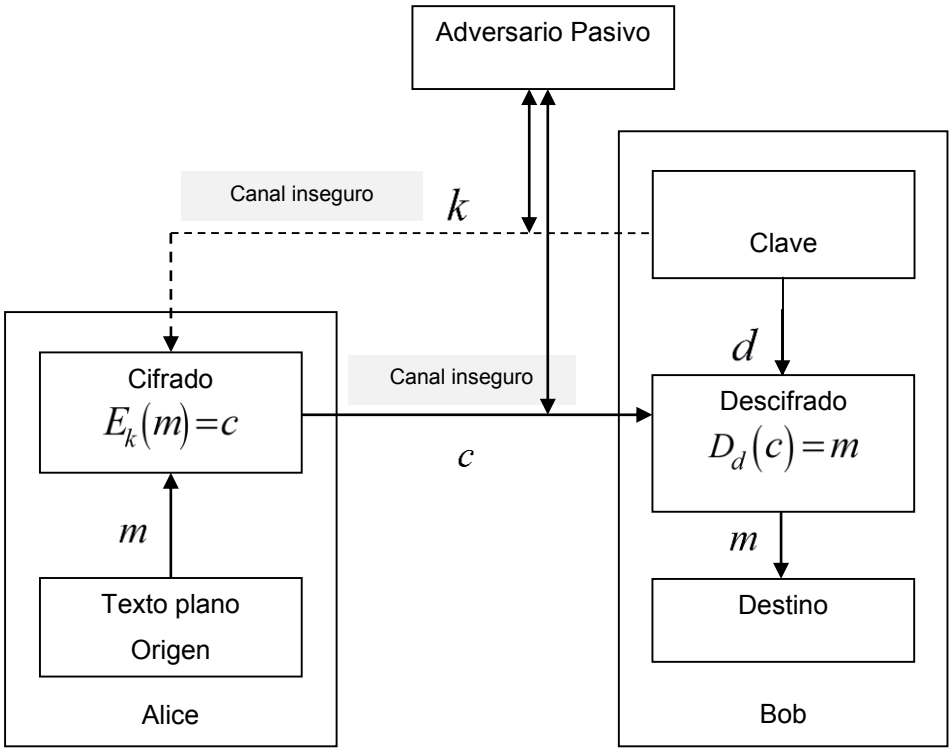


Figura 2. 3: Esquema de cifrado con clave pública.

Los sistemas criptográficos asimétricos tienen la propiedad de que dada la clave k de cifrado, es de manera imposible determinar computacionalmente la clave d correspondiente a la clave de descifrado, para que el esquema sea seguro lo anterior debe cumplirse [10], [13]. Bajo esta suposición, se considera el esquema de comunicación entre dos entidades Alice y Bob. Bob selecciona un par de llaves (k,d). Bob envía por medio de un canal la clave de cifrado k a Alice, esta clave es la clave pública, Bob a su vez conserva la clave d de manera segura y secreta, esta clave es la clave privada.

Alice envía un mensaje m (texto plano) a Bob, aplicando la transformación de cifrado E_k con la clave k de Bob para obtener $c = E_k(m)$. Bob descifra el texto cifrado c aplicando la transformación inversa D_d únicamente determinada por su clave d.

En la Figura 2.3, la clave de cifrado es transmitida por un canal inseguro.

Toda la seguridad en estos algoritmos está basada en la clave o claves, nada está basado en los detalles del propio algoritmo. Esto quiere decir que el algoritmo puede ser publicado y analizado y no es de importancia si algún espía o adversario lo conoce mientras no tenga conocimiento de la clave.

2.12 Clasificación de los algoritmos Criptográficos

Un sistema criptográfico o criptosistema es un algoritmo.

Los algoritmos criptográficos se pueden clasificar en base al número de claves que usan para cifrar y descifrar.

1. Algoritmos Simétricos

   Algunas veces llamados algoritmos convencionales donde la clave de cifrado puede ser calculada a partir de la clave de descifrado. En la mayoría de los algoritmos simétricos la clave de cifrado y descifrado son la misma, estos algoritmos son llamados algoritmos de clave secreta o privada, algoritmos de solo clave, o algoritmos de una clave.

Los siguientes algoritmos son algunos representativos de la criptografía simétrica: **IDEA (International Data Encryption Algorithm)** [10], **BLOWFISH** [10], **RC5 (Rivest Cipher 5)** [10], **DES (Data Encryption Standard)** [10], [12], **3DES o TDES**, **AES (Advanced Encryption Standard)** [8], [12].

2. Algoritmos Asimétricos

   Son llamados algoritmos de clave pública, porque la clave de cifrado puede hacerse pública, una persona desconocida o cualquier persona puede usar la clave para cifrar un mensaje, pero solamente la persona específica con su correspondiente clave puede descifrar dicho mensaje. La clave de descifrado es llamada clave privada y no puede ser calculada a partir de la clave de cifrado.

Algunas veces, el mensaje será cifrado con la clave privada y descifrados con la clave pública, esto es usado en firmas digitales [10], [12], [15].

Los siguientes algoritmos son algunos representativos de la criptografía asimétrica: **Diffie-Hellman** [8], [10], **El Gamal** [8], **RSA (*Rivest Chamir Adelman*)** [8], [10], **Funciones Hash** [8], [10], **Firmas Digitales** [8], [10], [12], **Curvas Elípticas** [8], [10]**.**

2.13 Canal de Comunicación

Los participantes en una comunicación son [13]:

Entidad: alguien o algo el cual envía, recibe o manipula la información. La entidad puede ser una computadora, una persona etc.

Emisor: es una entidad en una comunicación de dos participantes, el cual es el legítimo transmisor de información, pues cifra el mensaje. Al emisor se le denomina Alice.

Receptor: es una entidad en una comunicación de dos participantes, el cual es entendido como el destinatario de la información, pues descifra el mensaje. Al receptor se le denomina Bob.

Adversario: es una entidad en una comunicación de dos participantes, este no es ni emisor ni receptor e intenta sabotear el servicio de seguridad de la información entre el emisor y el receptor. Otros nombres que suelen denominarse al adversario son: enemigo, atacante, oponente, espía, intruso, etc. Un adversario frecuentemente intentará tomar el papel del emisor o el receptor. Al adversario o espía se le denomina Eve.

Los canales de comuniación para la información son [13]:

Canal: el medio para transportar información de una entidad a otra.

Canal Fisico seguro: fisicamente no es accesible por el adversario.

Canal inseguro: cualquier canal en el cual cualquier entidad puede reordenar, borrar, insertar, leer información.

Canal seguro: un canal en el cual un adversario no tiene la capacidad de reordenar, borrar, insertar o leer información.

Existe una ligera diferencia entre un canal físico seguro y un canal seguro, un canal seguro puede ser asegurado por técnicas físicas o criptográficas. En algunos canales se asume que son fisicamente seguros, por ejemplo un canal dedicado para la comunicación. La Figura 2.4 muestra un canal clásico de información. Se observa un canal seguro y un canal inseguro [15].

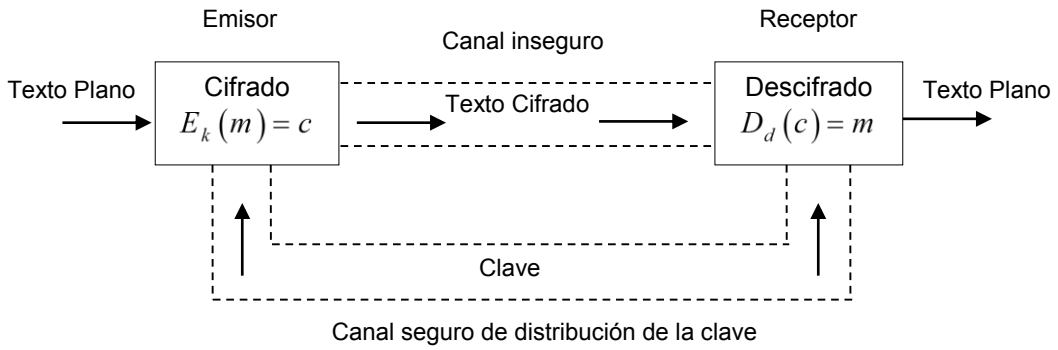


Figura 2. 4: Canal clásico de información.

El propósito de un esquema de cifrado es obtener confidencialidad. Las dos partes Alice y Bob de manera secreta escogen o intercambian las claves (k,d). Alice envía un mensaje m a Bob, Alice computa $c = E_k(m)$ y transmite el resultado a Bob, luego de recibir c Bob computa $D_d(c) = m$ y de esta manera recobra el mensaje original.

Una premisa fundamental en criptografía es que los conjuntos $\{E_k : k \in K\}$, $\{D_d : d \in K\}$ son de conocimiento publico. Cuando dos partes se comunican de manera segura usando un esquema de cifrado, el único detalle a mantener en secreto es el par de claves (k,d) que están usando y que deben seleccionar. Se puede agregar de manera adicional seguridad si se mantiene en secreto las transformaciones de cifrado y descifrado, pero no debe basarse en esto la seguridad del esquema completo.

2.14 Sistema Criptográfico

El sistema Criptográfico suele llamarse también criptosistema [15].

La definición de un sistema criptográfico está en función de sus componentes y es una quíntupla (M,C,K,E,D), donde:

1. Un espacio de mensajes de texto plano, M
2. Un espacio de mensajes de texto cifrado, C.
3. Un espacio de claves K.
4. Una familia de transformaciones de cifrado $E_k(m)=c$ o $E_k : M \to C$, donde $k \in K$.
5. Una familia de transformaciones de descifrado $D_k(c)=m$ o $D_k : C \to M$, donde $k \in K$.

La transformación E_k debe ser biyectiva $\forall k \in K$. Cada transformación de cifrado E_k está definida por un algoritmo de cifrado E y un clave k, que distingue de las demás transformaciones.

De manera similar cada transformación de descifrado D_k está definida por un algoritmo de descifrado D y una clave k.

Los sistemas criptográficos satisfacen los siguientes requerimientos [15].

1. Las transformaciones de cifrado y descifrado deben ser eficientes para todas las claves.
2. Los sistemas deben ser fáciles de usar.
3. La seguridad de un sistema depende solamente del ocultamiento o secreto de las claves y no del secreto de los algoritmos E y D.

Existen dos tipos fundamentales de sistemas criptográficos o criptosistemas:

- *Criptosistemas simétricos.* Son aquellos que emplean la misma clave k tanto para cifrar como para descifrar. Presentan el inconveniente de que para ser

empleados en comunicaciones la clave k debe estar tanto en el emisor como en el receptor, lo cual lleva a pensar como transmitir la clave de forma segura.

- *Critpsistemas asimétricos*. Estos emplean una doble clave (k,d). La clave k es privada y la clave d es pública. La clave k se utiliza en la transformación E de cifrado y la clave d en la transformación D de descifrado. Además estos criptosistemas deben cumplir que el conocimiento de la clave pública no permita calcular la clave privada. Para canales inseguros se pueden establecer comunicaciones seguras, puesto que por el canal viaja la clave pública que es para cifrar o para llevar a cabo autenticaciones.

En la práctica, se emplea la combinación de estos criptosistemas. El cifrado de mensajes largos se hace mediante algoritmos simétricos, que suelen ser muy eficientes y después se hace uso de la criptografía asimétrica para cifrar las claves simétricas (cortas).

La gran variedad de sistemas criptográficos produce necesariamente gran variedad de técnicas de criptoanálisis, cada una de ellas adaptada a un algoritmo o familia de ellos. Con toda seguridad, cuando en el futuro aparezcan nuevos mecanismos de protección de la información, surgirán con ellos nuevos métodos de criptoanálisis. De hecho, la investigación en este campo es tan importante como el desarrollo de algoritmos criptográficos. Esto es debido a que, mientras que la presencia de fallos en un sistema es posible demostrarla, su ausencia es por definición indemostrable.

La seguridad de los criptosistemas se suele medir en términos de número de computadoras y del tiempo necesarios para romperlos, y a veces simplemente en función del dinero necesario para llevar a cabo esta tarea con éxito.

Tampoco conviene depositar excesiva confianza en el algoritmo de cifrado, por ejemplo, no tiene sentido emplear algoritmos con niveles de seguridad extremadamente elevados si al escoger una clave esta sea fácil de adivinar.

2.15 Sistemas Criptográficos Clásicos

Ejemplos de sistemas criptográficos clásicos son el cifrado César.

Cifrado de César.

El sistema criptográfico es una quíntupla de (M,C,K,E,D), a continuación la quíntupla:

- $M = C = \Sigma^*$
- $K = \Sigma = \mathbb{Z}_{27} = \{0,1,...,26\}$

Se tiene el texto plano $m = \alpha_1, \alpha_2, ...\alpha_s$, entonces el texto cifrado es E_k $(\alpha_1, \alpha_2, ...\alpha_s) = c$ donde $c = (\beta_1, \beta_2, ...\beta_s)$ se tiene que $\beta_i = E_k(\alpha_i)$, $\forall i \in \{1,....,s\}$, i es la posición del caracter en el alfabeto (k).

- $E_k(m) = (m+k) \bmod 27$, donde para cada i se tiene $c_i = (m_i + k) \bmod 27$.
- $D_k(c) = (c-k) \bmod 27$, donde para cada i se tiene $m_i = (c_i - k) \bmod 27$.

Otra forma de ver E_k y D_k es:

$E_k(\alpha) = \alpha + k$

$D_k(\alpha) = \alpha - k$

El sistema criptográfico tiene 27 claves, se puede hacer un criptoanálisis por fuerza bruta, es decir, intentar cada clave k en un texto cifrado hasta obtener el texto plano.

Otro análisis puede ocurrir por análisis de frecuencia, dado que cifrar una letra siempre resultará en otra misma letra, y basta con encontrar aquella con más frecuencia de aparición.

Los sistemas criptográficos clásicos se clasifican de acuerdo a las técnicas que utilizaban, como la sustitución y la transposición sobre el mensaje original o texto plano.

2.16 Sistemas Criptográficos Modernos

Los sistemas criptográficos modernos, del cual el cifrado en *bits* está orientado a los caracteres ASCII o ANSI, usan por lo general la operación algebraica $\mathbb{Z}_n$, un conjunto finito. Este módulo no necesariamente debe corresponder al número de elementos del alfabeto o código utilizado.

La fortaleza de estos sistemas se debe basar en la imposibilidad de descubrir una clave secreta, pero si en la posibilidad de que el algoritmo criptográfico sea público.

La criptografía moderna utiliza las ideas básicas de la criptografía clásica, pero ahora se hace el uso de claves. El método de cifrado con claves permite cambiar el método con solo cambiar la clave. La Figura 2.5 muestra la clasificación de los sistemas criptográficos modernos.

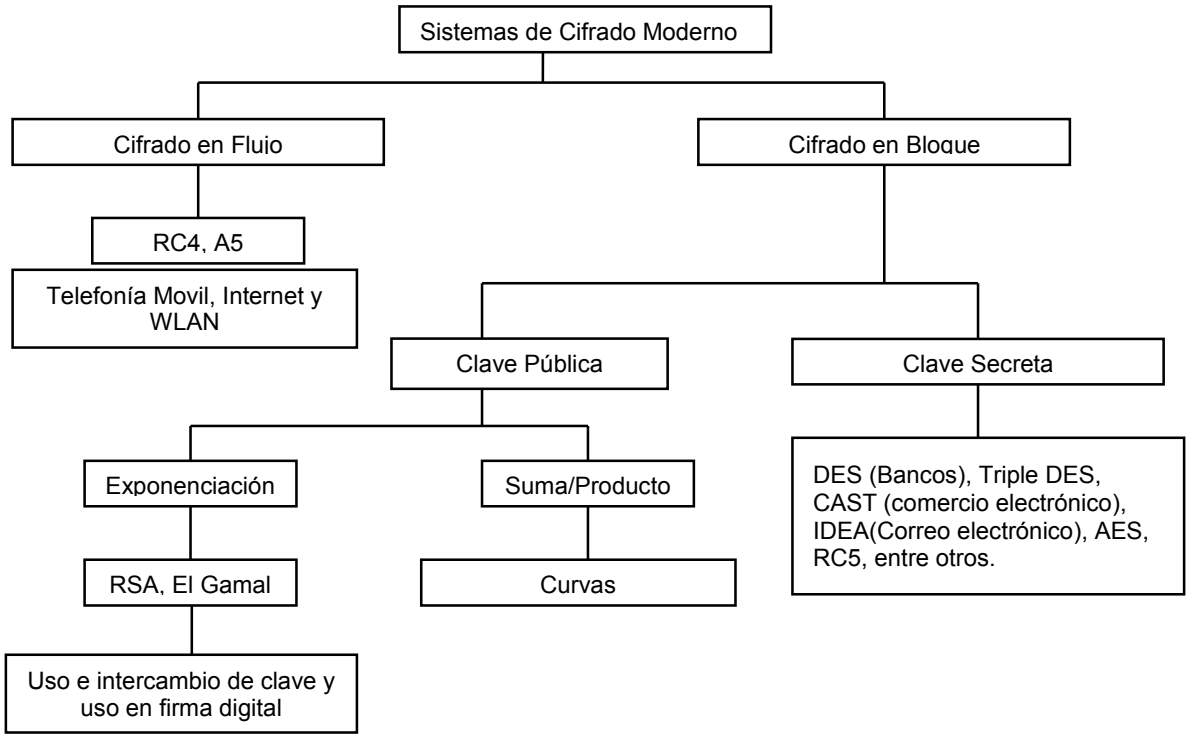


Figura 2. 5: Clasificación de los sistemas criptográficos moderno.

2.17 Protocolo

Un protocolo es una serie de pasos que involucran dos o más participantes, designados para cumplir una tarea. Un protocolo tiene una secuencia, es decir tiene un inicio y un fin. Cada paso del protocolo se ejecuta de manera ordenada y ningún paso se ejecuta hasta terminar los pasos previos. Algunas características de los protocolos son [10]:

- Los participantes en el protocolo deben saber con anticipación en que consiste el protocolo.
- Los participantes en el protocolo deben estar de acuerdo con seguir dicho protocolo.
- El protocolo no debe ser ambiguo; cada paso debe estar bien definido y no debe prestarse a malos entendidos.
- El protocolo debe ser completo; debe haber una acción específica para cada posible situación.

2.18 Protocolo Criptográfico

Un protocolo criptográfico es un protocolo que usa la criptografía. Un protocolo criptográfico involucra algunos algoritmos criptográficos, el objetivo generalmente va más allá que el ocultamiento, y es la seguridad.

De manera formal se tiene la siguiente definición de protocolo criptográfico:
Un protocolo criptográfico es un algoritmo definido por una secuencia de pasos precisos especificando las acciones requeridas de dos o más entidades para alcanzar un objetivo de seguridad específico [13].

El objetivo principal de la criptografía en un protocolo es prevenir o detectar espionaje o engaño. En general se puede establecer lo siguiente:
No debería ser posible hacer más o aprender más de lo que se especifica en el protocolo [10].

Algunos protocolos pueden fallar porque los diseñadores no estuvieron lo suficientemente involucrados en la definición de los requerimientos. Otros protocolos fallan porque los diseñadores no se involucran lo suficiente en el análisis del mismo. Hoy en día la interacción entre las personas toma lugar por medio de las redes computacionales. Las computadoras necesitan protocolos formales para interactuar de la misma forma que las personas lo hacen.

Mucha de la fortaleza de un protocolo para ofrecer seguridad está en base a la presencia de una persona.

Es de suponerse que una persona en una red de computadoras, como un administrador o diseñador, sea honesta. La mayoría lo son, pero aquellos pocos pueden causar mucho daño.

Al formalizar un protocolo se examinan las formas en que una parte de dicho protocolo es deshonesta y puede causar daño al mismo. Entonces se pueden desarrollar protocolos que son inmunes a tal corrupción.

Los participantes o partes en los protocolos son personas, como por ejemplo Alice y Bob. Como regla, Alice iniciará los protocolos y Bob los responderá. Si el protocolo requiere una tercera o cuarta persona, Carol y Dave pueden asumir los roles. La Tabla 2.3 muestra los personajes en los protocolos criptográficos.

Alice	Primer participante en todos los protocolos
Bob	Segundo participante en todos los protocolos
Carol	Participante en la tercera o cuarta parte de un protocolo
Dave	Participante en la cuarta parte de un protocolo
Eve	Espía
Mallory	Atacante activo malicioso
Trent	Arbitro confiable
Walter	Será el guardián de Alice y Bob en algunos protocolos
Peggy	Ejecutor de pruebas
Victor	Verificador

Tabla 2. 3: Personajes en los protocolos criptográficos.

Si se desea leer el protocolo para la comunicación usando criptografía simétrica y criptografía asimétrica consultar la referencia [10].

Capítulo 3

Criptografía Cuántica (Protocolos)

En este capítulo se definen algunos de los conceptos principales que conforman la criptografía cuántica así como de la computación cuántica que es una propuesta distinta a la computación clásica, considerando que con clásica nos referimos a las computadoras como las conocemos en la actualidad. La computación cuántica se basa en el uso de *qubits* en lugar de *bits*, dando lugar a nuevos elementos de diseño tales como compuertas lógicas que hacen posibles nuevos algoritmos, permitiendo que algunos problemas intratables pasen a ser tratables. Una computadora clásica equivale a una Maquina de Turing, una computadora cuántica equivale a una máquina de Turing cuántica [16].

3.1 Información Clásica

La computación cuántica permite una nueva forma de procesamiento de información, por lo que se considera que los métodos de computación y procesamiento de información tradicionales sean referidos como información clásica.

El *bit* (contracción de *binary digit*) es una unidad de información que básicamente representa un sí o un no como respuesta a una pregunta [1]. De manera matemática podemos usar este hecho como si se estuviera tratando de un sistema de dos estados para representar información seleccionando números binarios o usar la base 2.

El número binario puede ser 0 o 1 y el *bit* puede asumir estos valores (o uno o el otro). Un *bit* se puede implementar de manera eléctrica en un circuito para cero voltios o tierra (0 binario), o $+5$ voltios (1 binario). Técnicamente estos *bits* se relacionan con el funcionamiento de transistores que interrumpiendo o dejando

pasar una corriente eléctrica representan los estados 0 y 1. En teoría de la información (clásica) la unidad básica de la información es un *bit* binario.

Considerando lo anterior, es importante entender como conocer el número de *bits* requeridos para representar algo, esto puede ser determinado de la siguiente manera: suponer que alguna cantidad puede asumir uno de m diferentes estados. Entonces,

$$2^n \geq m$$

para alguna n. En donde la n nos dice el número de *bits* que se necesitan para representar o codificar una cantidad. Por ejemplo, si se desea representar los números $0,1,2,3$ en binario, entonces se tienen 4 unidades, y $2^2 = 4$. Por lo tanto, se necesitan al menos dos *bits* para representar estos números. Es decir, con dos *bits* codifico 4 mensajes o 4 números $(00,01,10,11)$ [2].

3.2 Información Contenida en una Señal

Ahora, dado un mensaje m, ¿Cuánta información está contenida en este mensaje? Si se aplica logaritmo base 2 en ambos lados, obtenemos que

$$\log_2 m = n$$

La ecuación anterior fue propuesta por Ralph Hartley [2]. Fue uno de los primeros intentos al cuantificar la cantidad de información en un mensaje. Lo que quiere decir es que, n *bits* pueden almacenar m diferentes mensajes.

Una de las unidades más básicas de información es el *byte* y se define para manipular la información con más facilidad, el *byte* está formado de ocho *bits*.

$$\log_2 256 = 8$$

Por lo tanto un *byte* puede almacenar 256 diferentes mensajes $(00000000, 0000001, 00000010, ..., 1111111)$ [2].

3.3 Computación Clásica

La encarnación moderna de las ciencias de la computación fue anunciada por el gran matemático Alan Turing en un notable artículo publicado 1936 "*On computable*

numbers, with and application to the Entscheidungsproblem" [18]. Turing desarrolla en detalle una noción abstracta de lo que ahora llamaríamos computadora programable, un modelo para el computo conocido como Maquina de Turing. Turing muestra que existe una Máquina de Turing Universal que puede ser usada para simular otra Máquina de Turing [17]. Por otra parte afirma que esta Máquina de Turing Universal captura completamente lo que significa desempeñar una tarea por medios algorítmicos. Esto es, si existe un algoritmo que puede desempeñarse sobre cualquier pieza de hardware (digamos una computadora), entonces hay un algoritmo equivalente para una Máquina de Turing Universal la cual desempeña exactamente la misma tarea ejecutándose en una computadora. Esta afirmación es lo que se conoce como la Tesis de Church-Turing [17], en honor a Turing y a Alonzo Church otro pionero de la computación. La tesis afirma la equivalencia entre el concepto de qué clase de algoritmos pueden ser desempeñados sobre algún dispositivo físico con el riguroso concepto de una Máquina de Turing Universal.

No tomó mucho tiempo después de las publicaciones de Turing donde las primeras computadoras con componentes electrónicos fueron desarrolladas. Jhon von Neumman aporta las primeras ideas para construir una computadora, desarrolló un modelo teórico, del cual describía como poner de una manera práctica todos los componentes necesarios para que una computadora sea totalmente tan capaz como una Máquina de Turing Universal.

Los requerimientos de hardware de las computadoras han crecido en desempeño a pasos agigantados, tanto es así que el crecimiento fue codificado por Gordon Moore en 1965 [17] conocida como la Ley de Moore, que establece que el desempeño de una computadora se duplicará para un costo constante cada dos años, o en otras palabras el número de transistores se duplicara cada dos años.

Ahora bien regresando a la Tesis de Turing-Church. Un algoritmo es eficiente cuando su velocidad de cálculo crece de manera polinomial con respecto a la información, es decir, en general, se considera que la solución de un problema es eficiente si el tiempo de ejecución está acotado por una función polinomial. La idea de que un algoritmo es eficiente o ineficiente fue precisada matemáticamente en el área de la complejidad computacional.

La tesis fuerte de Turing-Church dice:
Cualquier proceso algorítmico puede ser simulado eficientemente usando una Máquina de Turing. Esto implica que no importa el tipo de máquina que se esté empleando para desempeñar un algoritmo, esta máquina puede ser simulada eficientemente usando una Máquina de Turing [17].

Un problema fuerte que enfrentó la tesis de Church-Turing surgió a mediados de la década del año 1970, cuando Robert Solovay y Volker Strassen demostraron que es posible comprobar si un número entero es primo o compuesto utilizando un algoritmo aleatorio. Es decir, la prueba Slovay-Strasen prueba que la primalidad utiliza aleatoriedad como parte esencial del algoritmo. El algoritmo no determina con certeza si dado un entero este fue primo o compuesto. En vez de eso el algoritmo puede determinar que el número fue probablemente primo o compuesto con certeza. La tesis se viola porque al parecer las computadoras con acceso a generadores de números aleatorios podrían ser capaces de desempeñar de manera eficiente tareas computacionales con soluciones no eficientes sobre una Máquina de Turing determinística. Los algoritmos aleatorios plantean un desafío a la tesis fuerte de Turing-Church, lo que sugiere que hay problemas eficientemente solubles que, sin embargo, no pueden ser eficientemente resueltos sobre una Máquina de Turing determinística. Este desafío es resuelto simplemente modificando la tesis fuerte de Turing-Church:
Cualquier proceso algorítmico puede ser simulado eficientemente usando una Máquina de Turing Probabilística.

En 1985 David Deutsch preguntó si las leyes de la física se podrían utilizar para obtener una versión aún más fuerte de la tesis de Church-Turing [17], [19]. Debido a que la mecánica cuántica se encuentra en la raíz de todas las demás teorías físicas, Deutsch la elige para considerar nuevos dispositivos basados en la mecánica cuántica. La primer pregunta a resolver por Deutsch fue que si la

computadora cuántica postulada sería capaz de resolver eficientemente problemas que se resistieran a serlo en una Maquina de Turing Probabilística. Construyó un simple ejemplo que sugería el potencial de una computadora cuántica en comparación de una computadora clásica.

Varios investigadores dedicaron tiempo a mejorar los primeros pasos de Deutsch, hasta que en 1994 Peter Shor sorprendió con la demostración de dos problemas importantes [17]: el problema de encontrar los factores primos de un número y el llamado problema de logaritmo discreto, estos podrían ser resueltos eficientemente sobre una computadora cuántica. Los resultados de Shor son una fuerte indicación de que las computadoras cuánticas son más poderosas qua las Máquinas de Turing incluso las probabilísticas. Otra prueba del poder de las computadoras cuánticas llega en 1995 cuando Lov Grover mostró que otro problema importante el problema de la realización de una búsqueda a través de algún espacio de búsqueda no estructurada (una base de datos), el tiempo empleado en el mejor algoritmo clásico conocido escala con el tamaño de la base de datos, mientras que en el caso del algoritmo de Grover este escalado es con la raíz cuadrada de ese tamaño.

Más o menos al mismo tiempo que los algoritmos de Shor y de Grover fueron descubiertos, varias personas estaban desarrollando una idea que Richard Feynman había sugerido en 1982 [17], [19]. Feynman reflexionó acerca de las dificultades de las computadoras clásicas al simular sistemas mecánicos cuánticos y sugirió construir computadoras basadas en principios de la mecánica cuántica que permitieran evitar dichas dificultades. En particular, le interesaba el problema de hacer simulaciones del mundo real fin de cuentas es cuántico, por medio de computadoras cuánticas.

La descripción matemática de los fenómenos cuánticos es la ecuación de Schrödinger. Se trata de una ecuación diferencial que una computadora clásica puede manipular. En otras palabras, se pueden resolver ecuaciones de movimiento cuántico con una computadora clásica.

Pero la dificultad, como indicó Feynman, es que sólo se pueden resolver problemas muy simples y de poco interés, en los que intervienen sólo unas cuantas partículas (en un sistema cuántico de interés hay miles de millones de partículas).
Para simular procesos cuánticos no triviales la computadora clásica tendría que ser gigantesca, porque su capacidad aumenta en forma lineal. Richard Feynman propuso la utilización de sistemas cuánticos sencillos, llamados *qubits* (de *quantum bits*), como elementos estructurales básicos de una nueva computadora. El subcapítulo 3.8 aborda el tema del *qubit*.

La idea de la computación cuántica surge en 1981, cuando Paul Benioff expone una teoría para aprovechar las leyes cuánticas en el entorno de la computación [17].
Benioff teorizó una computadora tradicional (máquina de Turing) operando con algunos principios de la mecánica cuántica introduciendo la Máquina de Turing cuántica. En vez de trabajar con voltajes eléctricos, se trabaja a nivel de cuánto.
En la computación digital un *bit* solo puede tomar dos valores: cero o uno. En la computación cuántica, intervienen las leyes de la mecánica cuántica, y la partícula puede estar en superposición coherente o cuántica: por lo cual puede ser cero o puede ser uno y puede ser cero y uno a la vez.
El número de *qubits* indica la cantidad de *bits* que pueden estar en superposición. Con los *bits*, si se tiene un registro de tres *bits*, se tienen ocho valores posibles $(000,001,010,011,100,101,110,111)$ y el registro solo puede tomar uno de esos valores. Pero en el caso de los *qubits* la partícula puede tomar esos mismos ocho valores distintos al mismo tiempo. Es decir, mientras el *bit* crece con la proporción $1=1,\ 2=2$ o $n=n$, el *qubit* crece con la proporción 2^n.

Tal vez y de forma segura una de las dificultades por las que pasará la programación cuántica, será la propia adaptación a la misma. Además, la exigencia sobre un algoritmo de una computadora cuántica es mayor; no solo debe funcionar bien, debe funcionar mejor que cualquier algoritmo clásico existente para resolver el mismo problema.

A medida que evoluciona la tecnología se fabrican chips cada vez más pequeños, los chips o circuitos integrados o de alta integración se afirma, no pueden ser infinitamente pequeños, hay un límite en el cual dejan de funcionar debido al efecto túnel. Es donde comienzan a surgir problemas relacionados con la transición de la física clásica a la física atómica.

3.4 Algoritmos Cuánticos

Algunos algoritmos cuánticos, de los cuales trabajan reduciendo el margen de error a niveles comparables al nivel de error de las computadoras actuales son:

- Algoritmo de Shor [2], [24]
- Algoritmo de Grover [2], [24]
- Algoritmo de Deutsh-Jozsa [17], [24]

La complejidad computacional llamada BQP (*Bounded error quantum polynomial time*) [17], [20], representa la clase de algoritmos que pueden ser resueltos en una computadora cuántica en tiempo polinómico con un margen de error promedio inferior a $1/4$. Dicho de otra forma, existe un algoritmo de computación cuántica cuya cota superior en tiempo es polinómica para resolver ese problema, tal que la probabilidad de obtener una respuesta equivocada es menor de 25%.

La computación cuántica ha sugerido una alternativa superior a la computación clásica en la promesa del desarrollo de un conjunto de nuevos algoritmos para la resolución de varios problemas, dos clases de algoritmos que cumplen esta promesa se mencionan a continuación [20]:

La transformada de Fourier cuántica de Shor abarca notables algoritmos para factorizar números enteros y para hallar logaritmos discretos. La complejidad del logaritmo discreto es empleada en criptografía, siendo la base de los algoritmos como El Gamal y DSA (*Digital Signature Algorithm*) [8], [10].

El siguiente algoritmo está basado en la búsqueda cuántica de Grover, como un método de búsqueda.

Sin embargo otro problema que resuelve la computación cuántica es la simulación de sistemas cuánticos: donde Richard Feyman en 1982 conjetura que las

computadoras cuánticas como simuladores universales de sistemas cuánticos, y en 1996 se demostró que la conjetura era correcta [17].

3.5 Mecánica Cuántica

La mecánica cuántica es una nueva teoría física para describir el comportamiento de los microsistemas (átomos, moléculas, cristales, etc.) [31].

La mecánica cuántica como rama de la física, es la ciencia que tiene por objeto el estudio y comportamiento de la materia y la energía a escala reducida, es decir, en el entorno de átomos o partículas.

El concepto reducido se refiere a tamaños a partir de los cuales empiezan a notarse efectos como el principio de incertidumbre de Heisenberg [3], [17], que establece la imposibilidad de conocer con exactitud la posición de una partícula, o su energía, o conocer simultáneamente su posición y velocidad, sin afectar a la propia partícula. Entonces en cualquier sistema físico existe una diversa multiplicidad de estados (energía, posición, velocidad), los cuales habiendo sido descritos mediante ecuaciones matemáticas por los físicos, son denominados estados cuánticos. Una propiedad de los sistemas cuánticos consiste en que, como resultado de una medición, el estado inicial del sistema cuántico se destruye y el sistema se colapsa en un nuevo estado, que se determina según el resultado de tal medición [23].

La física clásica es una ciencia determinista, es decir, se puede calcular con exactitud valores como velocidad o posición de un objeto y su trayectoria. Por otro lado, la mecánica cuántica estudia el comportamiento de objetos microscópicos (átomos, electrones, quarks o fotones) a este nivel microscópico las leyes de la física clásica no son capaces de predecir el comportamiento de estas partículas, siendo su principal característica la indeterminación de valores como la posición o velocidad de una partícula. Es entonces que la principal diferencia de la cuántica respecto a la clásica es que esta última es una ciencia determinista y la primera es probabilista.

A nivel microscópico las partículas fundamentales como los electrones o los fotones se comportan de una manera diferente, algunos ejemplos de estos comportamientos son:

- El fenómeno de superposición indica una electrón está presente en muchos lugares a la vez con diferentes probabilidades siempre y cuando no sea observada [2], [17], [21].
- El fenómeno de entrelazamiento indica que un electrón puede estar entrelazado con otro que se ubica una gran distancia [2], [17].
- El efecto túnel nos afirma que un electrón puede atravesar en ocasiones muros o paredes delgadas de energía como si fuera un fantasma, esto se debe a la dualidad onda-partícula que confirma la mecánica cuántica [22].

La computación cuántica y la información cuántica es el estudio de las tareas de procesamiento de información que se pueden realizar utilizando sistemas mecánicos cuánticos [17].

3.6 La Computadora Cuántica

David Deutsch dio una base matemática sólida a la propuesta de Feynman. Deutsch explicó cómo podría funcionar una computadora cuántica universal y describió su funcionamiento como secuencias de operaciones elementales sobre *qubits* [16]. La operación de una computadora cuántica es muy distinta de la operación de la máquina de Turing. Había que formular algoritmos computacionales cuánticos.
Los algoritmos cuánticos hacen uso de las características de los *qubits*, por ejemplo, al iniciar se pueden poner los *qubits* en una superposición de cero y uno, estos *qubits* representan la información. Ahora solo hay que hacer un cálculo, es decir, aplicar el algoritmo. Operando sobre los *qubits* en superposiciones de 1 y 0, el algoritmo cuántico resuelve, en cierta forma, todos los cálculos posibles al mismo tiempo. Se puede imaginar una computadora cuántica como varias computadoras clásicas que funcionan en paralelo. A esto es lo que se le conoce como paralelismo cuántico, y disminuye representativamente el número de pasos para resolver un problema en una computadora cuántica.

Para crear una computadora cuántica se necesita un sistema físico que se comporte cuánticamente durante un periodo de tiempo suficientemente largo y que pueda ser

manipulado fácilmente. Se está experimentando en *qubits* con partículas cuánticas como átomos o fotones, entre otros. Aunque se han ideado algoritmos cuánticos para efectuar tareas como factorizar enteros, se afirma que no existe propiamente ningún algoritmo cuántico de criptografía. Hoy en día, lo que se entiende cuando se menciona criptografía cuántica es en realidad la distribución cuántica de claves, es decir, aún no está claro en qué situaciones una computadora cuántica proporciona una ventaja significativa sobre el equivalente clásico. Entender qué problemas son sencillos y cuáles complejos para una computadora cuántica, y cómo estos resultados se comparan con la computadora clásica, son uno de los objetivos primordiales de la computación cuántica, subdisciplina de la Teoría de la Información Cuántica.

La compañía canadiense D-Wave Systems presenta en el 2007 en Silicon Valley, una primera computadora cuántica comercial de propósito general de 16 *qubits*, la compañía admite más tarde que la maquina llamada Orion, no es realmente una computadora cuántica sino que esta máquina usa algo de mecánica cuántica. Los científicos de D-Wave admiten que Orion no resuelve tareas cuánticas como la factorización de enteros para explotar los algoritmos de cifrado. A pesar de la controversia la maquina resuelve una serie de tareas de optimización. Algunas de las inquietudes de la empresa Google ha sido el experimentar con esta máquina y colaborar con expertos en la cuántica con la intención de construir *hardware* cuántico[1]. Actualmente Google también cuenta con un equipo de inteligencia artificial cuántica[2].

3.7 Información Cuántica

En 1948 Claude Shannon publica un par de documentos que sientan las bases de la moderna teoría de la información y la comunicación. Uno de los primeros inicios

[1] http://spectrum.ieee.org/tech-talk/computing/hardware/google-hires-quantum-computing-expert-john-martinis-to-build-new-hardware

http://www.extremetech.com/computing/184242-d-wave-confirmed-as-the-first-real-quantum-computer-by-new-research

[2] https://plus.google.com/+QuantumAILab/about
http://www.extremetech.com/extreme/189155-google-begins-developing-its-own-quantum-computer-chips-to-prepare-for-the-future

de Shannon fue definir matemáticamente el concepto de información. Shannon estuvo interesado en dos cuestionamientos relacionados con la comunicación de la información sobre un canal de comunicación. El primero, ¿Qué recursos son requeridos para enviar información sobre un canal de comunicación?, el segundo, ¿Puede la información ser transmitida de tal forma que es protegida contra ruido en el canal de comunicación? Shannon responde a estos cuestionamientos con dos teoremas fundamentales de la teoría de la información. El primero, el teorema de codificación de canal silencioso de Shannon, el segundo, el teorema de codificación de canal ruidoso [17]. Shannon mostró que los códigos de corrección de errores se podrían utilizar para proteger la información que ha sido enviada. El segundo teorema de Shannon va más allá dando un límite superior, otorgando protección por medio de códigos de corrección de errores, pero tristemente el teorema no ofrece explícitamente una práctica útil de estos códigos para alcanzar ese límite. Desde la publicación de los documentos de Shannon a la fecha, investigadores se han dado a la tarea de construir más y mejores clases de códigos correctores de errores en sus intentos de acercarse cada vez más a los límites establecidos por el teorema de Shannon. La Teoría de la información Cuántica ha seguido similares desarrollos.

En 1995 Ben Schumacher provee un análogo al teorema de codificación sin ruido, y en el proceso define el *bit* quántico o *qubit* (*quantum bit*) como un recurso físico tangible [17]. Sin embargo, aún no se ha encontrado el análogo completo al segundo teorema sobre canales ruidosos en información cuántica, aunque la teoría de la corrección de errores cuántica ha encontrado códigos correctores de errores que permiten a una computadora cuántica transmitir información a través de canales ruidosos.

La teoría de la información cuántica está basada en *qubits*, actualmente, para guardar un *bit* de información se utilizan objetos macroscópicos, algunos son las nano partículas de los discos duros. El límite físico que impone la reducción de escala está condicionado por el llamado límite súper paramagnético que marca una barrera por debajo de la cual no es posible almacenar información sin que se borre de manera espontánea, una propuesta o reto es utilizar cada vez menos número de

partículas o átomos para estos propósitos, tal límite es llegar a usar objetos microscópicos, al llegar a dichos límites se están utilizando los estados cuánticos de algunos sistemas físicos para codificar ceros y unos [23].

3.8 El Qubit

La unidad básica de la computación cuántica es el *qubit* (de quantum *bit*), al igual que el *bit*, el *qubit* puede estar en uno de dos estados. Estos estados pueden etiquetarse por $|0\rangle$ y $|1\rangle$.

A su vez la unidad de información cuántica, es decir, la cantidad de información cuántica que se puede resguardar en el estado del sistema cuántico más pequeño, por ejemplo, una partícula espín ½, o los estados de polarización de un fotón, se llama *qubit*.

En teoría cuántica un objeto encerrado usando la notación $|\ \rangle$ puede ser llamado un estado, o un vector, o un *ket*. Esta notación es la notación de Dirac [2], [17], [23].

Un *qubit* puede existir en el estado $|0\rangle$ o en el estado $|1\rangle$, pero también puede existir en lo que se llama estado de *superposición*. Esto es, un estado que es una combinación lineal de $|0\rangle$ y $|1\rangle$. El *qubit* es de la forma $\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$, una base del espacio de *qubits* se considera por ejemplo $\begin{pmatrix}1\\ 0\end{pmatrix},\begin{pmatrix}0\\ 1\end{pmatrix}$ que son llamados autovectores.

En la mecánica cuántica cualquier propiedad observable corresponde a un autovector, recuérdese el subcapítulo 3.5 que habla de los estados cuánticos.

Entonces el *qubit* tendría la forma $\alpha\begin{pmatrix}1\\ 0\end{pmatrix}+\beta\begin{pmatrix}0\\ 1\end{pmatrix}$. Este estado lo etiquetamos como $|\psi\rangle$, entonces la superposición de un estado está escrita como combinaciones lineales [24]:

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

Donde α y β son números complejos, en la mayoría de los casos pueden suponerse como reales. Esto es, los números son de la forma $z = x + iy$, donde $i = \sqrt{-1}$ [2], [17].

Un *qubit* puede existir en una superposición de los estados $|0\rangle$ y $|1\rangle$ mientras no se haga una medida para conocer en qué estado se encuentra.

Cuando el *qubit* es medido, solo puede ser encontrado en el estado $|0\rangle$ o en el estado $|1\rangle$. Las leyes de la mecánica cuántica establecen que el modulo al cuadrado de α, β en $|\psi\rangle$, proporcionan la probabilidad de encontrar el *qubit* en el estado $|0\rangle$ o $|1\rangle$. En otras palabras:

$|\alpha|^2$: Dice la probabilidad de encontrar $|\psi\rangle$ in el estado $|0\rangle$

$|\beta|^2$: Dice la probabilidad de encontrar $|\psi\rangle$ in el estado $|1\rangle$

Los cuadrados de estos coeficientes están relacionados con la probabilidad de obtener las medidas resultantes y α, β tienen que cumplir con que

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

Se puede examinar los *bits* para saber si valen 0 o 1, pero con el *qubit* no ocurre esto, antes de medirlo se encuentra en un estado de superposición, es como El gato de Schrödinger [25], [26], es un célebre experimento del físico austriaco Edwin Schrödinger, según el cual un gato se encuentra encerrado en una caja con un átomo radioactivo, un contador Geiger y una ampolla de cianuro cuyo contenido cae en un cubo con ácido cada vez que el contador Geiger detecta la emisión de partículas radiactivas. Si se emite la partícula, el detector romperá la ampolla y el gato morirá. Si no se emite, la ampolla seguirá intacta y el gato vivirá. La probabilidad de que el gato permanezca con vida es del 50%. Solo abriendo la caja se averigua qué le ha ocurrido al gato, pero mientras tanto este estaría vivo y muerto a la vez. Es una forma de expresar un concepto fundamental de la física cuántica: la dualidad onda-partícula, que logra explicar por observación. Entonces el *qubit* está en 0 y en 1 a la vez, cuando se mide el *qubit* se obtiene o 0 con la probabilidad

$|\alpha|^2$ o bien 1 con la probabilidad de $|\beta|^2$. Por ejemplo, un *qubit* puede estar en el estado

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

el cual cuando es medido, el resultado de obtener 0 es el 50% $\left(|1/\sqrt{2}|^2\right)$ y el de obtener 1 el 50% del tiempo también, el *qubit* está en un estado puro si está como $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

El *qubit* es un sistema cuántico y también es un sistema microscópico, como el átomo, el spin nuclear, o un fotón polarizado. Una colección de n *qubits*, es llamado un registro cuántico de tamaño n.

3.9 Múltiples Qubits

Cuando se tiene dos *bits* se tienen cuatro posibles estados, es decir, 00, 01, 10 y 11. Cada estado en la notación de Dirac, se correspondería con $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$, que es lo que representa un vector de dos *qubits*. La superposición de estos estados queda de nuevo con la combinación lineal de los mismos:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \text{ (Sistema de dos } \textit{qubits}\text{)}$$

Un sistema de n-*qubit* es frecuentemente llamado un registro (cuántico) en el contexto de la computación cuántica.

Un registro de tamaño tres puede almacenar de manera individual el 3 o el 7,

$|0\rangle \otimes |1\rangle \otimes |1\rangle = |011\rangle = |3\rangle$

$|1\rangle \otimes |1\rangle \otimes |1\rangle = |111\rangle = |7\rangle$,

pero también puede almacenar 3 y 7 simultáneamente [25].

3.10 ¿Cómo utilizar el Qubit?

Se sabe que los objetos que siguen las leyes de la mecánica cuántica, como los átomos (partículas indivisibles), los electrones o los fotones que hacen la luz, están llenos de sorpresas. Pueden existir en más de un lugar al mismo tiempo, por

ejemplo, las partículas no tienen una posición y trayectoria definida, se hallan en muchos lugares a la vez con distintas probabilidades. La física cuántica acepta que ganar información de los sistemas cuánticos causa perturbación, esto es una característica intrínseca de la naturaleza cuántica. Este comportamiento abre la posibilidad de fenómenos sorprendentes, por ejemplo, una pareja de partículas puede desarrollar un tipo especial de relación, llamado *entrelazamiento*. Cuando algo modifica el estado de alguna de las dos partículas en el mismo momento se modifica el estado de la otra, esta especial conexión se mantiene incluso si las dos partículas están a los dos extremos de una galaxia. Aun a Einstein le costaba creerse los efectos de la física cuántica, definía el entrelazamiento como acción fantasmal a distancia.

En la actualidad se está experimentando en *qubits* con partículas cuánticas como átomos o fotones. Un medio que es capaz de transferir uno o más *qubits* se llama canal cuántico. Trabajar con fotones posibilita cubrir recorridos de grandes distancias, además de que pueden viajar por fibra óptica o por el espacio libre. Las fibras ópticas han sido el canal cuántico más popular hasta la fecha [3], [4]. En la actualidad se está experimentando en *qubits* con partículas cuánticas como átomos o fotones.

El fotón tiene masa cero y viaja en el vacío con una velocidad constante c. Como todos los cuantos de energía, el fotón presenta tanto propiedades corpusculares como ondulatorias. Se comporta como una onda en algunos fenómenos como la refracción que tiene lugar en una lente; o como una partícula cuando interacciona con la materia para transferir una cantidad fija de energía. Además de energía, los fotones llevan asociada una cantidad de movimiento o momento lineal, y tienen una polarización particular. Siguen las leyes de la mecánica cuántica, lo que significa que a menudo estas propiedades no tienen un valor bien definido. En su lugar se habla de las probabilidades de que un fotón tenga una cierta polarización, posición, o cantidad de movimiento.

Un fotón tratado como un *qubit* se puede polarizar. La representación física del *qubit* es el fotón. En algunos casos, $|0\rangle$ es sinónimo de un fotón polarizado horizontalmente $(0°)$ $|\leftrightarrow\rangle$, mientras que $|1\rangle$ representa un fotón polarizado

verticalmente $(90°)$ $|\updownarrow\rangle$, esta es llamada polarización lineal o rectilínea. Alternativamente se puede corresponder a fotones polarizados en diferentes direcciones, la polarización diagonal puede tomar $|0\rangle$ como un fotón polarizado a $45°$ $|\nearrow\rangle$ y $|1\rangle$ como un fotón a $135°$ $|\nwarrow\rangle$ [24].

Cada polarización de un fotón lleva un valor de cero o uno, a su vez también se toma en cuenta sus diferentes direcciones (lineal o diagonal), que será representado como un *qubit*.

Entonces representamos los vectores de polarización de la siguiente forma:

$$|\psi\rangle = \alpha|\leftrightarrow\rangle + \beta|\updownarrow\rangle \text{ base lineal o rectilínea}$$

$$|\psi\rangle = \alpha|\nearrow\rangle + \beta|\nwarrow\rangle \text{ base diagonal}$$

¿Hay un uso práctico para el uso de *qubits* individuales?, de manera sorprendente existe una forma de distribuir una clave criptográfica usando una secuencia de *qubits* de manera individual. La distribución cuántica de claves o QKD (*Quantum Key Distribution*) es una forma de distribuir una clave de cifrado y descifrado por medio del uso de *qubits*.

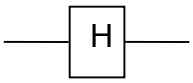
La mecánica cuántica señala que, la única medición posible es entre dos estados ortogonales. Por ejemplo, si se mide en una base lineal, los únicos resultados posibles son horizontal o vertical. Si el fotón fue creado con una polarización horizontal o vertical (con un generador lineal), entonces la medición será correcta. Pero si el fotón fue creado con una polarización diagonal $45°$ o $135°$ (generador diagonal), entonces se medirá al azar un resultado horizontal o vertical, es decir, el fotón quedará polarizado aleatoriamente en un estado horizontal o en un estado vertical. De esta manera se pierde toda la información inicial de la polarización, por lo tanto, si se envía un fotón que pasa a través de un filtro, no puede conocerse si poseía polarización horizontal o vertical, o bien si estaba en la base diagonal [3], [4].

3.11 Puerta Cuántica Hadamard H

Una puerta cuántica o puerta lógica cuántica es un circuito cuántico básico que opera sobre un pequeño número de *qubits*. Lleva a cabo una operación unitaria fija sobre una selección de *qubits* en un periodo fijo.

La compuerta cuántica Hadarmard H es usada por los algoritmos para crear estados de superposición, opera sobre un único *qubit* de entrada. H es una transformación unitaria definida por U_H:

$$H|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$
$$H|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right),$$

La representación gráfica del circuito es 

.

Ahora bien la compuerta Hadarmard se puede utilizar en paralelo, es decir, cuando n compuertas Hadamard actúan en paralelo sobre n *qubits*, se tiene la transformación Hadamard. Es la acción de Hadamard sobre un sistema de n-*qubits*. Una notación abreviada es $H^{\otimes n}$. Entonces 2^n estados son generados [2], [24]. La Figura 3.1 muestra dos compuertas Hadamard en paralelo.

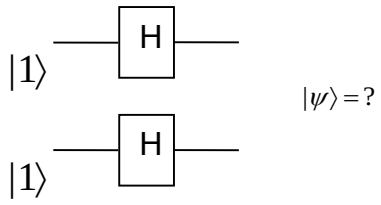


Figura 3. 1: Transformada de Hadamard $H^{\otimes 2}$ operando con dos qubits en paralelo.

Si hacemos $H^{\otimes 3}|0\rangle\,|0\rangle\,|0\rangle$ se obtiene:

$$\left(H \otimes H \otimes H\right)|0\rangle = \left(H|0\rangle\right)\left(H|0\rangle\right)\left(H|0\rangle\right)$$

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

En binario

$$= \frac{1}{2^{3/2}} \left\{ |000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle \right\}$$

Entonces se tiene la salida de la superposición de ocho números, del 0 al 7.

3.12 Criptografía Cuántica

En la comunidad criptográfica es común que al transmisor se le denomine Alice y al receptor o destinatario se le llame Bob y al espía o intruso Eve.

La criptografía cuántica es un conjunto de protocolos de comunicación cuántica, un protocolo importante es la distribución cuántica de claves o *Quantum Key Distribution* (QKD).

La distribución cuántica de claves o QKD se propuso como una solución a los problemas de distribución de claves en los sistemas criptográficos modernos (clásica).

La QKD permite que las partes involucradas en una comunicación puedan intercambiar las claves de forma segura, además las claves se pueden utilizar en la criptografía moderna.

La criptografía cuántica está fundamentada en la mecánica cuántica. Tres propiedades principales son usadas en la QKD [2]:

- El principio de incertidumbre de Heisenberg que establece que la medida de un estado cuántico cambia el mismo estado o lo colapsa [3], [17]. Un ejemplo de este efecto puede verse en la prueba de doble rendija.
- El teorema de No Clonación que establece que un estado cuántico no puede ser copiado, su prueba puede verse en las referencias [2], [17], [21].
- Las medidas son irreversibles.

Cuando Alice y Bob se están comunicando o transmitiendo *qubits*, Eve puede intentar espiar dicha comunicación con el objetivo de conocer lo que se está transmitiendo. El solo intento de espiar un *qubit* causa que su estado cambie y el

envío de información se interrumpe. Ahora por el teorema de no clonación Eve no puede medir el *qubit* y hacer una copia para enviarlo a Bob, es entonces que el acto del espía es detectado.

El establecer un canal de comunicación seguro no evita que un espía pueda extraer información del canal, pero si puede detectarse su presencia. Un canal de comunicaciones puede establecerse seguro si la información clásica se transmite codificada en estados cuánticos únicos. En el subcapítulo 3.14 se describe esta codificación con el uso de fotones.

La idea principal de la distribución cuántica de claves consiste en aprovechar el hecho de que un estado cuántico no se puede copiar y que no es posible realizar una medición sin tener que alterar dicho estado cuántico.

3.13 Detalles sobre la QKD

El propósito de la QKD es establecer una secuencia aleatoria de *qubits* compartida por Alice y Bob, cada uno de ellos puede estar seguro de que Eve el espía no sabe casi nada de la clave. En casos ideales con componentes tecnológicos perfectos, Eve no obtendría información y de hacerlo sería detectado escuchando la conversación y la transmisión sería cancelada.

Alice y Bob están interconectados a través de un canal clásico y un canal cuántico. Alice es el transmisor y Bob es el receptor [2], [21].

El canal clásico (público o convencional), es el canal por el cual se comunican Alice y Bob, considerando que previamente Alice y Bob se han autenticado. El canal clásico es un canal bidireccional que puede ser inclusive una línea telefónica, la red de Internet, un chat, etc.

La autenticación de Alice y Bob es una característica importante de los protocolos QKD, sin la autenticación estos protocolos pierden la seguridad.

El canal cuántico (privado), es el canal por el cual se distribuye la clave. Este canal solo tiene un sentido y generalmente es de fibra óptica. Las fibras ópticas han sido el canal cuántico más popular hasta la fecha.

La clave puede ser utilizada por Alice y Bob para el cifrado de un mensaje.

EL espía Eve, tiene como objetivos el obtener la máxima información posible acerca de la clave intercambiada entre Alice y Bob, para lo cual, debe disponer de acceso a los canales cuántico y clásico.

El acceso al canal cuántico es total, con la capacidad de hacer cualquier cosa que no esté prohibida por las leyes de la mecánica cuántica, que propician en resumen lo siguiente:

- No se puede copiar o duplicar la información transmitida a través del canal.
- La lectura de cualquier estado cuántico provocará la modificación del mismo.
- Eve podrá leer la información transmitida a través del canal clásico, pero no puede modificarla, ya que el canal se encuentra autenticado.

La Figura 3.2 muestra el esquema del canal cuántico y del canal clásico en un sistema de distribución cuántica de claves o QKD.

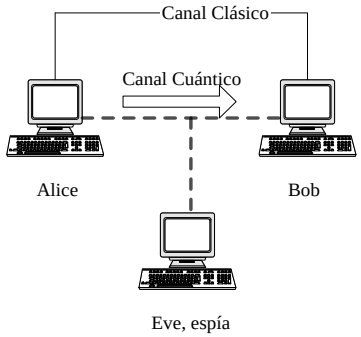


Figura 3. 2: Representación del sistema QKD.

3.14 Codificación de la Polarización

En la criptografía cuántica, los codificadores son los que definen como se representa la información clásica mediantes estados cuánticos.

Una forma de codificar información clásica es por medio de fotones polarizados. Una clave secreta podrá ser transmitida como fotones [21].

La luz es una onda electromagnética, que oscila y se propaga en diferentes direcciones, la polarización es la propiedad que describe en que plano vibra el campo electromagnético. La luz está formada por fotones, el fotón es la partícula de la luz más pequeña. Cuando un rayo de luz se absorbe por la materia, la energía que ésta retiene son cantidades finitas, o quantos. Un quanto de luz es llamado fotón, del cual también se demostró que tiene momento angular (estado de polarización) definido [31].

Los valores lógicos de los *qubits* están representados por los grados de libertad de la polarización de los fotones. La base ya sea lineal (o rectilínea) o diagonal es escogida aleatoriamente, esto hace la elección aleatoria de las direcciones de la polarización que representan los valores lógicos 0 y 1.
Explicando con un poco de detalle la polarización: se cuenta con una fuente de luz (un LED o un láser), esta luz se filtra para producir un rayo polarizado en ráfagas cortas y con muy baja intensidad. La polarización de cada ráfaga se modula por Alice de forma aleatoria en uno de los cuatro estados $(\leftrightarrow\updownarrow\swarrow\nwarrow\searrow)$. La Figura 3.3 muestra la representación de polarizar fotones en uno de los cuatro estados.

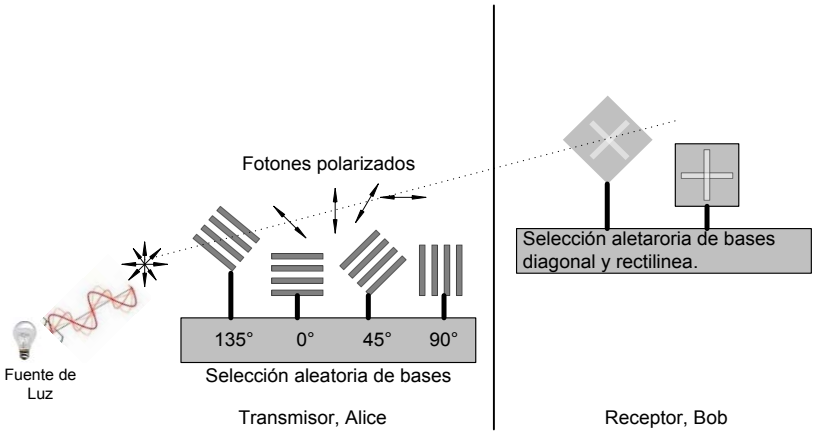


Figura 3. 3: Representación de la Polarización de fotones.

¿Cómo se modula la polarización?, el estado de polarización de un fotón individual es caracterizado por el ángulo de polarización α. Los fotones que emergen de la

fuente de luz con frecuencia tienen un ángulo de polarización desconocido. Para inducir a un ángulo μ° de polarización a un fotón, un filtro de luz es usado talque solamente permite el paso a fotones polarizados en el ángulo μ° [27].

Las bases de polarización son la base lineal (o la base rectilínea) y la denotaremos como B_{+} y la base diagonal (u oblicua) como $B_{\times}$.

En la B_{+} se toma como *bit* lógico 0 al *qubit* con estado de polarización horizontal o con $0°$ y como 1 al vertical o con $90°$. Para la $B_{\times}$ para 0 a $45°$ y para 1 a $135°$. La Figura 3.4 muestra los ángulos de polarización del fotón.

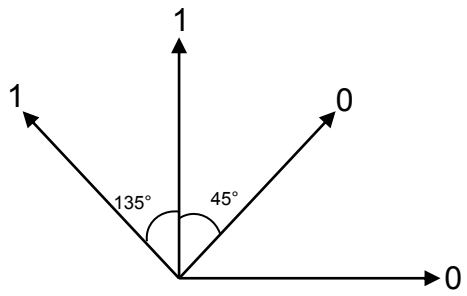


Figura 3. 4: Grados de polarización de los fotones, interpretación bit lógico.

3.15 Protocolos de QKD

Las primeras ideas sobre criptografía cuántica fueron propuestas por Stephen Wiesner a finales de 1960 [17], un estudiante de la universidad de Columbia que intenta publicar una idea a la que llamó dinero cuántico. Su idea no fue tomada en serio en la comunidad científica. Wiesner había tenido la idea de usar leyes de la mecánica cuántica para diseñar billetes imposibles de falsificar. Incorporando un total de 20 trampas de luz en los billetes de un dólar, y codificando en cada una de ellas uno de dos posibles valores con un fotón polarizado. No fue hasta 1983 donde esta idea fue publicada.

Charles H. Bennett que conocía a Wiesner desde que era estudiante, se asoció con Gilles Brassard y trataron de retomar la idea de su amigo para llevarla adelante, obteniendo como resultado en 1984 el primer protocolo de criptografía cuántica, que

sería denominado protocolo BB84 [2], [6], [7], [17]. Basado en los principios de la mecánica cuántica una disciplina relativamente moderna de la física tal protocolo constituye el primer diseño de un sistema criptográfico cuántico.

El término "primera generación" se utiliza para describir los sistemas de QKD que se basan en la transmisión de fotones individuales, o estados coherentes débiles, y la detección utilizando detectores de fotones individuales. El protocolo BB84 es un sistema QKD de primera generación. Ha habido muchos protocolos QKD teóricos posteriores utilizando estados de fotones individuales (o un único *qubit*), algunos protocolos son los siguientes: En 1991 Artur Ekert propone un protocolo QKD basado en pares de fotones entrelazados llamado E91 [33], este protocolo es un protocolo de pares entrelazados conocido como pares EPR.

En 1992 Bennett simplifico el protocolo original BB84 al proponer un protocolo QKD de dos estados no ortogonales conocido como protocolo B92 [34]. En 1998 se propuso un protocolo QKD de seis estados [35], [36].

En el 2004 se propone un protocolo de distribución de claves cuánticas que difiere del protocolo BB84 solo en la parte de la reconciliación de las bases [37]. Este protocolo propone incrementar la seguridad de forma robusta frente al ataque de división del número de fotones o *Photon Number Splitting* (PNS).

Capítulo 4

Protocolo BB84

En este capítulo se aborda uno de los primeros avances de criptografía cuántica y es el protocolo BB84. Este protocolo es propuesto como protocolo para intercambio seguro de claves entre Alice y Bob. Alice y Bob intentan acordar una clave utilizando el cifrado cuántico, la clave resultante o final es utilizada para cifrar información. De lo que se trata es de distribuir claves privadas de manera segura. El protocolo BB84 es un protocolo basado en la transmisión individual de *qubits* que formarán la clave final.
Este protocolo es el primero de una serie de protocolos llamados protocolos QKD (*Quantum Key Distribution*).

La criptografía cuántica hace uso de dos canales de comunicación entre los participantes de una transmisión. Un canal público, este canal es bidireccional. El otro canal es cuántico el cual tiene un único sentido que generalmente es de fibra óptica. La primera realización fue en 1992 [21].

4.1 Canal cuántico del Protocolo BB84

El protocolo BB84 fue propuesto por Bennett y Brasard en la *International Conference on Computers, Systems and Signal* [32], en la propuesta la codificación de la información que transporta el fotón es su polarización.
A continuación se inicia describiendo el protocolo BB84 [2], [6], [7], [24] en dos partes según el canal de comunicación utilizado. Una parte es el canal cuántico y otra la parte es el canal clásico.

Alice y Bob interactúan con un conjunto de *qubits* implementados en fotones, codificados en cuatro estados de polarización. Estos cuatro estados están agrupados de tal forma que se tienen dos bases mutuamente no ortogonales.
Alice y Bob quedan de acuerdo en la codificación asignada a cada polarización en notación binaria.
Alice y Bob también acuerdan el tamaño de la secuencia de fotones que se transmitirán.
Un estado de polarización es correcto al pasar por un polarizador correspondiente a su base, de otra manera la polarización se orienta de manera aleatoria si pasa por un polarizador que tenga otra base.
La siguiente nomenclatura mostrada en la Tabla 4.1 es utilizada en este trabajo en la implementación del protocolo BB84.

Base	Polarización	Estado	Valor del *qubit*
B_+	$\leftrightarrow$	$\|\psi\rangle_{+0}$	Polarización del fotón en horizontal con el valor lógico binario 0.
B_+	$\updownarrow$	$\|\psi\rangle_{+1}$	Polarización del fotón en vertical con el valor lógico binario 1.
$B_\times$	$\nearrow\!\!\swarrow$	$\|\psi\rangle_{\times 0}$	Polarización del fotón en diagonal a la izquierda con el valor lógico binario 0.
$B_\times$	$\nwarrow\!\!\searrow$	$\|\psi\rangle_{\times 1}$	Polarización del fotón en diagonal a la derecha con el valor lógico binario 1.

Tabla 4. 1: Nomenclatura del estado del qubit con la base y polarización de un fotón.

Alice es el transmisor y genera una secuencia aleatoria de fotones polarizados en los cuatro estados, es decir, genera una secuencia aleatoria de bases, a continuación usa el canal cuántico para enviar a Bob un fotón polarizado.

4.2 Canal público del Protocolo BB84

Ahora Bob es el receptor y ha recibido por el canal cuántico la secuencia completa de fotones polarizados generados por Alice. Alice y Bob se comunican por medio del canal público (el canal público o canal clásico debe soportar métodos de autenticación como por ejemplo iniciar en un dominio de red con usuario y contraseña, o con alguna entidad certificadora). Bob le comenta a Alice que bases

ha utilizado, para que ella lo utilice y compare con el registro de sus propias bases utilizadas, para solo tomar en cuenta las bases que coincidieron con las de Bob. Los valores de la polarización o los estados que no coincidieron no son tomados en cuenta.

El canal cuántico se utiliza solamente en el intercambio de claves privadas. Cuando Alice le comunica a Bob los fotones que han sido generados con la misma base, entonces acordada la clave, pueden comenzar a transmitirse información.

Eve el espía puede escuchar la conversación clásica, pero no puede participar en la misma, el canal cuántico está libre o abierto a cualquier manipulación por parte de Eve.

4.3 Seguridad del protocolo BB84

En un caso ideal el canal cuántico está libre de espías, sin ruido en la transmisión o defectos en cifrar y descifrar [3]. Cuando un espía está presente este intenta interceptar la transmisión de fotones. De la misma manera que Bob, este intenta medir las polarizaciones, entonces inevitablemente se causa una perturbación a los mismos. También intentará enviar una copia de los fotones que ha interceptado a Bob, pues intenta generar los fotones originales que generó Alice, pero por el teorema de no clonación esto no es posible.

Con el fin de detectar la presencia de un intruso, Alice y Bob pueden revelar porciones de la clave resultante o la clave generada por medio del canal clásico (público) [3], [4]. Pueden revelar al azar el primer y el último *qubit* por ejemplo para después descartarlos ya que fueron revelados, con esto se puede descubrir si algún valor difiere. Entonces la secuencia completa de la clave no es la correcta y se comprueba la presencia de un espía en la transmisión de fotones por lo tanto esa clave se descarta.

La Figura 4.1 muestra la representación de un esquema para el protocolo BB84 de distribución de claves cuánticas.

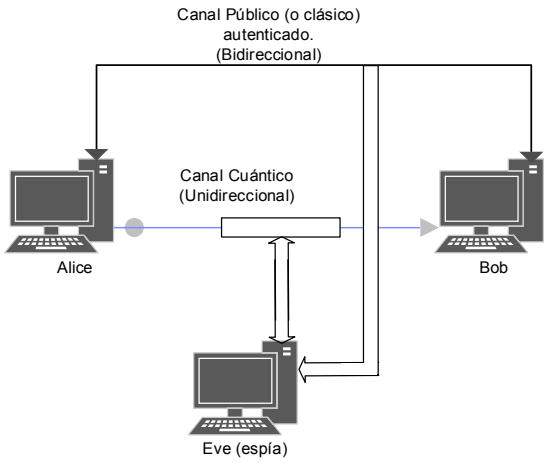


Figura 4. 1: Representación del protocolo de distribución de claves cuánticas BB84.

4.4 Protocolo BB84 transmisión sin Espías

A continuación se expone el protocolo BB84 [2], [3], [4], [17], [21], [24] en una comunicación ideal, es decir, una transmisión libre de ruido y de espías interceptando dicha transmisión en el canal cuántico.

1.- El protocolo comienza con Alice enviado una secuencia de fotones polarizados a Bob que son los *qubits*, a esta secuencia se le denomina la clave inicial. Cada vez que Alice envía un *qubit* a Bob selecciona una de las dos bases (lineal o diagonal), estos son los filtros de polarización, así envía el fotón polarizado en una de los cuatro estados de polarización, $0°$, $90°$, $45°$ y $135°$.

Alice utiliza el canal cuántico para enviar a Bob un fotón polarizado.

Alice registra las bases que utilizó y la polarización con la que fue emitido el fotón.

2.- Bob recibe cada fotón polarizado, Bob tiene que medir cada fotón que recibe, es decir, debe escoger un filtro para cada fotón, para esto selecciona aleatoriamente una de las dos bases (lineal o diagonal). Así Bob mide cada estado recibido del fotón.

Bob registra la secuencia de bases utilizadas al recibir cada fotón y los estados que midió. Algunos fotones pueden no ser recibidos debido a la imperfección de la transmisión y los dispositivos de medición, entonces no se detectan fotones.

Este trabajo no contempla el estudio de las pérdidas de fotones por imperfección.

3.- Terminada la transmisión de fotones por parte de Alice, Alice y Bob se comunican por medio de un canal público, se autentican y a continuación comentan sobre cuales bases utilizaron.

Bob le envía a Alice (por el canal público) el registro de las bases que utilizó con el orden al medir cada fotón recibido, para que Alice lo compare con el registro que utilizó al enviar cada fotón (en la secuencia de transmisión y el orden de las bases). Lo mismo ocurre con Bob, de esta manera Alice y Bob saben con certeza las bases utilizadas en común o las bases que coincidieron en sus mediciones.

En ningún momento Bob revelará el resultado de su medición de los estados cuánticos con las bases utilizadas (los filtros de polarización), solo comunica a Alice el orden y las bases.

4.- Después de realizada la comparación de las bases, Alice y Bob descartan las mediciones (los fotones polarizados) en donde las bases no coincidieron.

Los fotones polarizados que quedaron fueron los generados y medidos con la misma base, por lo que la polarización registrada es la misma para Alice y para Bob. En promedio se descartan la mitad de los fotones que se transmitieron desde Alice. Los fotones que se transmitieron y fueron medidos con las mismas bases son los fotones que conforman la clave secreta final.

4.5 Protocolo BB84 transmisión con Espías

A continuación se expone el protocolo BB84 [2], [3], [4], [17], [21], [24] con la presencia de espías. La finalidad es la siguiente, si un espía intenta medir los fotones que envía Alice de la misma forma que Bob, este no sabe con qué base se generaron. Por lo que para realizar sus mediciones tiene que usar bases al azar. La generación de estas bases al azar introducirá perturbación en los fotones enviados por Alice si no coinciden con su base.

1.- El protocolo comienza con Alice enviado una secuencia de fotones polarizados a Bob que son los *qubits,* esta secuencia se le denomina la clave inicial. Cada vez que Alice envía un *qubit* a Bob selecciona una de las dos bases (lineal o diagonal), estos son los filtros de polarización, así envía el fotón polarizado en una de los cuatro estados de polarización, $0°$, $90°$, $45°$ y $135°$.

Alice utiliza el canal cuántico para enviar a Bob un fotón polarizado.
Alice registra las bases que utilizó y la polarización con la que fue emitido el fotón.
2.- Eve intercepta y mide cada fotón que genera Alice. Entonces Eve selecciona aleatoriamente una de las dos bases (lineal o diagonal) para su medición, el resultado es que Eve genera una serie de estados medidos los cuales son registrados, a continuación estos estados son reenviados a Bob.
3.- Bob recibe cada estado que midió Eve, refiérase a estado como la polarización de un fotón, para el caso los fotones que había enviado Alice fueron previamente medidos por lo que fueron alterados de su estado inicial.
Bob tiene que medir cada fotón que recibe, debe escoger un filtro para cada fotón, para esto selecciona aleatoriamente una de las dos bases (lineal o diagonal). Bob registra la secuencia de bases utilizadas al recibir cada fotón y los estados que midió.
4.- Terminada la transmisión de fotones por parte de Alice, Alice y Bob se comunican por medio de un canal público, se autentican y a continuación comentan sobre cuales bases utilizaron para hacer la reconciliación de las bases. En el proceso de reconciliación de las bases sucede que Alice y Bob coinciden con una base, pero los estados medidos no coinciden, es entonces cuando se reafirma que un espía ha irrumpido en el canal de transmisión cuántico y ocasionó la alteración de los estados iniciales enviados por Alice.
5.- Después de realizada la comparación de las bases, Alice y Bob descartan las mediciones (los fotones polarizados) en donde las bases no coincidieron.

El punto 5.- es importante, porque descartan las mediciones donde las bases no coinciden, pero no se comenta al respecto que sucede con aquellas mediciones donde las bases coinciden, pero lo estados no. En la implementación de este trabajo se descartan las mediciones de los fotones donde sus bases coinciden pero sus estados no coinciden conforme los generados por Alice y medidos por Bob. Esto para mostrar como resultado cual sería la clave final descartando los *qubits* o lo estados que no coincidieron por la presencia de espías.

4.6 Canal Cuántico con Ruido

Un paso importante en la transmisión de datos en criptografía cuántica es la posibilidad de revisar o detectar la captura de información transmitida a través del canal cuántico.

Alice y Bob pueden hacer una revisión al seleccionar aleatoriamente una parte de la clave generada para comparar su igualdad, si algún escucha estuvo presente la clave no coincide.

Una desigualdad en la clave puede deberse a ruido en los detectores de fotones o imperfecciones en el canal cuántico, así como pérdida de los fotones en el recorrido.

Resulta también como ruido la presencia de un escucha externo o un espía llámese Eve. Este se introduce en el canal cuántico y lee información, debido que Alice envía una secuencia de fotones polarizados a Bob Eve tendrá que generar los mismos fotones y reenviarlos a Bob con la finalidad de esconder su presencia.

Con la presencia de Eve las polarizaciones observadas por Bob pueden resultar en no coincidir con las generadas y emitidas por Alice. Este resultado es debido al teorema de no clonación, donde Eve irreversiblemente alterará la polarización de los fotones y no será capaz de reproducirlos exactamente, ocasionando una discrepancia entre las mediciones de Alice y Bob.

El objetivo de Eve es acceder a la información intercambiada entre Alice y Bob.

Debido a que Eve alteró la polarización de un fotón, las mediciones de Alice y Bob son distintas aunque hayan coincidido en sus bases.

¿Cómo el espía Eve lee los fotones que envía Alice? Cuando Eve toca el canal de transmisión (canal cuántico) y logra interceptar los fotones que envía Alice, Eve lo hace con el mismo equipo de recepción de fotones como el que tiene Bob, que en específico es un detector, analiza el estado de polarización de cada fotón para después reenviarlo a Bob, la clave final de Alice y Bob no coincidirán como resultado.

La Figura 4.2 muestra la representación del sistema QKD para el protocolo BB84 con Eve interceptando el canal cuántico de transmisión.

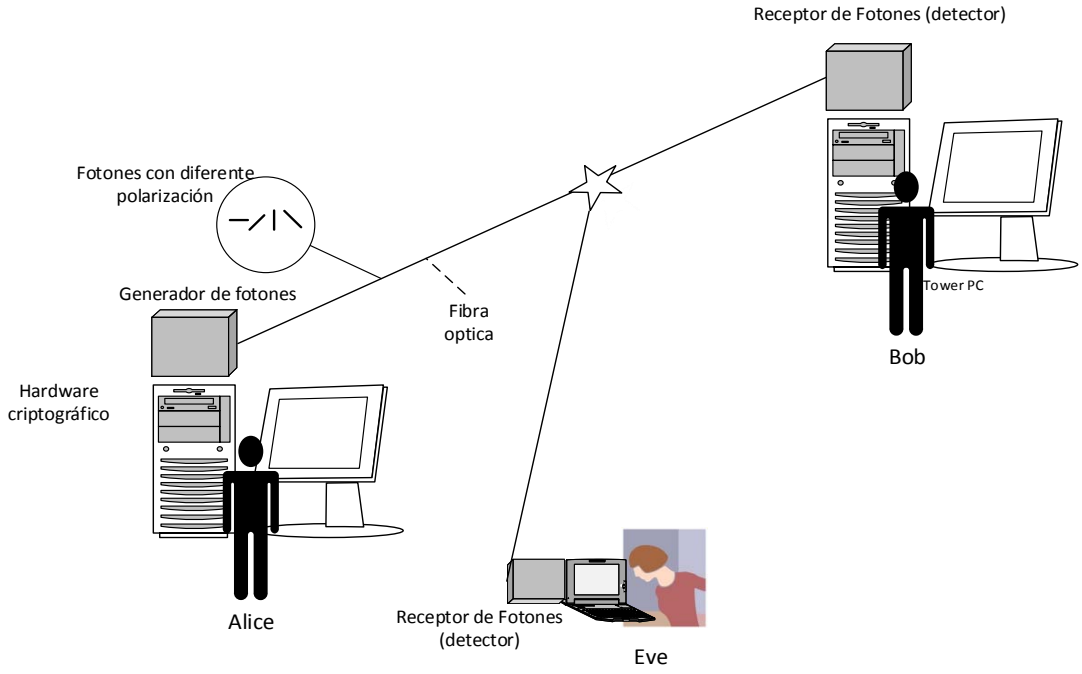


Figura 4. 2: Escenario para un sistema QKD (Quantum Key Distribution) con Eve interceptado en el canal de transmisión cuántico.

4.7 Reconciliación de las Bases

En el caso ideal sin espías o ruido, el resultado del proceso de reconciliación de las bases será una clave final idéntica en ambos extremos de la comunicación (Punto 4.- del subcapítulo 4.4). Es decir, al comunicarse las bases y descartar aquellas que no coinciden con las propias utilizadas para Alice y Bob. Alice y Bob obtienen la misma clave final, porque a bases iguales resulta la medición del mismo estado de polarización del fotón.

La Figura 4.3 muestra el proceso de reconciliar las bases.

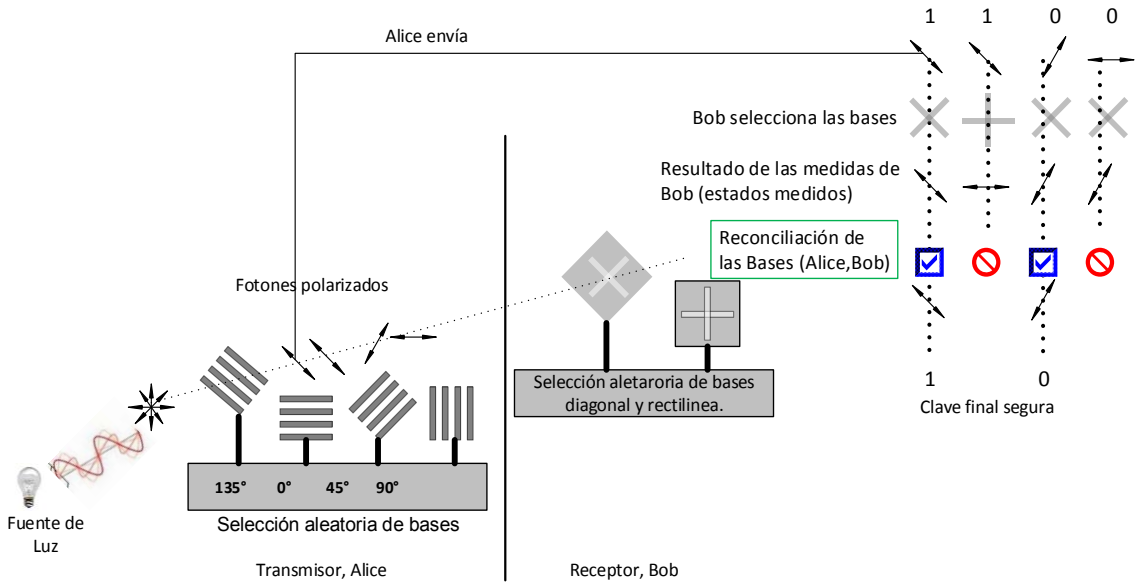


Figura 4. 3: Resultado de la reconciliación de las bases (*sifting stage*) en el protocolo BB84 sin espías.

4.8 ¿Qué sucede con el Protocolo BB84 al detectar un espía?

Si un espía intentó obtener información de los fotones, entonces las secuencias de estados generados y enviados por Alice, con los medidos por Bob no coinciden. Dado que Eve alteró la polarización del fotón, Alice y Bob obtienen distintos estados aunque utilicen las mismas bases.

Entonces hace falta un método o una estrategia para detectar a Eve. Una vez que la clave final es obtenida, remárquese que es la clave donde las bases coinciden, Alice y Bob públicamente vuelven a comunicarse y revelan segmentos de la clave generada. Bob le cuenta a Alice por ejemplo cuales son los primeros *qubits* de su clave o aleatoriamente selecciona un conjunto de *qubits*, lo mismo hace Alice. Debe entenderse que Alice revela una porción de su clave que es igual en tamaño de la que Bob revela que es el resultado de la reconciliación de las bases. El tamaño de la clave revelada (porción) se denomina como $r_{totales}$. Esta porción revelada ya no forma parte de la clave final obtenida causando una reducción de tamaño en la misma.

Al revelar estos segmentos de la clave final, los *qubits* de dichos segmentos se comparan uno a uno, los *qubits* que difieren se van contando y se le denomina $n_{incorrectos}$.

Los *qubits* que difieren son útiles para estimar la tasa de error de la clave final. La estimación de esta tasa de error proporciona un indicador para determinar si la comunicación entre Alice y Bob se detiene y por efecto se desecha la clave entera o si el protocolo se repite sobre un canal cuántico diferente o si se inicia la corrección de errores. La corrección de errores queda fuera del objetivo de este trabajo de tesis, sin embargo existe un protocolo llamado *Cascade* [18], [21], [38]. Definido en 1994 por Brassard y Louis Salvail basado en el intercambio de paridades por bloque, el objetivo de este protocolo es construir un procedimiento de corrección de errores que proporcione una cantidad de información mínima. Más procedimientos para la corrección cuántica de errores pueden consultarse en la referencia [24].

4.9 Estimación de Error

Se mencionó la estimación de una tasa de error. Alice y Bob deben estimar la tasa de error cuántico o QBER (*Quantum Bit Error Rate*). EL cálculo de QBER es un indicador para detectar posibles espías. El cálculo de QBER debe ser comparado con un umbral o límite de error $p_{\max}$ del cual no debe exceder, es decir, $p_{QBER} \leq p_{\max}$ [21]. Si la condición es verdadera, entonces que se continúa con la corrección de errores, la corrección de errores queda fuera del objetivo de este trabajo de tesis.

Si $p_{QBER} > 3\%$, se detiene la comunicación entre Alice y Bob por el canal cuántico y se repite el protocolo.

En este trabajo de tesis se establece $p_{\max}$ del 3%, es decir, si $p_{QBER} > 3\%$, se detiene la comunicación entre Alice y Bob por el canal cuántico y se repite el protocolo.

Entonces el cálculo de QBER, entiéndase también la probabilidad de error, es:

$$p_{QBER} = \frac{n_{incorrectos}}{r_{totales}}$$

Donde $r_{totales}$ representa el tamaño de la porción revelada de la clave final y $n_{incorrectos}$ representa el número de *qubits* diferentes encontrados en la porción revelada.

El tamaño de la porción a revelar es del 8% de la clave final. Se toman entonces bloques del tamaño del 8%.

El cálculo de la porción a revelar es el siguiente:

$$r_{totales} = (.08 * longitud \text{ (de la clave final)}) * 3$$

El tamaño de la porción a revelar se triplica, la razón es asegurar o detectar la presencia de más *qubits* que no coincidan en dichas porciones. Es decir, se está aumentando un poco más el tamaño de la porción a revelar. Para el cálculo de $r_{totales}$ se toma la parte entera sin decimales, si se obtiene un valor del $.96$ el tamaño de la porción a tomar es cero.

4.10 Estrategias de Ataques sobre los Protocolo QKD

Tres tipos de ataques son aplicados a los protocolos QKD y son: ataque de Intercepta y Reenvía, ataque de División del número de fotones, ataque de División del número de fotones e Intercepta y Reenvía. Los dos últimos ataques quedan fuera del alcance de este trabajo.

En este trabajo la implementación del protocolo BB84 con presencia de espías es un ataque de Intercepta y Reenvía [29].

Es decir en el ataque el intruso:

1.-Intercepta los fotones enviados y hace una medida de los mismos al escoger una de las dos bases (Rectilínea o Diagonal) de forma independiente y aleatoria.

2.-Renvia el nuevo fotón al receptor (Bob).

3.-Escucha el canal público durante la reconciliación de las bases.

La estimación de QBER causado por este ataque es del 25%.

4.11 Clave Secreta generada por el Protocolo BB84

Alice y Bob no deciden cual es la clave que van a utilizar para la transferencia de información. Al iniciar y finalizar el protocolo BB84 ocurre entonces que ya han definido dicha clave, esta clave puede utilizarse para el cifrado de información con *one-time pad*.

El cifrado de Vernam es llamado *one-time system* o *one-time pad* si la clave es escogida aleatoriamente y no se utiliza nuevamente.

Si se desea utilizar el cifrado de Vernam se requiere una clave aleatoria de, al menos, la misma longitud que el mensaje. Entonces la clave es sumada realizando un XOR al mensaje, de esta manera se obtiene el mensaje cifrado. Para cada cifrado o sesión, Alice y Bob comparten una clave nueva y aleatoria. Es entonces cuando se dice que el cifrado de Vernam es perfectamente seguro, y su vez es un subgrupo de cifrados seguros incondicionalmente.

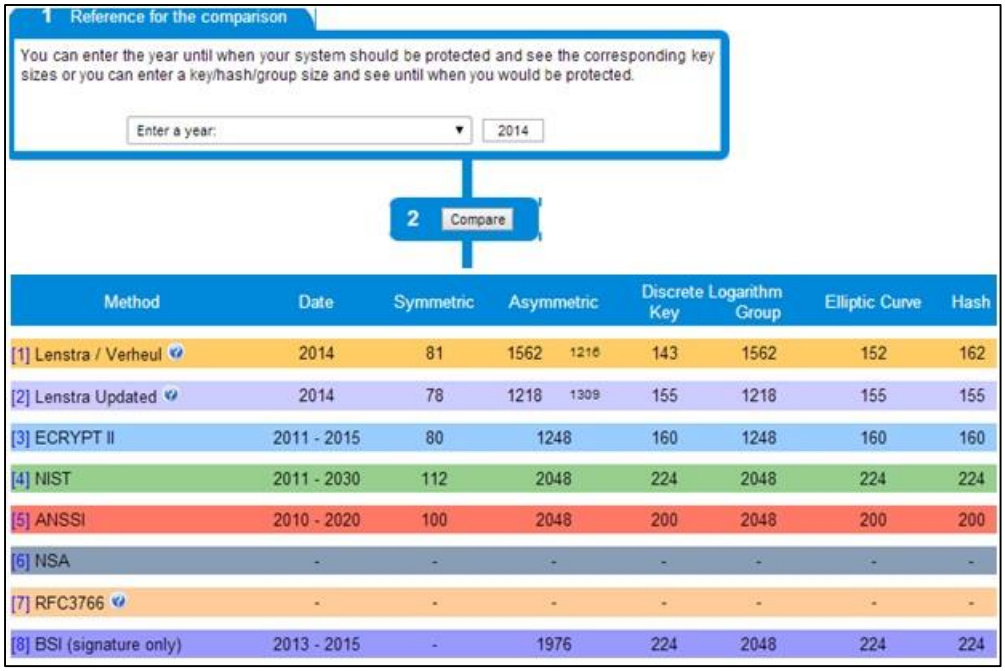
En el protocolo, el canal cuántico es utilizado para el intercambio de las claves privadas, el resto de información es transmitido por el canal público.

Alice y Bob han acordado la clave, entonces pueden iniciar una transmisión de información de manera segura.

La NIST (*The National Institute of Standards and Technology*) recomienda como fortaleza mínima el uso de claves de 112 *bits* para algoritmos simétricos, el uso de este tamaño en la clave abarca a partir del año 2011 hasta el año 2030.

El sitio *Keylength* [30] es una página de Web que ofrece una comparación interactiva entre los tamaños de clave que utilizan algunos algoritmos criptográficos, dicha comparación está hecha por diferentes instituciones como la NIST.

La Figura 4.4 muestra la comparación para el año 2014 que hacen diferentes instituciones en recomendación del tamaño mínimo de la clave para los algoritmos.



Method	Date	Symmetric	Asymmetric		Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash
[1] Lenstra / Verheul	2014	81	1562	1216	143	1562	152	162
[2] Lenstra Updated	2014	78	1218	1309	155	1218	155	155
[3] ECRYPT II	2011 - 2015	80	1248		160	1248	160	160
[4] NIST	2011 - 2030	112	2048		224	2048	224	224
[5] ANSSI	2010 - 2020	100	2048		200	2048	200	200
[6] NSA	-	-	-		-	-	-	-
[7] RFC3766	-	-	-		-	-	-	-
[8] BSI (signature only)	2013 - 2015	-	1976		224	2048	224	224

Figura 4. 4: Tamaños de clave mínimos recomendados por varias instituciones como la NIST en el año 2014.

Capítulo 5

Resultados de la Implementación del Protocolo BB84

En este capítulo se presentan pruebas del desempeño del protocolo BB84, ejecutando las pruebas se analizan los resultados. Para ello se hace la implementación por medio de código en el lenguaje de programación C y C#. Se presentan resultados para los casos en los que el protocolo se ejecuta con espías y sin espías.

Una aproximación a la simulación de la mecánica cuántica, especialmente en computación cuántica, se encuentra en LibQuantum, que es una librería de programas realizados en el lenguaje de programación C. Provee la estructura para un registro cuántico y operaciones para la manipulación de un registro cuántico.

A continuación la Tabla 5.1 muestra un resumen de algunos ambientes simuladores de computación cuántica.

Nombre	Lenguaje de Programación	Última actualización
LibQuantum	C	2008
QCL	C/PASCAL/OWN	2006
Eqcs	C++	2012
Q++	C++	2009
jQuantum	Java	2011
PyQu	Python	2009
QFC	MATLAB	2009

Tabla 5. 1: Algunos simuladores de Computación Cuántica.

5.1 Ejecución del Protocolo BB84 sin Espías con LibQuantum en C

Este escenario previamente fue descrito en el subcapítulo 4.4, el escenario considera que la comunicación entre Alice y Bob está libre de espías, es decir, en

la transmisión por el canal cuántico no se encuentra algún espía a la espera de interceptar *qubits*.

En este escenario se considera lo siguiente, el *qubit* se encuentra en superposición de dos estados. Entonces se inicializa un registro con un *qubit*, después al *qubit* se le aplica la transformación Hadamard para ponerlo en un estado de superposición. Posteriormente se destruirá dicha superposición haciendo una medición u observación al *qubit* para obtener el resultado de la medida, de esta forma se obtienen el valor de cada *qubit* que envía Alice a Bob.

Para llevar a cabo esta ejecución se muestra la implementación en el lenguaje de programación C usando LibQuantum de la cual se modifican las siguientes funciones [28]:

`quantum_reg reg;` se nombra el registro como reg.

`reg = quantum_new_qureg(0, 1);` se inicializa el registro, el primer parámetro es el valor al cual inicializamos el registro al comienzo, es decir 0, y el segundo parámetro es el número de *qubits* con el cual el registro inicia. Así el registro se inicia con el estado $|0\rangle$ y con un *qubit* en él.

`quantum_hadamard(0, &reg);` al *qubit* se le aplica la puerta Hadamard, cambiando al estado de superposición $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$.

`result = quantum_bmeasure(0, &reg);` se hace la medida del *qubit*, al medirlo se destruye la superposición. Esta función hace la medida para un solo *qubit*. El primer parámetro es la posición del *qubit*, cabe señalar que en esta notación el cero es el menos significativo. Finalmente se imprime la variable *result* para conocer el resultado de la medición.

Las funciones anteriores se ejecutan haciendo la llamada a la librería `#include <quantum.h>`.

A continuación se muestra la implementación ejecutándose en el lenguaje de programación C simulando el protocolo BB84.

La base B_+ se representa con el número 1.

La base $B_\times$ se representa con el número 0.

El *qubit* $|0\rangle$ será interpretado como un bit con valor 0.

El *qubit* $|1\rangle$ será interpretado como un bit con valor 1.

Alice envía 8 *qubits* a Bob, la Figura 5.1 muestra como Alice genera cada *qubit* alternando entre sus bases de manera aleatoria.

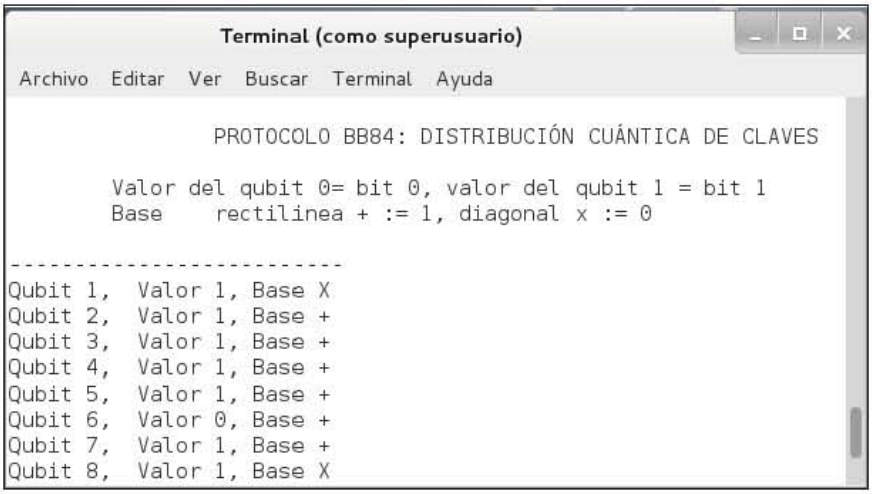


Figura 5. 1: Envío de 8 qubits, protocolo BB84 con LibQuantum en C.

La Figura 5.2 muestra las bases con las que mide Bob cada *qubit* enviado por Alice. Para después hacer la reconciliación de las bases (coincidencia de las bases) y obtener la clave final.

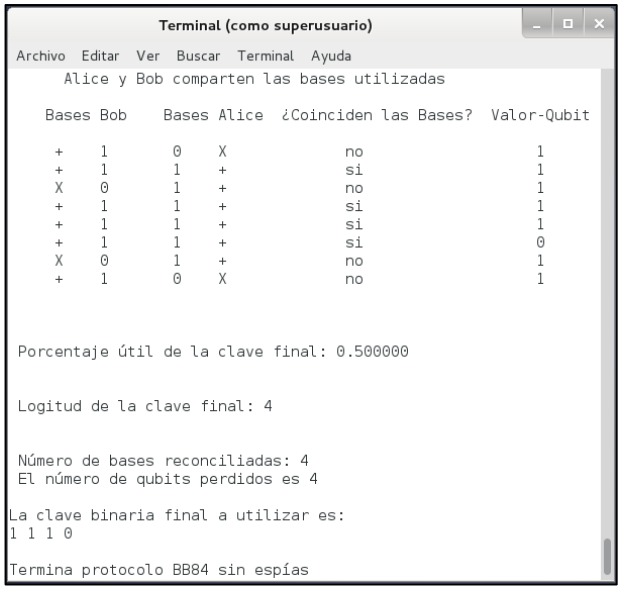


Figura 5. 2: Clave generada del envío de 8 qubits, protocolo BB84 con LibQuantum en C.

5.1.1 Resultados de la Ejecución del Protocolo BB84 sin espías con LibQuantum en C

A continuación se ejecutan diez simulaciones con el envío de 8 *qubits*. La Tabla 5.2 muestra en promedio el porcentaje de la clave final, el promedio fue del 45%. Esto significa que un porcentaje de la clave inicial se está perdiendo o sacrificando en la ejecución del protocolo BB84, reduciendo su longitud. Observando dichos resultados se obtiene que la clave final no se siempre es del mismo tamaño.

No. Simulación	Longitud final de la clave	Porcentaje de clave útil
1	3	37.50%
2	4	50.00%
3	2	25.00%
4	2	25.00%
5	7	87.50%
6	5	62.50%
7	2	25.00%
8	4	50.00%
9	3	37.50%
10	4	50.00%
	Promedio:	45.00%

Tabla 5. 2: Resultado del promedio para diez simulaciones de una clave inicial de 8 qubits, con el protocolo BB84 sin espías implementado con LibQuantum en C.

Al reducir su longitud implica que hay que tener en cuenta la longitud de la clave inicial para obtener claves seguras y eso se traduce en seguir las recomentaciones de la NIST. El tamaño mínimo establecido por la NIST para la clave en algoritmos simétricos es de112 *bits* [30].

Para sondear esta recomendación a continuación se obtienen los resultados de una simulación para cuatro longitudes iniciales propuestas de la clave, la Tabla 5.3 muestra el resultado.

Envío inicial de *qubits.* Tamaño inicial de la clave	Tamaño de la clave final	*Qubits* perdidos
16 *qubits*	8	8
128 *qubits*	77	51
256 *qubits*	119	137
4096 *qubits*	2077	2019

Tabla 5. 3: Resultado de las longitudes de las claves finales de una simulación con el protocolo BB84 sin espías implementado con LibQuantum en C.

Se observa en la Tabla 5.3 que una clave inicial propuesta de 256 *qubits* se acerca a la recomendación de la NIST para el tamaño mínimo de la clave en algoritmos simétricos. Pero para analizar más a fondo si se obtiene el mismo resultado ahora

se evalúa la clave inicial de 256 *qubits* para diez simulaciones. La Tabla 5.4 muestra el resultado de estas simulaciones.

No. Simulación	Longitud final de la clave	Porcentaje de clave útil
1	144	56.25%
2	120	46.88%
3	117	45.70%
4	134	52.34%
5	125	48.83%
6	133	51.95%
7	126	49.22%
8	132	51.56%
9	132	51.56%
10	134	52.34%
	Promedio:	50.66%

Tabla 5. 4: Resultado del promedio para diez simulaciones de una clave inicial de 256 qubits, con el protocolo BB84 sin espías implementado con LibQuantum en C.

De igual forma se continuó ejecutando el protocolo BB84 hasta llegar a 100 simulaciones de la cuál en la simulación número 22 solo se presentó una longitud de clave final de 112 *qubits*.

Para claves en algoritmos asimétricos la NIST recomienda como tamaño mínimo establecido 2048 *bits*. Se necesita entonces enviar el doble de 2048 *qubits* para esta especificación. La Tabla 5.5 muestra el resultado de diez simulaciones para una clave inicial de 4096 *qubits*.

No. Simulación	Longitud final de la clave	Porcentaje de clave útil
1	1990	48.58%
2	2077	50.71%
3	2063	50.37%
4	2078	50.73%
5	2055	50.17%
6	2047	49.98%
7	2077	50.71%
8	2050	50.05%
9	2025	49.44%
10	1975	48.22%
	Promedio:	49.90%

Tabla 5. 5: Resultado del promedio para diez simulaciones de una clave inicial de 4096 qubits, con el protocolo BB84 sin espías implementado con LibQuantum en C.

Con estos resultados (Tabla 5.4 y Tabla 5.5) se puede observar que entre más larga sea la clave inicial el porcentaje de clave final útil se aproxima al 50% en promedio, lo que implica que la mitad en promedio de la clave inicial se pierde.

Ahora bien, la implementación con LibQuantum en este subcapítulo ofrece la oportunidad de mostrar una comparación con la implementación en C# para el escenario sin espías. La gráfica de la Figura 5.3 muestra una comparación de simular con LibQuantum y con C# para la distribución cuántica de claves sin espías. En esta ejecución se tomó el promedio de tres simulaciones para cada clave inicial. Las claves inicales fueron de 16 *qubtis*, 32 *qubits*, 64 *qubits*, 128 *qubits*, 256 *qubits*, 1024 *qubits,* 2048 *qubits*. Se puede observar como a medida que las claves iniciales son más largas, los resultados en las simulaciones con C# se aproximan a los resultados con LibQuantum. Se afirma entonces como LibQuantum muestra el comportamiento de una computadora cuántica al simular el protocolo BB84, es entonces que los siguientes resultados y ejecuciones mostrados en este trabajo se continuaron con el desarrollo en C#.

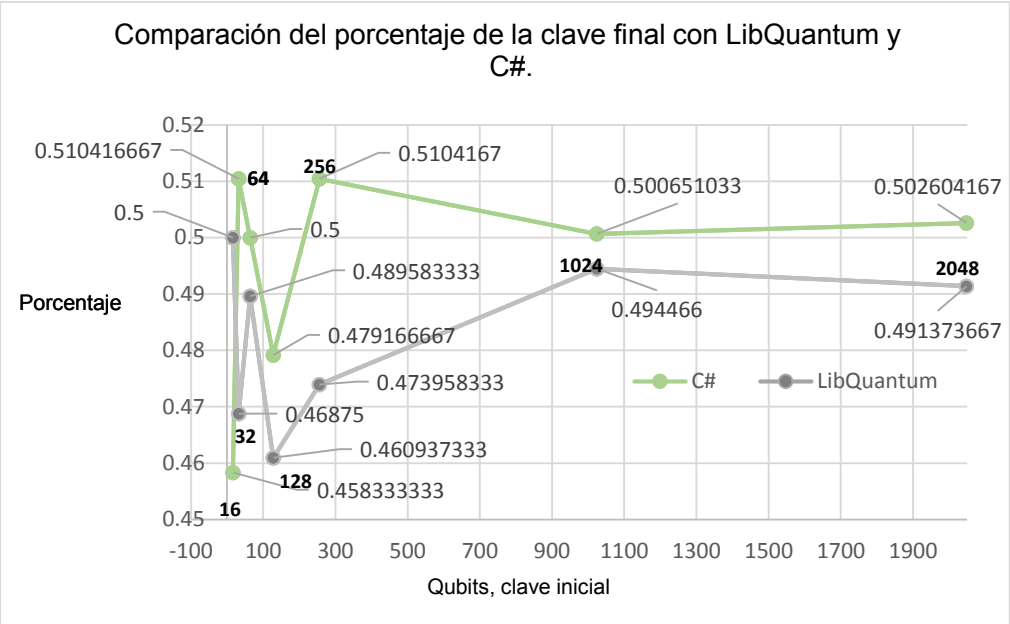


Figura 5. 3: Gráfica de porcentaje útil de la clave inicial para obtener la clave final con el protocolo BB84 comparando la implementación en LibQuantum y C# (sin espías).

5.2 Ejecución del Protocolo BB84 sin Espías y con Espías en C#

El escenario para el protocolo BB84 de la cual la transmisión se hace sin espías y con espías fue descrito en los subcapítulos 4.4 y 4.5 respectivamente.

En los escenarios se considera lo siguiente:

Un *qubit* es representado por un fotón polarizado, la polarización es el grado de inclinación que se provoca en dicho fotón, Alice enviará fotones de forma individual.

La Tabla 5.6 muestra la nomenclatura del estado del *qubit* y su polarización que se usa en estos escenarios.

Base	Polarización	Estado Qubit	Valor del *qubit*	Base
B_+	- 0°	$\|0\rangle$	Polarización horizontal. Valor lógico binario 0.	lineal
B_+	| 90°	$\|1\rangle$	Polarización vertical. Valor lógico binario 1.	
$B_\times$	(45°	$\|0\rangle$	Polarización derecha. Valor lógico binario 0.	diagonal
$B_\times$	) 135 °	$\|1\rangle$	Polarización izquierda. Valor lógico binario 1.	

Tabla 5. 6: Nomenclatura del estado del qubit --base y polarización de un fotón-- para la ejecución del protocolo BB84 sin espías y con espías implementado en C#.

Cabe mencionar que la implementación en C# no hace uso de librerías cuánticas, pues estas no están definidas en el lenguaje de programación. La implementación en C# se hace desde generar los estados polarizados del fotón, que serán interpretados como *qubits*. En esta implementación el *qubit* no está en estado de superposición de dos estados, además no se puede clonar (subcapítulo 3.12).

5.2.1 Ejecución del Protocolo BB84 sin Espías en C#

A continuación se muestra la implementación del protocolo BB84 sin espías ejecutándose con el lenguaje de programación C#. En este escenario Alice inicia el protocolo generando una clave inicial de 128 *qubits*. La Figura 5.4 muestra la simulación del protocolo BB84, donde puede observarse la secuencia de fotones polarizados (*qubits*) que envía Alice resultado de las bases utilizadas para generarlos.

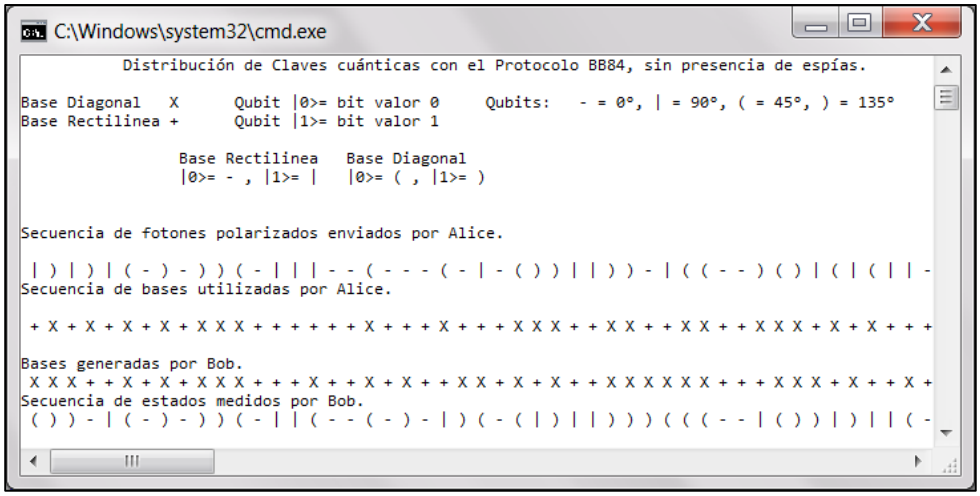


Figura 5. 4: Envío de 128 qubits, protocolo BB84 sin espías, implementación en C#.

En el momento que Alice envía un *qubit* y este es medio por Bob en su recepción el protocolo BB84 inicia la distribución de la clave.

Ahora bien Bob ha medido cada fotón recibido y comienza la reconciliación de las bases. La Figura 5.5 muestra la reconciliación de las bases entre Alice y Bob.

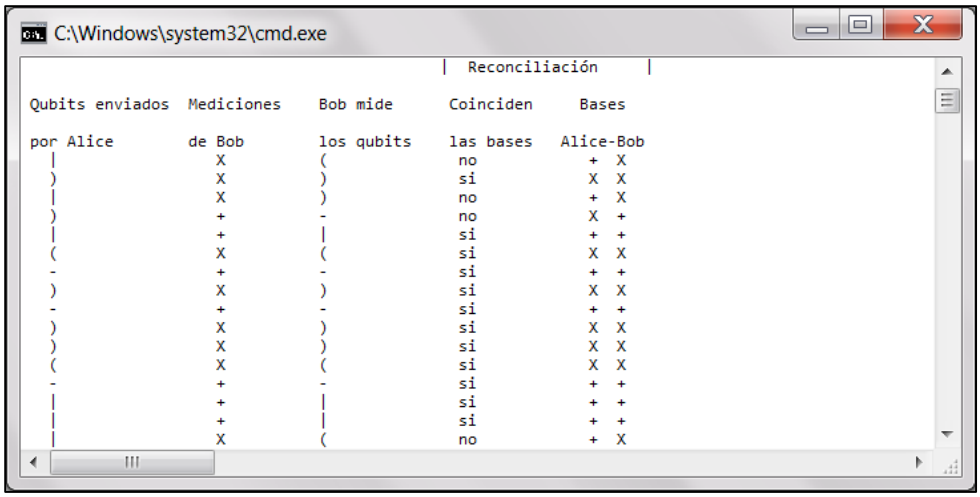


Figura 5. 5: Reconciliación de las bases, protocolo BB84 sin espías, implementación en C#.

5.2.1.1 Resultados de la Ejecución del Protocolo BB84 sin espías en C#

Como resultado de la ejecución del protocolo BB84 se obtiene la clave final, la Figura 5.6 muestra la clave final generada a partir de una clave inicial de 128 *qubits*.

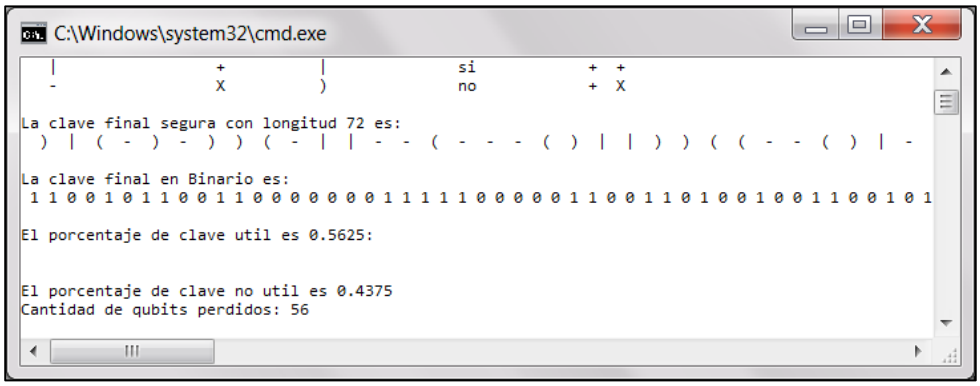


Figura 5. 6: Clave generada con el Protocolo BB84 sin espías, implementación en C#.

A continuación se presenta el resultado de ejecutar el protocolo BB84 sin la presencia de espías. La ejecución se hace diez veces, en estos resultados se analiza que sucede con la clave final, la clave inicial es de 128 *qubits*. La Tabla 5.7 muestra los resultados.

No. Simulación	Longitud final de la clave	Porcentaje de clave útil	Clave generada
1	63	49.22%	0 1 1 0 1 1 1 1 0 1 1 0 0 0 1 1 0 1 1 0 1 1 1 1 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 1 1 0 1
2	51	39.84%	
3	63	49.22%	0 0 0 1 1 0 0 1 0 1 1 1 0 1 0 0 1 0 1 0 1 1 1 0 0 0 0 1 1 1 0 1 1 0 1 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 1 0 1 1 0 0 1
4	63	49.22%	0 1 1 1 0 1 0 1 0 0 0 0 1 0 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 0 1 0 0
5	62	48.44%	
6	63	49.22%	1 0 0 1 0 1 0 1 1 1 0 1 1 1 1 1 0 0 0 0 1 0 1 1 1 0 1 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0 1 1 1 0 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 1
7	68	53.13%	0 0 0 1 0 0 1 0 0 0 0 1 1 1 1 1 1 1 0 1 1 0 1 1 0 0 0 1 0 1 0 0 0 1 0 0 1 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 0 1 0 0 1
8	68	53.13%	1 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 1 1 1 1 1 1 1 0 0 1 0 1 1 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 1 1 0 0
9	65	50.78%	
10	67	52.34%	
	Promedio:	49.45%	

Tabla 5. 7: Resultado de las claves generadas (diez simulaciones) de una clave inicial de 128 qubits con el protocolo BB84 sin espías, implementación en C#.

En los resultados de la Tabla 5.7 el protocolo BB84 también genera claves de un mismo tamaño, se puede apreciar cómo se han generado diez claves, seis de las claves tienen el mismo tamaño, aunque tienen el mismo tamaño se encontró que las claves no son iguales. Pero ¿Qué sucede cuando el protocolo ha generado claves iguales?, este hecho se puede ver en los resultados de la Tabla 5.8, donde se ejecutaron más de 11 simulaciones con una clave inicial de 8 *qubits*.

No. Simulación	Longitud final de la clave	Clave generada
1	4	0 1 1 1
2	4	1 1 1 0
3	4	1 1 0 0
4	5	0 1 1 1 1
5	3	1 1 1
6	5	1 1 1 0 1
7	5	0 0 1 0 0
8	3	0 1 0
9	2	1 1
10	5	0 1 0 1 0
11	4	0 1 1 1

Tabla 5. 8: Resultado de Claves generadas iguales de una clave inicial de 8 qubits con el protocolo BB84 sin espías, implementación en C#.

En la simulación número 1 y número 11 se encontró que el protocolo BB84 generó claves iguales, esto puede interpretarse a causa del tamaño inicial de la clave.

Al iniciar el protocolo BB84 la clave inicial de 8 *qubits* pierde en promedio la mitad de esos *qubits*, para el caso donde la clave final es de 4 *qubits* se tienen 2^4 posibles valores para esa clave, propiciando con más frecuencia obtener combinaciones iguales para la clave final. ¿Esto qué significa?, el protocolo BB84 no queda exento de distribuir claves iguales, pero con claves finales mayores, háblese por ejemplo de claves de 112 *qubits*, mayor es el número de combinaciones que incluso un atacante probaría si usa un ataque por fuerza bruta y la frecuencia de obtener combinaciones iguales se reduce considerablemente. Para el resultado de 50 simulaciones con una clave inicial de 128 *qubits,* el promedio de la longitud de la clave final fue 63.76 y la frecuencia de encontrar claves iguales fue 0.

5.2.2 Ejecución del Protocolo BB84 con Espías en C#

A continuación se muestra la implementación del protocolo BB84 con espías ejecutándose con el lenguaje de programación C#. En este escenario la secuencia

de *qubits* (fotones) generados por Alice como clave inicial son expuestos a la presencia de algún espía en el canal cuántico.

La Figura 5.7 muestra la secuencia de *qubits* generada por Alice, la secuencia es de 128 *qubits*.

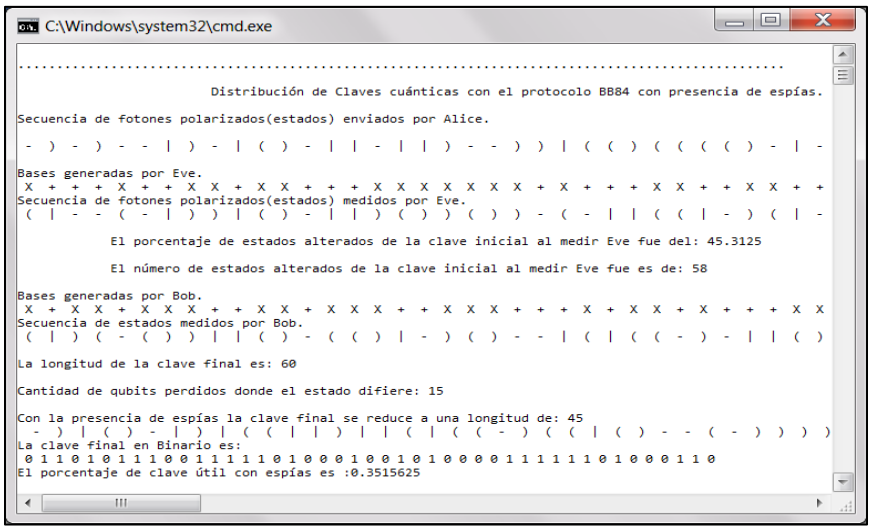


Figura 5. 7: Clave generada con el Protocolo BB84 con espías, implementación en C#.

En la Figura 5.7 se observa que Eve el espía intercepta y mide cada fotón que Alice envía a Bob, y genera una nueva secuencia de fotones medidos. Esta secuencia de fotones medidos por Eve no se parece a la secuencia que envía Alice lo que significa que Eve al medir cada fotón de Alice ha provocado alterar su estado. Ahora Bob recibe la secuencia de fotones medidos por Eve como si fuera la secuencia que le envía Alice.

La Figura 5.8 muestra las bases con las que mide Bob cada *qubit* enviado por Alice. Para después hacer la reconciliación de las bases (coincidencia de las bases) y obtener la clave final. Obsérvese ahora un detalle, aquellas bases de Alice y Bob que si coinciden, el estado del *qubit* (fotón) medido no coincide. La causa es la alteración que Eve provocó al interceptar y medir el envío de Alice.

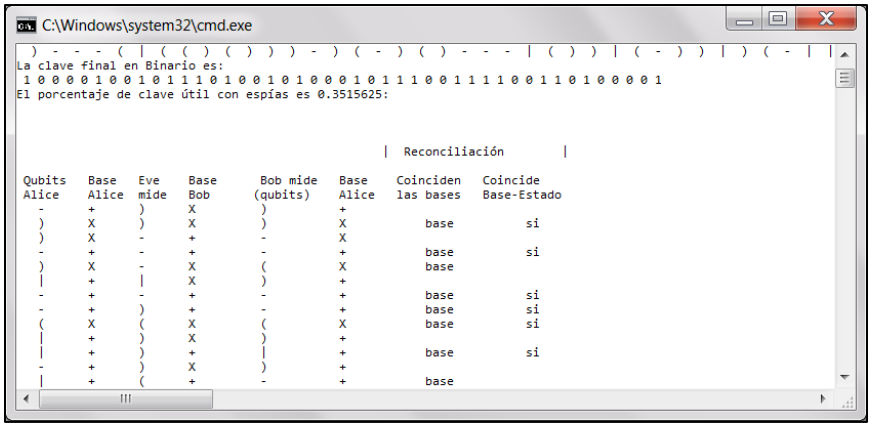


Figura 5. 8: Reconciliación de las bases donde con bases iguales se tienen estados diferentes, estas mediciones no son parte de la clave final del envío de 128 qubits, implementación del protocolo BB84 sin espías en C#.

5.2.2.1 Resultados de la Ejecución del Protocolo BB84 con espías en C#

El comportamiento de distribuir una clave con espías y sin espías se aborda en estos resultados. Se comparan los promedios de la clave final en ambos escenarios. Para mostrar estos primeros resultados se realizan diez simulaciones del protocolo BB84 con presencia de espías y sin espías, estos resultados se muestran en la Tabla 5.9 con una clave inicial de 128 *qubits*.

	Porcentaje de clave útil	
No. Simulación	Sin espías	Con espías
1	46.88%	35.16%
2	49.22%	41.41%
3	56.25%	43.75%
4	53.91%	36.72%
5	49.22%	35.94%
6	51.56%	43.75%
7	45.31%	35.16%
8	50.78%	35.16%
9	50.00%	39.06%
10	54.69%	40.63%
	50.78%	**38.67%** **Promedio**

Tabla 5. 9: Resultados de los promedios del porcentaje de clave útil (con espías y sin espías) con el protocolo BB84 con una clave inicial de 128 qubits, implementación en C#.

En la Tabla 5.9 se muestra el promedio de clave útil sin espías y con espías en el canal cuántico de comunicación. Se encuentra una diferencia en ambos promedios, el promedio de clave útil con presencia de espías es siempre menor que sin espías. Esto significa que los fotones que envía Alice fueron alterados por Eve al espiar el canal cuántico, esta actividad se ve reflejada en la reconciliación de las bases.

El protocolo BB84 genera claves finales en el proceso de la reconciliación de las bases. La implementación del protocolo BB84 en este trabajo con el escenario de la presencia de espías genera claves finales cuando Alice y Bob se comunican las bases para hacer la reconciliación. En el proceso de reconciliación se revisa si al encontrar bases iguales los estados también los son, de lo que resulta que los estados medidos no necesariamente también fueron iguales. Lo que implica que estos estados serán descartados y no forman parte de la clave final, se consideran entonces *qubits* perdidos. Entonces no solo se hace una reconciliación de las bases, sino también la reconciliación de sus estados como parte de la implementación.

La Figura 5.9 muestra el resultado de reconciliar las bases en el protocolo BB84, donde se muestra que aunque las bases coincidieron los estados no lo hicieron y la clave final no es la misma para Alice y para Bob.

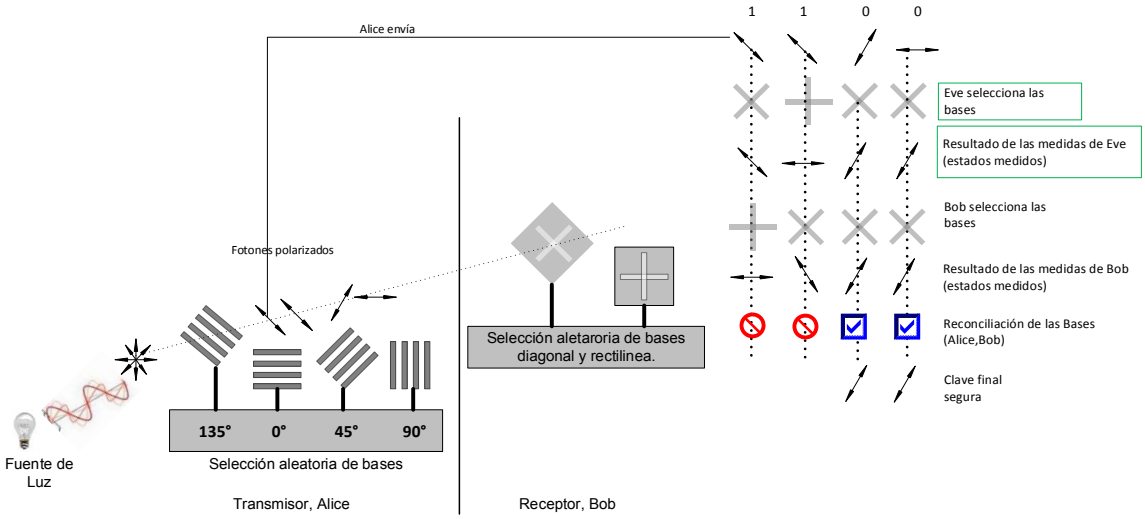


Figura 5. 9: Resultado de la reconciliación de las bases (*sifting stage*) en el protocolo BB84 con espías.

La Tabla 5.10 muestra el resultado de los *qubits* perdidos en comparación al ejecutar el protocolo BB84 con espías y sin espías para una clave inicial de 128 *qubits*. En promedio se están perdiendo 16 *qubits* más.

	Qubits perdidos	
No. Simulación	Sin espías	Con espías
1	68	83
2	65	75
3	56	72
4	59	81
5	65	82
6	62	72
7	70	83
8	63	83

9	64	78	
10	58	76	
	63	79	**Promedio**

Tabla 5. 10: Resultado del promedio de los qubits perdidos con una clave inicial de 128 qubits en la simulación del protocolo BB84, implementación en C# sin espías y con espías.

La Figura 5.10 muestra gráficamente la diferencia del número de *qubtis* perdidos cuando la comunicación está libre de espías y con presencia de espías. Se observa que los *qubits* perdidos en el primer escenario son menos que en el escenario del protocolo BB84 con espías. Claramente se nota como la intervención de Eve el espía, altera considerablemente la clave inicial con la que Alice inicia.

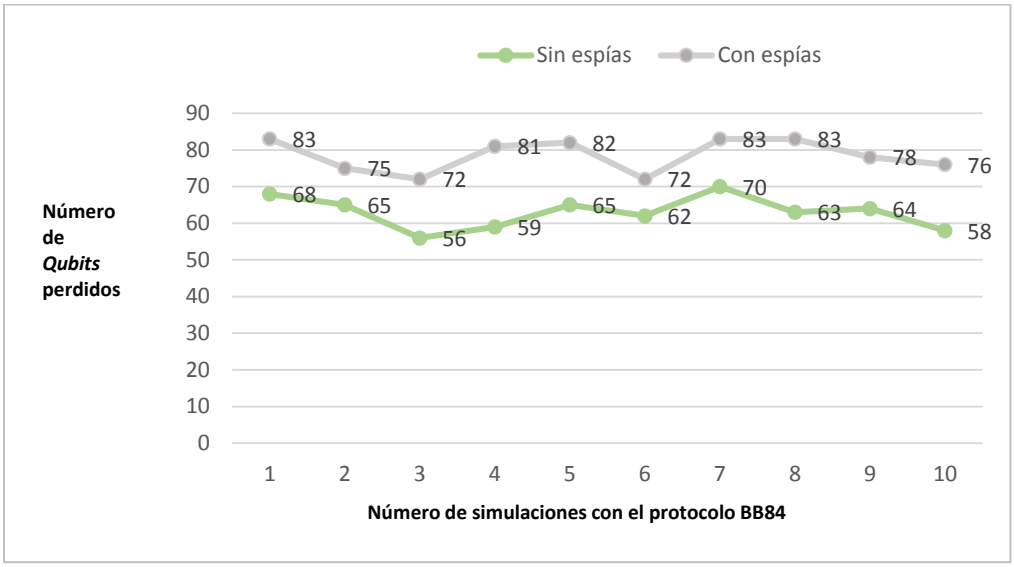


Figura 5. 10: Resultado de los qubits perdidos, implementación del protocolo BB84 en C# con espías y sin espías.

5.2.3 Ejecución del Protocolo BB84 con espías en C#, estimación del Error

En este escenario la secuencia de *qubits* (fotones) generados por Alice como clave inicial son expuestos a la presencia de algún espía en el canal cuántico llamado Eve. Ahora después de que Eve ha interceptado el canal, se calcula la estimación del error que indica si el protocolo se reinicia debido a su presencia.

A continuación se ejecuta el protocolo BB84 con una clave inicial de 16 *qubits*. La Figura 5.11 muestra que la clave final resultante es de 10 *qubits* (resultado de la reconciliación de las bases --10 bases).

Entonces el valor de la porción a revelar es de cero, recuérdese el subcapítulo 4.9 donde se calcula la porción a revelar de la clave final, como la porción a revelar fue cero, no se calculó el valor de QBER. Aunque la porción fue cero, la implementación del protocolo BB84 indica como esté debe reiniciarse.

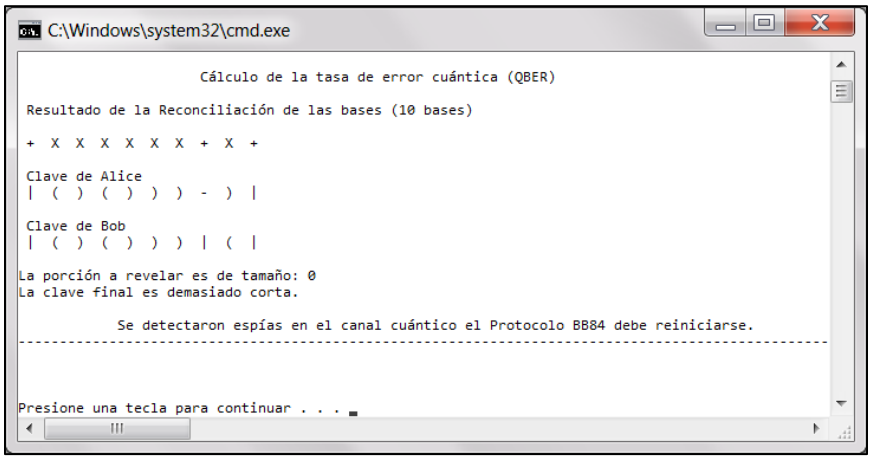


Figura 5. 11: El valor de la porción revelada igual a cero para el cálculo de la tasa de error cuántico (QBER), ejecución con 16 qubits de clave inicial, implementación en C#.

A continuación se ejecuta el protocolo BB84 pero con una clave inicial de 64 *qubits*. La Figura 5.12 muestra el resultado de la estimación del error (QBER). El tamaño de la porción a revelar (o bloque) reduce el tamaño de la clave final.

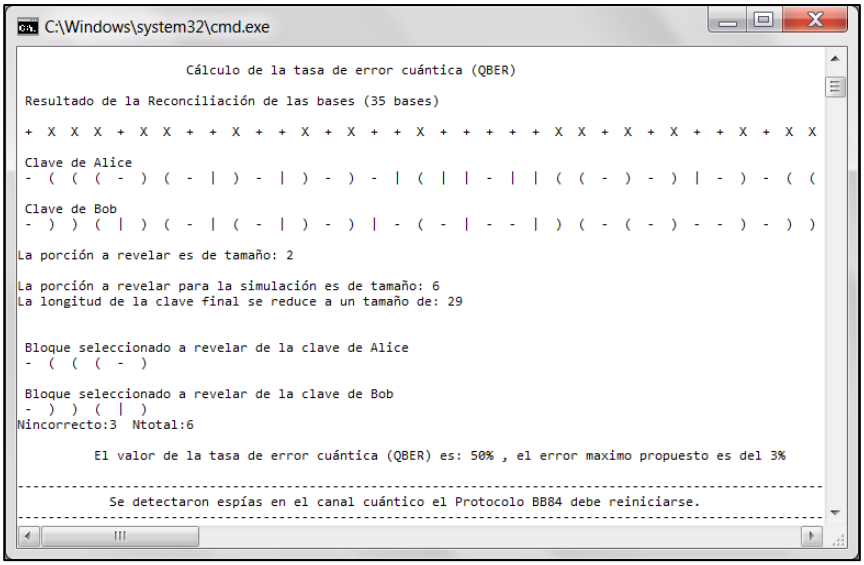


Figura 5. 12: Tamaño del bloque revelado para el cálculo de la tasa de error cuántico (QBER), ejecución con 64 qubits de clave inicial, implementación en C#.

5.2.3.1 Resultados de la Ejecución del Protocolo BB84 espías en C#, estimación del Error

Anteriormente se hizo un sondeo de una simulación para cuatro longitudes iniciales propuestas de la clave (subcapítulo 5.1.1). En estos resultados se hace el sondeo para ver cómo afecta el tamaño de la clave final sin espías y cuando se han perdido los *qubits* a causa de revelar una porción de la misma con espías en la estimación de QBER. La Tabla 5.11 muestra los resultados de una simulación.

	Sin espías	**Con espías, calculando QBER**	
Envío inicial de *qubits.* Tamaño inicial de la clave	Longitud de la clave final	Longitud de la clave final	Valor de QBER
16 *qubits*	4	*	*
128 *qubits*	58	46	41.67% ~
256 *qubits*	116	89	29.63% ~
512 *qubtis*	258	198	21.67% ~
4096 *qubits*	2034	1548	21.40% ~

* No se estimó QBER, porción a revelar de la clave es cero, clave corta.

~ se reinicia el protocolo BB84

Tabla 5. 11: Resultado de las longitudes de las claves finales calculando QBER, de una simulación con el protocolo BB84 con espías implementado con C#.

En la Tabla 5.11 puede notarse que la estimación de QBER reduce el tamaño de la clave final, porque necesariamente se tuvo que tomar una porción de la clave a causa de detectar la presencia del espía Eve alterando *qubits* por sus mediciones. Ahora si la porción a revelar es cero, se reconoce que la clave final es de un tamaño corto para el cálculo de QBER, aunque la porción fue cero, la implementación del protocolo BB84 indica como esté debe reiniciarse.

El tamaño mínimo establecido por la NIST para la clave en algoritmos simétricos es de 112 *bits*, al probar con claves de 512 *qubits* se puede establecer que se está dentro de esta recomendación. La Tabla 5.12 muestra los resultados de diez simulaciones probando con claves iniciales de 512 *qubits*.

No. Simulación	**Longitud de la clave final sin espías**	**Longitud de la clave final después de calcular QBER**	**Valor de QBER**	**Protocolo bb84 se reinicia**
1	252	192	20.00%	•
2	267	204	19.05%	•
3	270	207	17.46%	•
4	247	190	33.33%	•
5	246	189	38.60%	•
6	261	201	30.00%	•
7	251	191	20.00%	•
8	249	192	26.32%	•
9	254	194	15.00%	•
10	251	191	18.33%	•
Promedio:	254.8	195.1	23.81%	

Tabla 5. 12: Resultado de la comparación de las longitudes de las claves finales sin espías y calculando QBER con una clave inicial de 512 qubits, de una simulación con el protocolo BB84 con espías implementado con C#.

La Tabla 5.12 muestra el promedio de clave final sin presencia de espías, en promedio se tiene una clave final de 254.8 *qubits*, con la presencia de espías se sacrifica una porción de la clave para obtener el valor de QBER. En promedio con

la presencia de espías la clave final es de 195.1 *qubits*, pero esta clave es desechada, debido al valor que el valor de QBER es mayor que el valor máximo establecido, es decir, cuando $p_{QBER} > 3\%$ (subcapítulo 4.9).

¿Qué sucede cuándo el cálculo de QBER es igual a 0% ?, Cuando el valor de QBER es igual a 0% significa que en las porciones reveladas por Alice y Bob ($r_{totales}$) al buscar *qubits* diferentes estos no fueron encontrados. Esto indica, que al no encontrar *qubits* diferentes no se encontraron espías en el canal cuántico y la clave final es utilizada para cifrar. La Figura 5.13 muestra el resultado de obtener QBER con valor 0% para una clave inicial de 64 *qubits*.

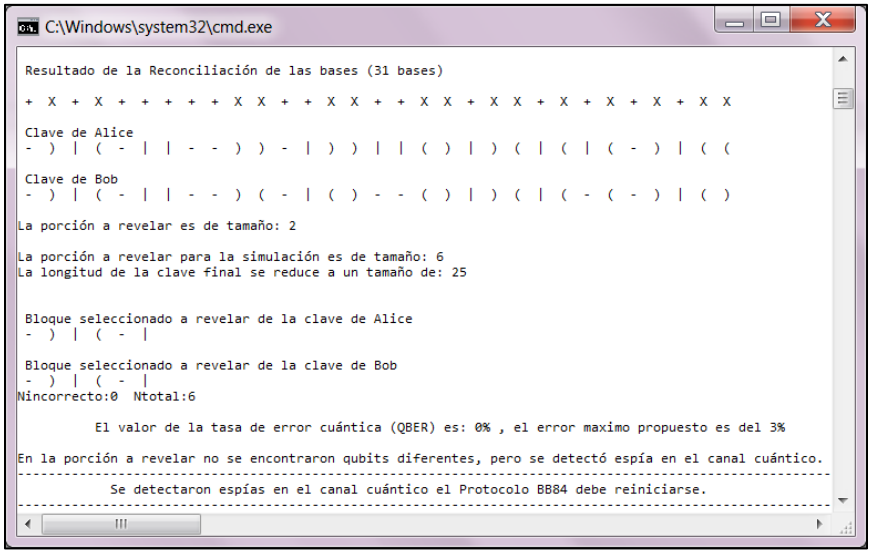


Figura 5. 13: La tasa de error cuántico (QBER) con valor 0 %, una simulación con 64 qubits, implementación en C#.

En la Figura 5.13 se observa que los *qubits* de los bloques seleccionados a revelar de la clave de Alice y de la Clave de Bob, al compararlos son iguales. Si son iguales implicaría que las claves de Alice y de Bob son iguales, pero no lo son. Entonces lo que está sucediendo es, que en los bloques seleccionados toda esa porción de *qubits* fueron iguales. Pero si se hubiera seleccionado otra porción de bloque se encontrarán diferencias. De todo esto las pruebas realizadas analizando los valores del cálculo de QBER la tasa de error cuántico se encontró que, puede diferir al

indicar ausencia de espías en el canal cuántico de comunicación si la porción a revelar no indica presencia de espías, es decir, que la selección de la porción a revelar puede no contener *qubits* diferentes. Aunque el valor de QBER es 0%, la implementación del protocolo BB84 indica como esté debe reiniciarse, en el código se lleva una variable que registra el conteo de fotones que fueron alterados al medir Eve el espía el envío de Alice.

La Tabla 5.13 muestra el resultado de cinco simulaciones, se observa como el valor de QBER es del 0% para claves iniciales respectivamente cortas, conociendo que en promedio su longitud se reduce a un 50%. En tal caso donde el resultado indica que la porción es cero, resultó de obtener una clave final demasiado corta, donde no se sugiere tomar un bloque para calcular el valor de QBER porque el bloque es de tamaño cero *qubtis*. A medida que la clave inicial aumenta de tamaño la posibilidad de que el valor de QBER sea 0% se va reduciendo. Lo que significa que la selección de la porción de la clave final indica la presencia de espías.

	Valor de QBER				
No. Simulación	**32 *qubits***	**64 *qubits***	**128 *qubits***	**512 *qubits***	**4096 *qubits***
1	0.00%	16.67%	41.67%	36.67%	23.11%
2	0.00%	33.33%	26.67%	23.33%	23.93%
3	porción 0	33.33%	16.67%	23.33%	24.54%
4	0.00%	16.67%	38.89%	24.07%	26.67%
5	0.00%	0.00%	33.33%	20%	23.05%

Tabla 5. 13: Valor de la tasa de error cuántico (QBER) para diferentes claves iniciales, implementación en C#.

Capítulo 6

Resultados Generales

Los objetivos de este trabajo se cumplieron totalmente, existió la necesidad de implementar por medio de un lenguaje de programación (C#) el protocolo de criptografía cuántica, el protocolo BB84.

La implementación realizada en este trabajo es una herramienta que permite conocer el comportamiento de la distribución de una clave siguiendo el protocolo BB84. Vista la implementación como una herramienta, con esta se pueden conocer previamente aspectos como la cantidad de *qubits* que fueron perdidos en el proceso o el caso en que el protocolo debe reiniciarse o la cantidad de *qubits* que Eve intencionalmente ha modificado de la clave inicial.

Propiamente las claves generadas en esta implementación también pueden utilizarse para los fines que convengan al lector interesado, aunque en la actualidad es de conocimiento que existen programas generadores de claves.

Ahora bien en el procedimiento de detectar espías en el canal cuántico como se vio en los resultados del subcapítulo 5.2.3.1, las porciones seleccionadas en este trabajo se hicieron con los primeros *qubits*. Pero también se probó seleccionando porciones en diferentes partes de la clave final, observando también que en la selección de estas porciones (algunas) no se encontraron *qubits* diferentes, es decir el valor de $n_{incorrectos}$ es igual a cero.

Como logro obtenido en este trabajo en base a los resultados, se encontró que para el protocolo BB84 existe un detalle en el proceso de calcular QBER que es fundamental en la detección de espías. Se especifica la forma en la que se inicia la detección de espías en base a la selección de las porciones a revelar y algunas porciones reveladas contenían los mismos *qubits,* eran iguales. Entonces se

concluía que no había espías en el canal, pero en la implementación en este trabajo se lleva el registro de que hubo existencia del espía. Si estas porciones son iguales y hubo detección de espías en el canal, entonces ¿Cuál es la acción que el protocolo debe seguir? Numerosas bibliografías y trabajos fueron consultados (bibliografías que se registran en este trabajo), con el objetivo de conocer que sucede cuando las porciones son iguales y hubo espías en el canal cuántico, en dichas consultas no se especifica este resultado. Es entones que como continuación de este trabajo de tesis de forma futura se abre el estudio de concluir o analizar lo que debe proseguir en el protocolo BB84 de acuerdo a los resultados.

Algunas posibles propuestas son: Seguir implementado esta herramienta pero en el análisis de pérdida de *qubits* por distancia o simplemente ver el comportamiento y la obtención de la clave final en la distribución por distancia.

Conclusiones

El presente trabajo desarrolló una implementación por medio de código para simular el protocolo BB84, un protocolo para la distribución cuántica de claves. El lenguaje de programación utilizado fue C# y C por considerarse más apropiados para este fin. Durante el desarrollo de la implementación del protocolo se conocieron varios resultados de su ejecución.

Ahora ¿qué es entonces la criptografía cuántica?, por lo anterior expuesto en este trabajo podemos atrevernos a decir que es una sinergia entre la física cuántica y la teoría de la información. ¿Qué ventajas ofrece esta sinergia? A diferencia de la criptografía convencional que basa su seguridad en la capacidad del cómputo y recordando que una computadora cuántica el problema de factorizar no es exponencial, la criptografía cuántica basa su seguridad en la leyes de la física cuántica y no depende de los progresos matemáticos. En concreto con el principio de incertidumbre de Heinserberg se tiene una ventaja y es la detección de un intruso en la comunicación de la clave. En la criptografía convencional un intruso pasivamente puede registrar todas las conversaciones para después descifrarlas.

Como todo, en términos de seguridad nada está dicho, todo depende de la demostración de las hipótesis. Cuando se afirma que la criptografía cuántica se ha logrado romper, en realidad lo que se ha roto es la implementación y esta ha de mejorarse o arreglarse. En el mejor de los casos aportará información para su análisis.

Una desventaja de la criptografía cuántica es que es un sistema simétrico que no tiene el proceso de firma digital o la autenticidad.

Sobre este trabajo de tesis, el lector interesado puede encontrar en él las bases para iniciar su estudio y comprender con mayor grado sus resultados.

Bibliografía

Referencias Bibliográficas

[1] ***Seguridad Informática***, Roa buen día José Fabián. Interamericana de España, S.L.: McGraw-Hill. 2013.

[2] ***Quantum Computing Explained***, McMahon David. Hoboken, New Jersey: John Wiley & Sons. 2008.

[3] ***Quantum cryptography: State-of-art, challenges and future perspectives***, Teja V. , Banerjee P. , Sharma N.N. y Mittal R.K., Nanotechnology, 2007. IEEE-NANO 2007. 7th IEEE Conference on,
pp. 1296-1301. IEEE Conference Publications.

[4] ***Protocols for Quantum Cryptography***, Kurochkin V.L., Micro/Nanotechnologies and Electron Devices (EDM), 2011 International Conference and Seminar of Young Specialists on, pp. 114-115. IEEE Conference Publications.

[7] ***The Story of Secret Writing***, Kahn David, New York: Macmillan Publishing Co. 1967.

[10] ***Applied Cryptography: Protocols, Algorithms and Source Code in C***, 2ªEd. Schneier Bruce, EE.UU: Jhon Wiley & Sons, Inc. 1996.

[12] ***Introduction to Cryptography: Principles and Applications***, Delfs Hans y Knebl Helmut, Alemania: Springer. 2002.

[13] ***Handbook of Applied Cryptography***, Menezes Alfred J., van Oorschot Paul C. y Vanstone Scott A., EE.UU: CRC Press. 1996.

[15] ***Cryptography and Data Security***, Denning Dorothy E., EE.UU: ADDISON-WESLEY. 1983.

[17] ***Quantum Computation and Quantum Information***, Nielsen Michael A. y Chuang Isaac L., United Kingdom: Cambridge University Press. 2010.

[21] ***Applied Quantum Cryptography***, Lect. Notes Phys. 797, Kollmitzer C. y Pivk M., Berlin Heidelberg: Springer-Verlag. 2010.

[24] ***Quantum Computing: From Linear Algebra to Physical Realizations*****,** Nakahara Mikio y Ohmi Tetsuo, EE.UU.: CRC Press. 2008.

[31] ***Introducción a la Mecánica Cuántica***, 2ªEd. México: FCE, UNAM 2006, Luis de la Peña, México: Fondo de Cultura Económica, 2012 Segunda reimpresión.

[6] ***Communication Theory of Secrecy Systems***, Shannon Claude E., The Bell System Technical Journal, Volumen: 28, No. 4, Año de publicación: 1949, pp. 656-715. IEEE. Acatel-Lucent Journal.

[11] ***New Directions in Cryptography***, Diffie Whitfield y Hellman Martin E., Information Theory, IEEE Transactions on, Volumen: 22, Año de publicación: 1976, pp. 644-654. IEEE Journals & Magazines.

[14] ***A Mathematical Theory of Communication***, Shannon Claude E., The Bell System Technical Journal, Volumen: 27, Año de publicación: 1948, pp. 623-656. IEEE. Acatel-Lucent Journal.

[18] ***On computable numbers, with and application to the Entscheidungsproblem***, Turing Alan M., Oxford Journals, Proceedings of the London Mathematical Society, 2 s. Volumen: 42, Año de publicación: 1936-1937, pp. 230-265.

[29] ***Limitations on Practical Quantum Cryptography***, Brassard Gilles, Lütkenhaus Norbert, Mor Tal, y Sanders Barry C., Physical Review Letters scientific Journal, Volumen: 85, No. 6, Año de publicación: 2000, pp. 1330-1333.

[32] ***Quantum cryptography: public key distribution and coin tossing***, Bennett C. H., Brassard G., IEEE International Conference on Computers, System and Signal Processing. IEEE Press 1984, pp. 175-179.

[33] ***Quantum cryptography based on Bell's theorem***, Ekert Artur K., Physical Review Letters scientific Journal, Volumen: 67, No. 661, Año de publicación: 1991.

[34] ***Quantum cryptography using any two nonorthogonal states***, Bennett C. H., Physical Review Letters scientific Journal, Volumen: 68, No. 3121, Año de publicación: 1992.

[35] ***Optimal Eavesdropping in Quantum Cryptography with Six States***, Bruß D. Physical Review Letters scientific Journal, Volumen: 81, No. 3018, Año de publicación: 1998.

[36] ***Incoherent and coherent eavesdropping in the six-state protocol of quantum cryptography***, Bechmann-Pasquinucci H. y Gisin N. Physical Review A scientific Journal, Volumen: 59, No. 4238, Año de publicación: 1999.

[37] ***Quantum Cryptography Protocols Robust against Photon Number Splitting Attacks for Weak Laser Pulse Implementations***, Scarani Valerio, Acín Antonio, Ribordy Grégoire, y Gisin Nicolas, Physical Review Letters scientific Journal, Volumen: 92, No. 057901, Año de publicación: 2004.

[38] ***Secret-Key Reconciliation by Public Discussion***, Brassard G. and Salvail L., Lecture Notes in Computer Science (LNCS) - Springer, Volumen: 765, pp.410-423, Año de publicación: 1994.

Referencias Web de interés

[8] ***Fundamentos de criptografía***, Universidad Nacional Autónoma de México, Facultad de Ingeniería.
URL: http://redyseguridad.fip.unam.mx/proyectos/criptografia/criptografia/

[9] ***National Institute of Standards and Technology (NIST),*** Sitio Web de la agencia federal de tecnología que trabaja con la industria para desarrollar y aplicar tecnología, medidas y normas en los Estados Unidos.
URL: http://csrc.nist.gov/publications/PubsFIPS.html

[20] ***Fundamentos de computación cuántica para su aplicación en teoría de la información cuántica y criptografía cuántica***, Hetch Juan Pedro, Universidad de Buenos Aires, Argentina 2005.
Libro electrónico, Base de datos: Recursos Libres.
URL: http://bibliotecas.unam.mx/

[22] ***Cuentos cuánticos***. Sitio Web de ciencias y física.
URL: http://cuentos-cuanticos.com/2013/05/16/el-efecto-tunel/

[25] ***Basic concepts in quantum computation***, Artur Ekert, Patrick Hayden y Hitoshi Inamori, University of Oxford, United Kingdom. Año de publicación 2008, pp.1-37.
URL: http://arxiv.org/pdf/quant-ph/0011013v1.pdf

[26] ***¿Qué es el gato de Schrödinger?***, Sitio Web de ciencias (Muy interesante). URL: http://www.muyinteresante.es/ciencia/preguntas-respuestas/que-es-el-gato-de-schroedinger-181376290562

[28] **libquantum**, Librería de C para computación cuántica y simulación cuántica.
Autores de libquantum: Björn Butscher y Hendrik Weimer.
URL: http://www.libquantum.de/

[30] ***Cryptographic Key Length Recommendation***, Sitio Web provisto por BlueCrypt, que reúne las recomendaciones y fórmulas matemáticas para los requisitos de tamaño mínimo de clave para la seguridad.
Autores: Damien Giry.
URL: http://www.keylength.com/en/4/

[5] ***La criptografía clásica***, Fernández Santiago, Revista Matemática (matematika aldizkari) No. 24, Año de publicación: 2004, pp. 119-142.
URL: http://dialnet.unirioja.es/ejemplar/95554

[16] ***Computación cuántica***, Revista de Divulgación de la Ciencia de la UNAM (¿cómoves?,), No. 67, Año de publicación: 2004, pp. 10-14.
URL: http://www.comoves.unam.mx/assets/revista/67/computacioncuantica.pdf

[19] ***Computación y comunicación cuánticas***, Revista digital de divulgación matemática (matematicalia), Volumen: 3, Nos. 4-5, Año de publicación: 2007, Sección: Comunicación.
URL:
http://www.matematicalia.net/index.php?option=com_content&task=view&id=430&Itemid=254

[23] ***Información cuántica: ideas y perspectivas***, Klimov Andrei B., Revista Cinvestav, Volumen: 27, No. 1, Año de publicación: 2008, pp. 12-17.
URL: http://www.fis.cinvestav.mx/~orosas/REVCINV/p12.pdf

[27] ***Distribución de claves cuánticas***, Bradler Kamil, Revista Cinvestav, Volumen: 27, No. 1, Año de publicación: 2008, pp. 24-29.
URL: http://www.fis.cinvestav.mx/~orosas/REVCINV/p24.pdf

Acrónimos

AES Advanced Encryption Standard, algoritmo de criptografía simétrica.

ANSI American National Standards Institute, organización sin fines de lucro que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos.

ASCII American Standard Code for Information Interchange, código para caracteres del alfabeto latino y estaunidense.

BB84 Primer protocolo de criptografía cuántica, de la siglas de sus autores Charles H. Bennett y Gilles Brassard, y creado en 1984.

BIT Binary digit, es una unidad de información puede tomar dos valores: cero o uno en la computación digital.

BQP Bounded error Quantum Polynomial time, representa la clase de algoritmos que pueden ser resueltos en una computadora cuántica.

DES Data Encryption Standard, algoritmo de criptografía simétrica.

DSA Digital Signature Algorithm, algoritmo para firmar información.

EFF Electronic Frontier Foundation, organización de Estados Unidos sin fines de lucro, su objetivo son los derechos digitales.

FIPS PUB Federal Information Processing Standards Publication, estándares desarrollados por Estados Unidos en seguridad computacional y comunicación.

GSM Global System for Mobile communication, ejemplo de la segunda generación de un sistema telefónico celular.

IACR Internal Association for Cryptologic Research, organización científica internacional sin fines de lucro cuyo objetivo es la investigación adicional en criptología y campos relacionados.

IDEA International Data Encryption Algorithm, algoritmo de criptografía simétrica.

IZArc Programa libre de compresión, descompresión y cifrador de archivos, desarrollado por el programador búlgaro Ivan Zahariev.

LibQuantum Library Quantum**,** librería de C para computación cuántica y simulación cuántica.

Log Símbolo del logaritmo, exponente de una potencia al que se eleva la base (a) para encontrar el número $\log(a)$.

MIT Massachusetts Institute of Technology, universidad privada de Estados Unidos, la mission del MIT es educar estudiantes en la ciencia, tecnología y otras areas, así como el avance en el conocimiento.

NIST The National Institute of Standards and Technology, agencia federal de tecnología que trabaja con la industria para desarrollar y aplicar tecnología, medidas y normas en los Estados Unidos.

PNS *Photon Number Splitting*, ataque basado en la imprecisión de las fuentes de emisión de fotones individuales.

QBER Quantum Bit Error Rate, cálculo para estimar la tasa de error cuántico.

QKD Quantum Key Distribution, protocolo de criptografía cuántica.

QUBIT Quantum bit, unidad básica de información de la computación cuántica.

RC5 Rivest Cipher, cifrado de bloque de clave simétrica del autor Ronald Linn Rivest.

RSA De las siglas de sus autores R.L. Rivest, A. Shamir y L. Adleman. Algoritmo criptografía asimétrica.

XOR Exclusive Or, es una función que representada como una puerta en la electrónica digital da una salida igual a 0 cuando sus entradas son iguales y 1 cuando son diferentes.