Método criptográfico para cifrar información usando los estados cuánticos de polarización de fotones individuales

Johan Javier Urrego Urrego

Institución Universitaria ITM

Facultad de Ingeniería

Medellín, Colombia

2019

Método criptográfico para cifrar información usando los estados cuánticos de polarización de fotones individuales

Johan Javier Urrego Urrego

Trabajo de grado presentada como requisito para optar al título de:

Maestría en seguridad informática

Director:(Ph.D., Doctor, en Física.) Boris Anghelo Rodríguez Rey

Codirector:(Ph.D., Doctor, en Física.) Camilo Valencia Balvín

Línea de Investigación:

Criptografía y Tratamiento de Información Cuántica

Grupo de Física Atómica y Molecular, GFAM

Grupo de Investigación en Física Teórica, Aplicada y Didáctica. GRITAD

Institución Universitaria ITM

Facultad de Ingeniería

Medellín, Colombia

2019

“Si el espacio y el tiempo son infinitos, nuestra existencia también lo es, por ser parte malformada de la materia.”

Agradecimientos

El presente trabajo de investigación fue realizado bajo la supervisión del profesor Boris Anghelo Rodríguez Rey, del Instituto de Física de la Universidad de Antioquia, a quien expresar mi más profundo agradecimiento, por hacer posible la realización de este estudio. Además, de agradecer su paciencia, tiempo y dedicación. Doy gracias por los conocimientos aportados, por su apoyo. Sin su ayuda este trabajo no hubiese sido posible. Gracias profesor.

Agradezco a todas las personas que en forma directa o indirecta aportaron con sus conocimientos para que este trabajo de investigación se llevara a cabo. Por último un agradecimiento profundo a mis padres por su constante apoyo, motivación y paciencia que siempre manifestaron.

Resumen

La teoría de información cuántica ha mejorado los sistemas de comunicación en todos los niveles, como en el almacenamiento, procesamiento, transmisión, seguridad, y capacidad de respuesta, pero es la criptografía cuántica la que tiene un crecimiento más rápido en cuanto a resultados de investigación. La criptografía cuántica es un modelo físicos para la generación y distribución de claves criptográficas, que aprovecha las propiedades de la mecánica cuántica como la superposición de estados, la polarización de fotones individuales, el entrelazamiento, la teleportación, y la codificación densa, propiedades que proporcionan ventajas computacionales para transmitir, procesar y codificar información a más alta velocidad y con mejor seguridad. Además se pueden desarrollar aplicaciones para factorizar números en tiempo polinómico y construir sistemas de seguridad basados en criptografía cuántica con el fin de mantener la confidencialidad, integridad y disponibilidad de la información.

Bajo este contexto, en este trabajo se describen los conceptos teóricos de la mecánica cuántica que permitieron simular un método criptográfico que utiliza la generación y distribución de claves cuánticas del protocolo BB84, como un sistema de intercambio de claves seguras entre un transmisor Alice y un receptor Bob. Estos sistemas utilizan las propiedades las propiedades de la mecánica cuántica para distribuir la clave y no problemas matemáticos difíciles de resolver como los que utiliza la criptografía construida en una base de estados clásicos. Para este método criptográfico, la generación y distribución de la clave cuántica se desarrolla en dos escenarios, en el primero no se tendrá ninguna entidad que trate de interceptar la comunicación entre el transmisor y el receptor, en el segundo escenario se tendrá un espía Eve en el canal cuántico de comunicación que tratará de obtener la clave que se está trasmitiendo entre Alice y Bob.

La generación y distribución de la clave cuántica utilizando el protocolo BB84 con y sin espía, se mostrará en un ambiente de programación y simulación desarrollado con el lenguaje cuántico Qiskit implementado por IBM para sus computadoras cuánticas, este lenguaje se encuentra descrito en el desarrollo del método criptográfico y en el Apéndice A de este trabajo. Se explican paso a paso el funcionamiento del protocolo BB84 y como a partir de la tasa de error generada por el intruso en el canal de comunicación, el transmisor y el receptor detectan una intrusión en el sistema. Además en la simulación del modelo se muestra como al integrar claves cuánticas con cifradores como el de Vernam, se puede obtener una seguridad perfecta en la codificación y decodificación de la información.

Palabras Claves: Criptografía, Teoría de la información y computación cuántica, Protocolos y algoritmos cuánticos.

Abstract

Quantum information theory has improved communication systems at all levels, such as storage, processing, transmission, security, and responsiveness, but it is the quantum cryptography that has the fastest growth in terms of research results. Quantum cryptography is a physical model for the generation and distribution of cryptographic keys, which takes advantage of the properties of quantum mechanics such as overlapping states, polarization of individual photons, entanglement, teleportation, and dense coding, properties that provide computational advantages to transmit, process and encode information at a higher speed and with better security. In addition applications can be developed for factoring numbers in polynomial time and build security systems based on quantum cryptography in order to maintain the confidentiality, integrity and availability of information.

In this context, this paper presents the theoretical concepts of quantum mechanics that allowed simulating a cryptographic method that uses the generation and distribution of quantum keys of the BB84 protocol, as a system secure keys exchange between a transmitter, Alice, and a receiver, Bob, who use the properties of quantum mechanics to distribute the key and not hard-to-solve mathematical problems like those used by cryptography built on a base of classical states. For this cryptographic method, the generation and distribution of the quantum key will be developed in two scenarios, in the first one there will be no entity that tries to intestate the communication between the transmitter and the receiver, in the second scenario there will be an Eve spy in the quantum communication channel that will try to obtain the key that is being transmitted between Alice and Bob.

The generation and distribution of the quantum key using the BB84 protocol with and without a spy will be displayed in a programming and simulation environment developed with the quantum language Qiskit implemented by IBM for its quantum computers, this language is described in the development of the cryptographic method and in Appendix A of this work. The operation of the BB84 protocol is explained step by step and, as from the error rate generated by the intruder in the communication channel, the transmitter and the receiver detect an intrusion into the system. In addition, the simulation of the model shows how to integrate quantum keys with ciphers such as Vernam, and you can obtain perfect security in the encoding and decoding of information.

Keywords: Cryptography, Quantum computation and informatio theory, Quantum protocols and algorithms

Prefacio

La criptografía es parte fundamental en el desarrollo de los procesos de seguridad de la información, y a medida que esta evoluciona se han implementado nuevas modelos que permiten mejorar los sistemas, a nivel de gestión, capacidad de respuestas y seguridad de los mensajes transmitidos. Pero con el uso masivo de las comunicaciones digitales y los avances científicos en computación cuántica, los problemas de seguridad de la información se ha incrementado notablemente, hasta el punto de tenerse que desarrollar tecnología especializada como la criptografía cuántica para mantener la seguridad de los datos.

En los últimos años, la criptografía cuántica ha tomado una gran importancia en la protección de información, ya que los sistemas criptográficos basados en la transmisión de claves cuánticas tienen un nivel de seguridad superior que cualquiera de los método combinacionales y en general estos algoritmos proporcionan las herramientas para controlar los riesgos informáticos generados por ataques criptográficos.

Sobre la Criptografía Clásica y Cuántica

A partir de la idea de que el desarrollo de la criptografía, como el desarrollo de todas las ideas, ha sido un proceso evolutivo ligado a las avances tecnológicos, científicos y sociales de la humanidad, en el siguiente trabajo de investigación se propone una nueva clasificación de la criptografía, diferente a la usada en la literatura, que estará determinada por su estructura teórica y sus posibilidades de realización física. En ese sentido, clasificaremos la criptografía en dos grandes grupos: la criptografía clásica y la criptografía cuántica. En la criptografía clásica se agrupan todos los algoritmos de cifrado construidos en un espacio de estados clásicos que pueden realizarse a través de sistemas físicos clásicos, es decir, que en este grupo se encuentran todos los sistemas criptográficos antiguos y los algoritmos de criptografía simétrica, asimétrica y las funciones hash que se conocen comúnmente como criptografía matemática o moderna. La criptografía cuántica se fundamenta en el espacio de estados cuántico y cuya única realización física es un sistema cuántico *per se*, donde se encuentran agrupados todos los algoritmos de trasmisión de claves cuánticas. Esta clasificación propuesta puede parecer extraña a la luz de los desarrollos usuales en criptografía, pero es completamente natural a la división que realiza la física en sistemas clásicos y cuánticos, división que es la más general que puede ofrecer la Física en la actualidad.

Estructura del Trabajo de investigación

Con esta nueva clasificación de la criptografía, el siguiente trabajo se divide en tres partes: en la primera parte se presenta de forma general los principales algoritmos de criptografía clásica que se dividió en lo que se conoce como la la criptografía antigua y la criptografía moderna, agrupación de técnicas criptográficas que utilizan sustitución, transposición, y problemas matemáticos difíciles de resolver para generar claves de cifrado. Adicionalmente se muestran algunas métodos teóricos y computacionales que pueden vulnerar los modelos de criptografía actuales (Capítulo 2). En la segunda parte se describen las propiedades de la mecánica cuántica, teoría de la información y computación cuántica con las cuales se puede vulnerar la criptografía moderna, además se realiza una descripción de los principales protocolos de distribución de claves cuánticas (Capítulos 3 y 4). Finalmente en la última parte de esta investigación se muestra como la transmisión de claves cuánticas puede fortalecer la seguridad de los criptosistemas clásicos. Para la prueba de este método se utilizará el lenguaje de programación QisKit [1], desarrollado por IBM para sus computadores cuánticos (Capítulo 5).

A lo largo del trabajo se usará la siguiente clasificación paras las figuras: i) Figura modificada, si esta ha sido rediseñada de una referencia previa. En este caso la figura cuenta con la respectiva referencia al final de la descripción. ii) Figura tomada de, si la figura se ha tomado completamente de la literatura existen. En este caso también cuenta con la referencia al final de la descripción. iii) Figura propia, en este caso la figura fue construida por al autor de la tesis y no tiene ninguna referencia o comentario que indique lo contrario.

Objetivos de Investigación

El desarrollo de este trabajo de investigación se centra principalmente en dar respuesta a los siguientes objetivos:

Objetivo General

- Proponer un método sobre criptografía cuántica para cifrar información usando los estados de polarización de fotones individuales

Objetivos Específicos

- Caracterizar las propiedades de los estados de polarización en partículas elementales como los fotones, para construir un método de cifrado de información de modo cuántico.
- Implementar un ambiente de prueba (simulación) que haciendo uso del método de criptografía cuántica propuesto, permita realizar su valoración.
- Evaluar el método propuesto mediante simulación computacional.

Con estos objetivos se construyo un marco metodológico que permite entender la criptografía a partir del paradigma de la teoría cuántica de la información.

Índice general

Índice de figuras

Índice de cuadros

CAPÍTULO 1

INTRODUCCIÓN

Se puede decir que la criptografía se originó con la misma escritura y es tan antigua como ella, pero con los desarrollos tecnológicos de las ultimas décadas, el crecimiento exponencial de los datas y la velocidad de procesamiento en las maquinas, la información se ha hecho vulnerable y puede ser manipulada por sectores organizados que invierten todo su tiempo y esfuerzo en desarrollar herramientas sofisticadas para apropiarse de sistemas clasificados [2], riesgo demasiado grande, pues la información es el activos más importante en la sociedad moderna, tener el control de la información implica directamente tomar buenas decisiones y resolver problemas que pueden alterar el curso de la historia, con esta se crear estrategias que soportan los gobiernos, la economía y general cualquier acción cotidiana que asegure la existencia humana, es decir, que garantizar la seguridad de la información es fundamental para preservar y expandir las especies en este universo.

En este sentido, la criptografía se a convertido en el pilar de la seguridad de la información durante toda la historia, en la modernidad esta se a encargado de la transmisión y almacenamiento de datos de tal manera que no puedan ser comprendidos ni modificados por terceros, proceso en el cual se realiza un intercambio de una o varias claves que permite codificar mensajes con seguridad absoluta siempre y cuando se realice un intercambio de claves de manera presencial, echo que no es posible con criptografía clásica por que existen múltiples sistemas que necesitan intercambiar información confidencial (bancos, tiendas, personas en Internet, etc). Pero este proceso si se puede garantizar cuando de utiliza criptografía cuántica, que es una rama de teoría de la información cuántica en pleno crecimiento, se proyecta como la única teoría capaz de garantizar la seguridad en la generación y transmisión de clave criptográficas y convertir Internet en un lugar completamente seguro, esta criptografía ha adquirido fuerza en los últimos años, se están estudiando las propiedades y las correlaciones de los sistemas físicos, la superposición, el entrelazamiento, el teorema de Bell y el teorema de no clonación, características de los sistemas cuánticos que se están aprovechar para construir criptosistemas seguros capases de mantener la confidencialidad e integridad de la información [3, 4].

Esta nueva criptografía se puede entender como la implementación física de la información, pasar del bit 0 o 1 desarrollado por Shannon en la teoría matemática de la información [5], al qubit que se puede representar con el $|0\rangle$, el $|1\rangle$ o una superposición de estados cuánticos $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, principio fundamental de la mecánica cuántica que afirma que cualquier sistema físico en un estado cuántico dado $|\psi\rangle$ se puede escribir como una combinación lineal de estados de una base $\{|\alpha_i\rangle\}$ en la forma $|\psi\rangle = a_1\,|\alpha_1\rangle + \ldots + a_n\,|\alpha_n\rangle$, donde la probabilidad de encontrar el sistema que se ha preparado en el estado $|\psi\rangle$ en alguno de los estados cuánticos $|\alpha_i\rangle$ es $|a_i|^2$ y por supuesto se cumple la condición $\sum_{i=1}^{n} |\alpha_i|^2 = 1$ [6, 7]. Con esta nueva idea, se abre la posibilidad de superar las limitaciones del procesamiento de información clásico, se puede pensar en superar la barrera de las leyes físicas

clásicas, donde los dispositivos electrónicos tiene un limite de miniaturización, y cuando estos alcanzan una escala de nanómetros aparecen los efectos cuánticos, la transmisión de información clásica se pierde en los microcircuitos generando pérdidas de información. Este fenómeno en computación clásica es conocido como efecto túnel y no permite la miniaturización de los dispositivos electrónicos infinitamente [8].

En este sentido se puede decir, que la criptografía cuántica surge como respuesta de seguridad informática a los desafíos tecnológicos impuestos por la computación cuántica, que se empieza desarrollar con la crítica hecha a la mecánica cuántica en el artículo EPR, publicado en 1935 por Albert Einstein, Boris Podolsky, y Nathan Rosen, quienes argumentaban que la mecánica cuántica no describe completamente la realidad física. La tesis principal del artículo EPR intentaba mostrar que la mecánica cuántica era una teoría incompleta, que no describe la realidad fielmente, y por lo tanto, se tenía que completar con elementos abstractos denominados variables ocultas que se acomodan al formalismo de la teoría para poder construir predicciones deterministas no probabilísticas, pues Einstein pensaba que las probabilidades cuánticas tenían un origen subjetivo, hecho que no le permitió aceptar la aparente acción fantasmal a distancia [9].

En un primer análisis a la incompletitud de la teoría cuántica propuesta en el artículo EPR, aparece la propiedad de entrelazamiento, acuñada por Erwin Schrödinger, quién la destaca como el rasgo cuántico característico[10, 11]. Pero es hasta 1964 con la demostración de las llamadas desigualdades de Bell [12], desarrolladas por el físico irlandés John Stewart Bell, y su posterior verificación experimental que se puede entender el fenómeno de entrelazamiento entre sistemas cuánticos sin realizar suposiciones ad-hoc. Se comprende que dos sistemas pueden tener correlaciones extremadamente complejas, hasta tal punto que violan desigualdades exigidas por las correlaciones clásicas. Por ejemplo, cuando se modifica el estado de uno de los dos sistemas, el otro puede interpretar el cambio y modificar su estado, es decir que, si se tienen dos sistemas Alice y Bob, los cuales han tenido una interacción previa en el tiempo, y como consecuencia quedan en un estado entrelazado y se alejan progresivamente el uno del otro, entonces para esta situación la mecánica cuántica puede predecir el estado del sistema Bob conociendo la modificación en los estados del sistema Alice, este fenómeno se presenta en forma instantánea y sin importar la distancia que exista entre los dos sistemas [13].

Las desigualdades de Bell fueron construidas a partir de dos conceptos que aparecen en el artículo EPR: el realismo y la localidad. El realismo, se entiende según EPR como la correspondencia entre los elementos de un sistema físico y los conceptos teóricos, conceptos que pueden predecirse con certeza sin perturbar el sistema. Por ejemplo, para EPR los estados cuánticos del sistema Bob son reales debido a que se puede predecir su estado mediante el sistema Alice. La localidad, en EPR, aparece cuando el resultado de las mediciones en el sistema Bob no dependen del sistema Alice, si no exclusivamente del estado completo del sistema y del polarizador que utiliza Bob para realizar las medidas. Bell impone los conceptos de realismo y localidad presentados en EPR, en un contexto clásico muy general, para deducir sus desigualdades que son formas lógico-matemáticas con las que se valida el comportamiento no clásico. Es decir, si las ideas de EPR son ciertas, entonces las desigualdades de Bell son correctas y en consecuencia la teoría cuántica es incompleta, pero se ha demostrado que las desigualdades de Bell son violadas por la teoría cuántica [14].

Desde los años 80 y gracias a los seminales trabajos de Alain Aspect [13] se han desarrollado múltiples experimentos que han demostrado la violación a las desigualdades de Bell, por lo tanto, la mecánica cuántica da cuenta de la correlación entre los sistemas Alice y Bob mediante el entrelazamiento cuántico [15]. También a partir de las desigualdades de Bell se empiezan a desarrollar

algoritmos cuánticos como el de Peter Shor que ataca el problema de la factorización y el algoritmo de Grover que puede realizar búsquedas en listas desordenadas, algoritmos que reducen los tiempos de cómputo utilizando el entrelazamiento, la interferencia, y el paralelismo cuántico, conceptos que permiten resolver problemas clásicos que eran difíciles. Si a lo anterior se agrega el supuesto de la alta velocidad de procesamiento de los computadores cuánticos, es posible generar ataques a los algoritmos de criptografía clásica: RSA [16], ECC, AES [17], y las funciones hash, sistemas actuales integrados a la seguridad de la información para proteger los datos [18, 19].

El algoritmo de Peter Shor ataca el problema de la factorización, problema matemático difícil de resolver utilizando computación clásica y que es la base teórica para construir RSA, algoritmo clásico de criptografía asimétrica, que genera las claves criptográficas mediante el cálculo de operaciones matemáticas sencillas con números primos pero que son muy difíciles de invertir. Por ejemplo, la multiplicación de dos números primos grandes es un computo básica en una computadora, pero en el caso contrario, si se tiene solo el resultado del computo del par de números y se quiere obtener cada unos de los números primos que originaron este resultado, ya el computo no es fácil, y dependiendo del tamaño en bits de estos números es casi imposible realizar las permutaciones que den los resultado correcto en tiempos cortos (días o meses), es por esto, que un ataque de factorización no es práctico para el criptosistema RSA, ya que un par de claves criptográficas construidas con números primos de 2048 bits de longitud, requiere un tiempo de factorización que oscila entre varios años de operaciones utilizando supercomputadoras clásicas, pero no es suficiente la seguridad de RSA para proteger información. En 1984 el matemático Peter Shor demostró que se pueden factorizar números primos grandes utilizó la potencia de cómputo cuántica, su algoritmo cuántico tiene la capacidad de reducir los tiempos operacionales de computo de exponenciales a polinómicos, echo que permite vulnerar el flujo de información en la red [20].

Este algoritmo tiene la potencia teórica para factorizar un número N en un tiempo polinomial de orden $O((\log(N))^3)$ [21], mientras que el mejor algoritmo clásicos (*general number field sieve*) no pueden factorizar en tiempos menores a $O(\exp\big((64b/9)^{1/2}(\log b)^{2/3}\big))$, donde b es el número de bits [22]. Otro algoritmo cuántico que puede vulnerar parte de la criptografía moderna es el algoritmo de Grover, que tiene la capacidad de realizar búsquedas no ordenadas en una secuencia de tamaño N en tiempo $O(\sqrt{N})$ y almacenamiento de $O(logN)$ [23], lo que significa que el criptosistema AES y a las funciones hash no pueden proteger la infracción de la potencia de computo cuántico. En este sentido la reducción de los tiempos de operación de los algoritmos de Shor y Grover son una amenaza a la seguridad de la información en forma clásica, pero a la vez permite que se desarrolle la nueva criptografía basada en sistemas cuánticos.

La criptografía cuántica se fundamenta en las propiedades de la mecánica cuántica para generar y transmitir las claves de cifrado, en canales cuánticos de fibra óptica o espacio libre. El primer modelo de criptografía cuántica que se construyo para la distribución de claves cuánticas fue el protocolo BB84 desarrollado en 1984 por C.H. Bennett y G. Brassard [24], este sistema cartográfico se caracterizo por tener un transmisor conocido como (Alice) y un receptor (Bob), que utilizan cuatro estados del fotón $(|\updownarrow\rangle, |\leftrightarrow\rangle, |\nearrow\rangle, |\nwarrow\rangle)$ estas dos bases (vertical-horizontal + y oblicua X)para polarizar los fotones y medir los estados de los qubits que periten generar la clave cartográfica para codificar o decodificar la información. La seguridad de la claves se garantiza parcialmente con el teorema de no clonación, dice que si un intruso (Eva) quiere espiar el flujo de información generado por Alice y Bob, éste tiene que escoger una de las dos bases posibles para medir y en consecuencia el estado observado será modificado lo que genera errores en la trasmisión de los qubits y les permite Alice y Bob saber que existe un intruso en el canal de comunicación cuántico.

A partir del protocolo de criptografía cuántico BB84 se han desarrollado barios modelos para la transmisión de claves cuánticas basados en conjuntos de estados no ortogonales, como los protocolos B92, SARG04 y K05, que son modificación de BB84. En el protocolo B92 se utilizan la misma metodología de medición y codificación que en BB84, se tiene las dos bases para codificar y medir, pero ya no se usan los cuatro estados de polarización, se tiene solo los dos estados no ortogonales $(|\nearrow\rangle, |\nwarrow\rangle)$ entre sí para codificar (ver figura 4.8 del capítulo 3). La diferencia del protocolo SARG04 con respecto a BB84 se presenta cuando Bob interpreta los estados enviados por Alice utilizando un canal clásico, en este protocolo Bob mide los estados enviados y los asocia con un resultado opuesto a la medida hecha por Alice (ver subsección 3.2.2). En el protocolo K05 en ves de codificar bits individuales $0, 1$ los estados de polarización, se codifican cadenas binarias aleatorias en cada uno de los estados para generar la clave (ver figura 19 del capítulo 3). también se han desarrollado modelos de transmisión de claves cuánticas que no utilizan la polarización de fotones, es el caso del protocolo E91, que utiliza la propiedad de entrelazamiento de la mecánica cuántica, el teorema de Bell, de tal manera que el proceso de transmisión de la clave cuántica utiliza fotones entrelazados, donde Alice y Bob obtienen un fotón de cada par enviado para formar la clave (ver subsección 3.2.3) [25].

Con los modelos de criptografía cuántica iniciales y el estudio de las propiedades y correlaciones de los sistemas cuánticos, se han desarrollado protocolos de distribución de claves cuánticas, que mejoran la eficiencia y la seguridad de las claves criptográficas, como es el caso de los protocolos COW, DOS-QKD, KBM09 y EQKD, en donde se integra propiedades de la mecánica cuántica como el entrelazamiento, la codificación superdensa, el teorema de Bell y el análisis de las correlaciones y coherencias en canales cuánticos de comunicación ruidosos. Por ejemplo El protocolo COW (Coherent One-Way protocol), opera con una tasa alta de pulsos coherentes, lo que permite reducir la interferencia y ataques por división de fotones [26]. El protocolo DOS-QKD (Differential phase-shift QKD), utiliza un tren de pulsos de estados débiles coherentes al que se le realiza una modulación aleatoria de fase entre $0, \pi$, para prevenir ataques por división de número de fotones [27]. El protocolo KMB09 utiliza para la generación y transmisión de la clave cuántica dos bases mutuamente imparciales para realizar la distribución de claves, en una de sus bases se codifica el cero 0 y en la otra el uno 1, este proceso permite que se tenga una tasa mínima de errores en las transferencias de fotones en canales cuánticos ruidosos [28]. El protocolo EQKD, se fundamenta en el entrelazamiento como estrategia de seguridad en la trasmisión de clave y en la codificación densa para incrementar la eficiencia de codificación. Con estos protocolos se esta construyendo el nuevo paradigma de seguridad de la información, capas de soportar la alta velocidad de procesamiento de la computación cuántica.

CAPÍTULO 2

CRIPTOGRAFÍA EN EL ESPACIO DE ESTADOS CLÁSICO

En este capítulo se mostrarán los principales sistemas de criptografía que se han utilizado históricamente desde la antigüedad hasta la modernidad para codificar la información. Se empezará con la descripción de los sistemas criptográficos por sustitución y transposición de caracteres, que les permitió a las civilizaciones de la antigüedad cifrar sus estrategias de combate para que las tropas enemigas no pudiesen saber el plan de acción. Además se muestra como la criptografía matemática o moderna, transformó la forma de compartir información de modo seguro a partir de tres modelos: la criptografía simétrica, la criptografía asimétrica, y las funciones hash, técnicas responsables de mantener la confidencialidad, integridad, y autenticación de la información en la actualidad. Adicionalmente se presentan algunas ideas que pueden vulnerar la criptografía moderna si son implementadas.

2.1. Criptografía Clásica

La criptografía desde su origen ha proporcionado herramientas para garantizar la privacidad de las comunicaciones. Por ejemplo, la civilización griega hacía el año 400 a.C desarrollo un sistema criptográfico para compartir información, llamado la Escítala Espartana, que permitía codificar los mensajes dirigidos a los ejércitos utilizando un trozo de madera con un diámetro específico y una tira de cuero que se enrollaba en la madera para escribir el mensaje. Si se quería saber el contenido de información era necesario conocer el diámetro del trozo de madera, que correspondía a la clave con la que se descifraría el mensaje [29, 30]. Posteriormente en el año 150 a.C. aparece un un método criptográfico conocido con el nombre de Tablero de Polibio, que utilizaba un tablero de $5 * 5$ espacios, donde se codificaba un mensaje con un algoritmo que remplazaba cada carácter del alfabeto con las coordenadas de su posición en un cuadrado [31]. Más tarde, en el año 100 a.C se desarrollaría uno de los sistemas criptográficos más importantes de la criptografía antigua, el cifrado del César. Este sistema criptográfico permitió realizar sustituciones de caracteres en el alfabeto, es decir, si el mensaje empezaba con la letra A, entonces para cifrar se realizaba un corrimiento en el alfabeto a una letra diferente de la A. Partiendo de esta técnica se construyó toda la criptografía que utilizo transposición y sustitución de caracteres hasta la edad media, época en la cual se impulsaría de forma religiosa el análisis por frecuencia de los árabes, para buscando la frecuencia con la que ciertas palabras aparecían en el texto sagrado del Corán y entender la cronología de las palabras del Profeta. Con esta técnica de análisis por frecuencia se rompieron todos los cifrados por sustitución monoalfabéticos existentes hasta la invención del cifrado polialfabético desarrollado por Leon Battista Alberti en 1465 [32, 33]. Por ejemplo si se quiere codificar el mensaje $M = ENTRELAZAMIENTO$ con el cifrado del César,

entonces se le asigna a cada carácter del alfabeto un peso número, como se muestra en el tabla 2.1.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
Z	Y	X	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W

Tabla. 2.1 Alfabeto y cifrado por sustitución de caracteres: En la tabla se muestran la asignación de caracteres para cifrar por sustitución.

Para cifrar se escoge la clave de substitución que representa al corrimiento en el alfabeto de cada carácter, entonces para $M = ENTRELAZAMIENTO$ y una clave de 3 el mensaje cifrado seria $C = 4, 13, 20, 18, 4, 11, 0, 26, 0, 12, 8, 4, 16, 20, 15$, de donde $C = BKQOBIZWZJFBPQM$, como se muestra en el tabla 2.2. Similarmente para cifrar con una sustitución monográmica polialfabeto, se encogen varios alfabetos sobre los cuales se realizan múltiples sustituciones de forma ordenada.

4	13	20	18	4	11	0	26	0	12	8	4	16	20	15
B	K	Q	O	B	I	Z	W	Z	J	F	B	P	Q	M

Tabla. 2.2 Cifrado por sustitución de caracteres: En la cuadro se muestra el criptosistema cuando se realiza una sustitución simple.

2.1.1. Cifrado por Transposición de Caracteres

El cifrado por transposición es un algoritmo con un diseño geométrico que permite que los caracteres del mensaje permuten y se reorganicen de forma pseudoaleatoria para formar el criptosistema. Estas transposiciones se puede realizar por grupos, filas y columnas. En la codificación por grupos, el mensaje se ordena utilizando una permutación en donde la variable independiente represente la acción sobre los caracteres del mensaje. En la codificación por series, el mensaje se distribuye en una cadena de submensajes que representan el criptosistema con una función o una serie especifica. En la codificación por columnas, el mensaje es agrupado en un número determinado de columnas para genera una transposición de caracteres a una posición adyacente para formar el criptosistema, similarmente en la codificación por filas se distribuye el mensaje en las filas y se transponen para generar el texto cifrado [34]. Por ejemplo, para realizar una transposición por columna, se le asigna un peso numérico a las letras del alfabeto como se muestra en el tabla 2.3.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z

Tabla. 2.3 Peso número del alfabeto: El peso numérico permite tener una clave, donde cada carácter tiene una peso específico.

Para realizar el cifrado se escoger una clave, que para este caso es la palabra (LIBRO=12,9,2,19,16) que determinará la ubicación de las columnas del mensaje, por lo tanto el texto (LO MÁS MARAVILLOSO DE LA CIENCIA ES QUE ESTÁ VIVA) queda ordenado como se muestra en la parte izquierda de la tabla 2.4, y en la parte derecha de esta tabla se muestra la transposición de las columnas del criptograma, según el peso numérico de los caracteres de la clave, es decir, para la clave (LI-

BRO=12,9,2,19,16) se tiene la secuencia de transposición ordenada (2,9,12,16,19=BILOR), que permite genera el siguiente criptograma (MOLSÁRAMVALLIS0EDOALEICCNEAIQSEEUTSIVÁAV).

11	8	1	18	15	1	8	11	15	18
L	I	B	R	O	B	I	L	O	R
L	O	M	Á	S	M	O	L	S	Á
M	A	R	A	V	R	A	M	V	A
I	L	L	O	S	L	L	I	S	0
O	D	E	L	A	E	D	O	A	L
C	I	E	N	C	E	I	C	C	N
I	A	E	S	Q	E	A	I	Q	S
U	E	E	S	T	E	E	U	T	S
Á	V	I	V	A	I	V	Á	A	V

Tabla. 2.4 Cifrado con transposición por columnas: En la parte izquierda de la Tabla se muestra como se organizado el mensaje (LO MÁS MARAVILLOSO DE LA CIENCIA ES QUE ESTÁ VIVA) en un arreglo de 5 columnas dependientes de la clave (LIBRO), y en la derecha de la Tabla se muestra el criptograma por transposicion de columnas (MOLSÁRAMVALLISOEDOALEICCNEAIQSEEUTSIVÁAV) en un arreglo de 5 columnas.

2.1.2. Cifrado por Sustitución de Caracteres

El cifrado por sustitución realiza transformaciones alfabéticas en el mensaje para redistribuir los caracteres de la codificación en una estructura sintáctica sin un contenido lógico aparente. Con este cifrado se realiza una sustitución monográmica monoalfabeto módulo n, es decir se sustituye cada carácter del mensaje por un único elemento del criptograma. Por ejemplo, si M es el mensaje y cada letra del alfabeto tiene un peso numérico entre $(0, 26)$, entonces el mensaje cifrado se obtiene como.

$$C = (a * M + b) \mod n, \tag{2.1}$$

donde a representan una constante de multiplicación denominada constante de decimación, destrucción o aniquilación, y b es una constante de adición. Entonces si se quiere cifrar el carácter $F = 6$ en mód 27, con un multiplicador de $a = 4$ y un desplazamiento de $b = 2$, C será.

$$C = (4 * F + b) \mod n = (4 * 6 + 2) \mod 27 = 26 \mod 27 = 26, \tag{2.2}$$

por lo tanto, la letra F se cifra como Z. Para descifrar se necesita que la constante a tenga inverso multiplicativo mód n y para la constante de desplazamiento b no se tienen restricciones, entonces, el mensaje M se recupera con:

$$M = (C - b) * a^{-1} \mod n. \tag{2.3}$$

Para descifrar se verifica que $inv(a, n) = inv(4, 27) = 7$, de donde:

$$M = (C - b) * a^{-1} \mod n = (26 - 2) * 7 \mod 27 = (24 * 7) \mod 27 = 168 \mod 27 = 6, \tag{2.4}$$

de donde el mensaje descifrado seria $F = 6$. Como se tomo mód 27 entonces a puede tomar los valores: $1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23, 25$ y 26, y b puede tomar los valores de $(0, 26)$.

El cifrado por sustitución también se puede realizar por homófonas[1] y permite que las letras más frecuentes del mensaje tengan varias representaciones, para que el criptograma adquiera una distribución normal que oculta el contenido real de información. Otra substitución que mejora la seguridad de la codificación es la sustitución polialfabeto que utiliza múltiples alfabetos para realizar las sustituciones, en donde la estadística del lenguaje no se presenta de manera clara el criptograma [35]. Sin embargo, todos los sistemas criptográficos que han utilizado cifrado por sustitución o transposición fueron rotos con estadística del lenguaje, que permite analizar la frecuencia con la que se repiten los caracteres en un texto. Por ejemplo, partiendo de la estadística del lenguaje se puede saber el contenido de información de un mensaje cifrado con sustitución monoalfabeto, ya que esta técnica criptográfica desplaza las frecuencias características de las letras manteniendo constante sus valores, es decir, que para el texto ***(será mejor ver pasar el tiempo con la mirada perdida en el horizonte, con vagos pensamientos recorriendo los pasillos de una existencia ya casi olvidada, preguntas sin respuestas como sombras en la oscuridad atravesando los laberintos de los recuerdos, como resolver los interrogantes que se presentan como fantasmas en la noche, las respuestas son tenues entre tantos muros y calabozos)*** se tiene una frecuencia de distribución estadística, en donde el carácter que más se repite es la E con 13,7 % como se muestra en la figura 2.1 [36].

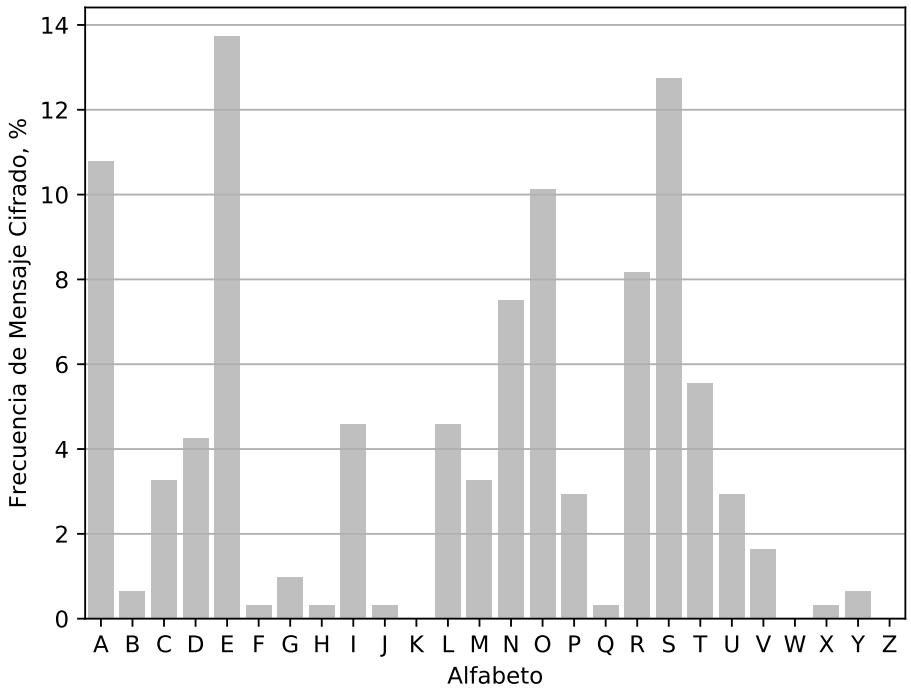

Figura. 2.1 Frecuencia de caracteres en un texto: En la figura se muestra la frecuencia en forma porcentual de la repetición de los caracteres de un texto.

[1]Palabra de igual pronunciación o sonido y distinto significado.

Para obtener el texto cifrado se realiza un desplazamiento de 3 a la derecha del alfabeto, con lo que se tiene el criptograma ***(Hv phmru yhu sdvdu ho wlhpsr frq od pludgd shuglgd hqwuh ydjrv shqvdplhqwrv, uhfruulhqgr orv sdvloorv gh xqd hxlvwhqfld bd fdvl roylgdgd, suhjxqwdv vlq uhvsxhvwdv frpr vrpeudv hq od rvfxulgdg dwudyhvdqgr orv odehulqwrv gh orv uhfxhugrv, frpr uhvroyhu orv lqwhuurjdqwhv txh vh suhvhqwdq frpr idqwdvpdv hq od qrfkh, odv uhvsxhvwdv vrq whqxhv hqwuh wdqwrv pxurv b ghvhr)***, y obteniendo la frecuencia de los caracteres en el criptograma —ver la figura 2.2— se observa que la mayor frecuencia le corresponde a la letra H con una frecuencia de 13,7 % idéntica a la frecuencia de la letra E del texto sin cifrar, por lo tanto de las figuras 2.1 y 2.2 se puede decir que la frecuencia del texto sin cifrar se desplaza al criptograma con el mismo patrón, lo que permite encontrar similitudes entre los caráteres para romper la codificación [36].

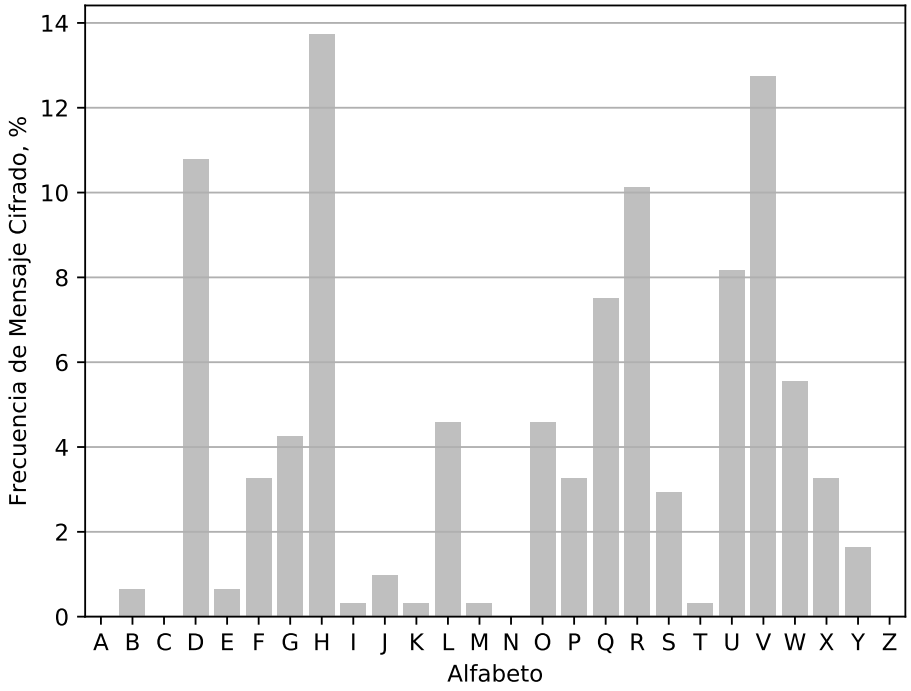

Figura. 2.2 Frecuencia del criptograma cifrado por un dezplazamiento 3 a la derecha: En la figura se muestra la frecuencia de los caracteres en forma porcentual del criptograma de texto cifrado. Note que corresponde a la misma distribución del texto original 2.1. desplazada.

Los cifrados por sustitución y transposición permanecieron inalterados por un largo periodo, con la notable excepción de los trabajos de Charles Babbage sobre criptoanálisis matemático de los cifrados polialfabético en la guerra de Crimea, y es hasta la primeras décadas del siglo XX donde se empezarían a utilizar nuevas ideas para codificar información [37]. En 1917 aparece el cifrado de Vernam, que podría ser la primera versión de una técnica de cifrado perfecta, y a partir de 1918 se comenzaría a revolucionar la forma de compartir información con sistemas mecánicos y eléctricos. Apareció la máquina Enigma usada por el ejército alemán para cifrar y descifrar los mensajes dirigidos a las tropas, y a la par que se desarrollaba la criptografía, el criptoanálisis permitía encontrar las debilidades de los sistemas criptográficos para romper su seguridad sin el conocimiento previo de la información secreta.

2.1.3. Cifrado de Vernam

El cifrado de Vernam o libreta de un solo uso, permite combinar un texto en claro con una clave de igual longitud para generar un criptosistema con una seguridad tan eficiente que Joseph Mauborgne reconoció, que si el algoritmo utilizaba una clave completamente aleatoria se incrementaría la dificultad criptoanalítica del sistema hasta el punto de ser irrompible [38]. Para el cifrado, cada carácter del mensaje se transforma en una cadena binaria que es operada en una compuerta XOR[2] bit a bit con una clave pseudoaleatoria del mismo tamaño del mensaje. Es decir, que a cada carácter del mensaje se le asignan n bits los cuales se suman en una compuerta XOR mód 2 con una clave de igual longitud como se muestra en la figura 2.3, en donde M_i representan los n bits de cada carácter del mensaje y K_i la clave, entonces el mensaje cifrado C_i se obtiene de la siguiente operación:

$$C_i = M_I \oplus K_i, \text{ para } i = 1, 2, ..., n. \tag{2.5}$$

Para el proceso de descifrado se utiliza la propiedad involutiva de la compuerta XOR:

$$C_i \oplus K_i = (M_i \oplus K_i \oplus K_i), \tag{2.6}$$

dado que $K_i \oplus K_i = 0$, para cualquier K_i, entonces el mensaje descifrado se obtiene con [39]:

$$C_i \oplus K_i = M_i, \tag{2.7}$$

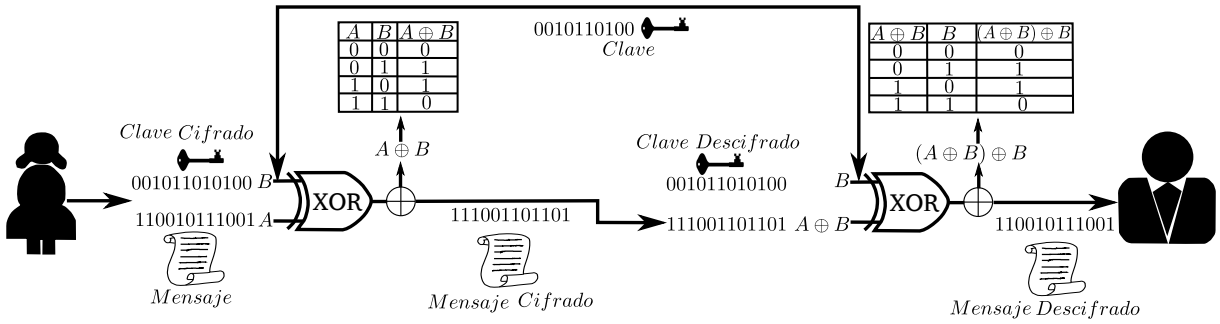

Figura. 2.3 Cifrado de Vernam: Cifrado y descifrado de un mensaje utilizando una puerta XOR.

2.1.4. Máquina de Cifrado

Las maquinas de cifrado se utilizaron extensamente en la segunda guerra mundial para codificar la información de las fuerzas militares, razón por la cual, en este periodo también se desarrollaron avances significativos en el análisis para romper los cifrados. Por ejemplo, los alemanes usaron una maquina de rotores electrónica llamada Enigma, que apareció patentada en 1918 por la empresa alemana Scherbius y Ritter, cofundada por Arthur Scherbius, quien compro la patente de un inventor neerlandés y se puso a la venta en 1923 para uso comercial [40]. En 1926 la Armada alemana la adoptó para uso militar y por la seguridad de su cifrado fue integrada a todas sus fuerzas armadas para codificar la información. Esta máquina usaba un mecanismo rotatorio, compuesto por una parte eléctrica y una mecánica, que permitían cifrar mensajes con alta seguridad, hasta 1929 que el matemático Marian Rejewski de la oficina de cifrado polaca reconstruyó la maquina Enigma del ejército alemán y se pudieron descifrar cientos de códigos [41, 42]. Pero en 1938 los alemanes

[2]La lógica de la XOR utiliza algebra booleana $x = A * \overline{B} + \overline{A}B = A \oplus B$ es decir, si A y B son 0 o 1 entonces la salida es $x = 0$, pero si $A = 0, B = 1$ o $A = 1, B = 0$ entoces la salida $x = 1$.

cambiaron el método para codificar información, lo que dio origen al primer computador mecánico construido por los polacos para analizar código, y finalmente los criptólogos británicos como Gordon Welchman y Alan Turing terminaron descifrado Enigma en 1942 [43, 44], hecho que les permitió a los criptógrafos norteamericanos y británicos romper los sistemas criptográficos de la armada Japonesa JN-25 y Púrpura [45, 46].

2.2. Criptografía Moderna

Después de la guerra empezaría la revolución de la criptografía con Claude Shannon, quien es considerado el padre de la criptografía matemática, por sus publicaciones Communication Theory of Secrecy Systems de la Bell System Technical Journal en 1949 [47], y el libro Mathematical Theory of Communication, con Warren Weaver [48]. Con estos trabajos y los de la teoría de la información y la comunicación se completaría la base teórica para desarrollar lo que se conoce como la criptografía moderna, la cual utiliza tres modelos para codificar y garantizar la integridad y autenticidad de la información: La criptografía simétrica codifica la información con una sola clave, la criptografía asimétrica cifra y descifra la información a partir de dos claves relacionadas, y las funciones hash que permiten verificar la integridad de la información y crear los bloques blockchain [49, 50].

2.3. Criptografía Simétrica

La criptografía simétrica utiliza un cifrado de clave única que permite codificar y decodificar la información, como se muestra en la figura 2.4 donde se esquematiza el funcionamiento general de estos algoritmos integrados por un emisor Alice y un receptor Bob. Para cifrar información con este sistema primero se genera y se comparte una clave que permitirá codificar el mensaje entre Alice y Bob, luego para decodificar el mensaje se utiliza la mismo clave que se uso para cifrar. En cuanto a la seguridad de estos criptosistemas, se pueden presentar problemas cuando se utiliza varias veces la misma clave de cifrado, ya que si el mensaje es interceptado se puede encontrar rastros de la clave y con esto descifrar el contenido de información del mensaje [51]. Todos los algoritmos de clave secreta utilizan cifrado en bloque y cifrado en flujo para realizar las operaciones de transposición y sustitución de caracteres en los criptosistemas, esta técnica tiene una base matemática fundamentada en la teoría de números, la teoría de la información, y la estadística, que soportan la complejidad algorítmica de la seguridad de la información en la actualidad, y es lo que ha permitido mejorar aspectos como confidencialidad, integridad y disponibilidad de la información. Además estos nuevos algoritmos permiten una mejor eficiencia en la optimización de recursos y la seguridad ofrecida.

Figura. 2.4 Cifrado simétrico: El emisor Alice Y el receptor Bob comparten la clave para cifrar y descifrar los mensajes. Para el proceso de cifrado, Alice o Bob codifica el mensaje, el que recibe la información decodifica con la misma clave.

2.3.1. Algoritmo DES

El algoritmo DES (Data Encryption Standard), utiliza cifrado por bloques fijos de 64 bits y una clave de cifrado con una longitud de 64 bits; 8 de los bits de los bloques son para realizar la verificación de paridad del criptosistema [52]. Una modificación del algoritmo DES es el triple DES, algoritmo conformado por la integración de tres DES simultáneos y una clave de 168 bits para mejorar la seguridad. Sin embargo, estos algoritmos son vulnerables y no soportan ataques computacionales, un computador con procesamiento básico puede descifrar las claves en tiempos que oscilan alrededor de unas horas; también las claves se pueden obtener con el criptoanálisis diferencial, técnica estadística para analizar parejas del texto cifrado y obtener patrones característicos de la clave [53]. Con esta técnica se pueden construir ataques por aproximación simple al cifrado por bloques, donde al operar las parejas de texto se obtiene la clave de cifrado simplificado que corresponde con alta probabilidad a la clave de cifrado real [54].

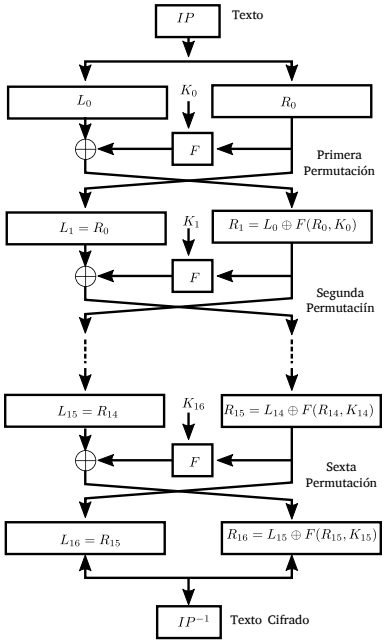

Figura. 2.5 Algoritmo DES: cifrado por bloques de 64 bits, usando una clave de 64 bits, de los que 8 son de paridad, produciendo así 64 bits cifrados. La clave es de 66 bits, pero viene expresada en una cadena de 64 bits, Los espacios de los textos originales y los textos cifrados son $M = C = \{0,1\}^{64}$, y el espacio de claves es $k = (b_1, ..., b_n) \in \{0,1\}^{64}.Tambinsecumpleque \sum_{i=1}^{8} b_{8k+i} \equiv 1 \pmod 2, 0 \leq k \leq 7$ donde $b_{8i}, (1 \leq i \leq 8)$ son los bits de paridad [43].

2.3.2. Algoritmo IDEA

El algoritmo IDEA (International Data Encryption Algorithm) fue desarrollado 1991 por Xuejia Lai y James L. Massey, para reemplazar el algoritmo DES. IDEA está conformado por bloques simétricos de 64 bits y utiliza una clave de 128 bits que se opera en tres funciones: una XOR, una suma módulo 2 (base 16), y el producto modulo $2(base16) + 1$, que se agrupan en ocho rondas para el proceso de cifrado. El algoritmo divide el bloque de texto p de 64 bits en cuatro partes p_1, p_2, p_3, p_4 de 16 bits cada uno, que se mezclan con una clave k_i que representa las 52 subclaves de 16 bits necesarias para realizar las ocho rondas iguales, donde se utilizan 6 subclaves de 16 bits cada una y una transformación de salida denominada media ronda formada por 4 subclaves. Para este proceso, las primeras ocho subclaves se calculan dividiendo la clave de entrada en bloques de 16 bits, las

siguientes ocho se calculan rotando la clave de entrada 25 bits a la izquierda y volviendo a dividirla, y así sucesivamente se continúa con el proceso de obtención de la clave [55].

Para el proceso de descifrado, las subclaves se obtienen cambiando el orden de k_i y calculando el inverso de las operaciones. En la figura 2.6 se muestra el funcionamiento de una de las rondas del algoritmo IDEA, es considerado uno de los cifrado por bloques más seguro, pero sin embargo, puede ser vulnerado con la técnica de ataque por saturación, permite que una parte del mensaje se mantenga constante mientras se realizar un barrido en la otra, para obtener indicios del mensaje cifrado. Por ejemplo, si un ataque usa 256 textos elegidos que tienen 8 de sus bits diferentes, y el resto iguales y el conjunto de los textos elegidos tiene una suma $XOR = 0$, entonces el texto cifrado muestra información del mensaje [56, 57].

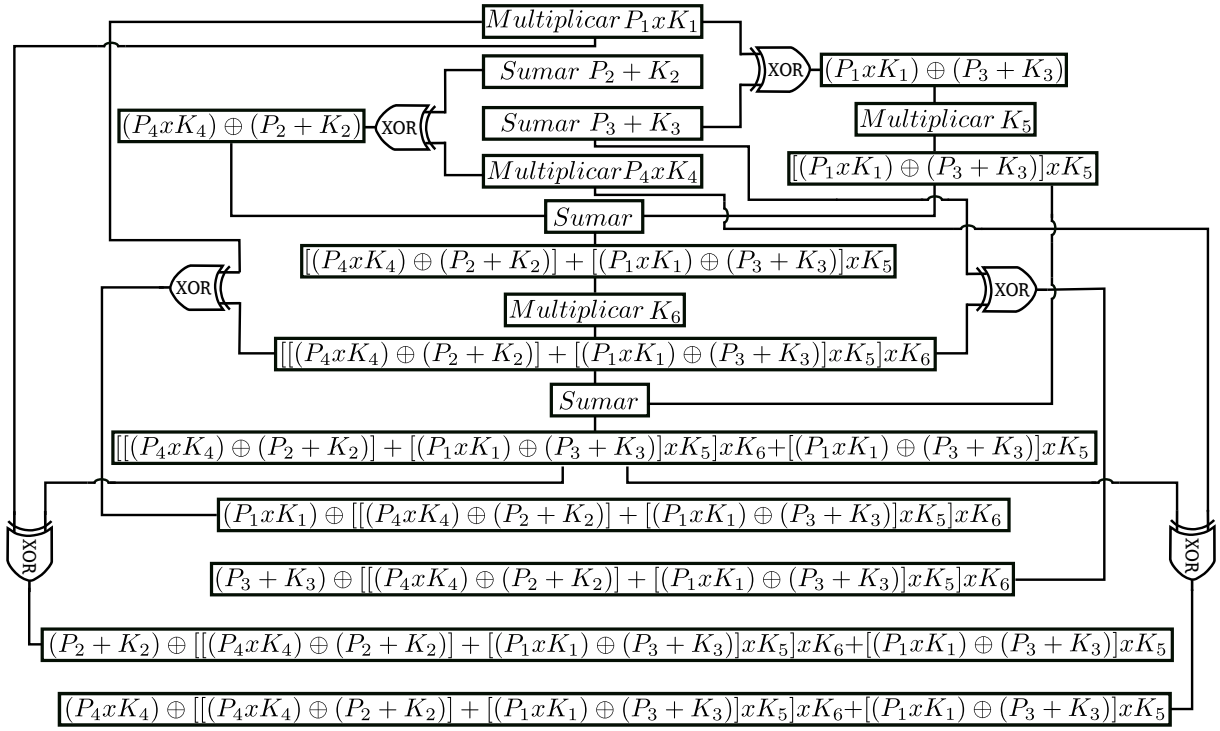

Figura. 2.6 Algoritmo IDEA: En la figura se muestra la salida de una ronda de cifrado que luego entra al siguiente ciclo, en el que se emplearán las siguientes seis subclaves, hasta un total de 48. Después de la octava iteración, se realiza la siguiente transformación: 1) Multiplicar $p_1 \times k_{49}$, 2) Sumar $p_2 + k_{50}$, 3) Sumar $p_3 + k_{51}$, 4) Multiplicar $p_2 \times k_{52}$, figura modificada del trabajo [57].

2.3.3. Algoritmo AES

El algoritmo AES (Advanced Encryption Standard) nace en 1997, a partir de un concurso propuesto por el NIST (National Institute of Standards and Technology), quien escogió en 1999 cinco algoritmos finalistas: RIJNDAEL, MARS [58], RC6 [59], SERPENT [60], TWOFISH [61], y en octubre de 2000, la NIST dio como ganador a RIJNDAEL, que está formado por bloques fijos simétricos de 128 bits y claves que son múltiplo de 32 bits, con un mínimo de 128 bits y un máximo de 256 bits. El algoritmo RIJNDAEL es el mismo AES estándar (FIPS 197) de la NIST [62], que opera con claves de 128, 192 o 256 bits, en una matriz de 4×4 bytes, llamada state [63]. A continuación se describe el algoritmo que dio origen al estándar (FIPS 197) de criptografía simétrica.

El algoritmos AES divide el texto plano P en palabras de 8 bits que forman una secuencia $(p_7, p_6, p_5, p_4, p_3, p_2, p_1, p_0)$ que representa un byte en un campo finito como un polinomio de la forma $\sum_{i=0}^{7} = p_i x^i$, en donde cada binario 01010111 tiene un polinomio asociado $x^6+x^4+x^2+x^1+1$. También se puede utilizar una notación hexadecimal para representar cada byte como se muestra en la tabla 2.5, la cual permite escribir un byte como una cadena secuencial $(P_{15}, P_{14}, P_{13}, P_{12}, P_{11}, P_{10}, P_9, P_8, P_7, P_6, P_5, P_4, P_3, P_2, P_1, P_0)$ que origina una matriz de estado, donde $P_{3,3}$ es el byte más significativo y $P_{0,0}$ el byte menos significativo, como se muestra en la ecuación 2.8. Similarmente se construye una matriz de estados para la clave K ecuación 2.9, que puede tener una longitud de 128, 192, o 256 bits, agrupadas en subclaves que varían entre 10, 12, y 14 rodas [62].

$$\begin{bmatrix} P_{15} & P_{14} & P_{13} & P_{12} \\ P_{11} & P_{10} & P_9 & P_8 \\ P_7 & P_6 & P_5 & P_4 \\ P_3 & P_2 & P_1 & P_0 \end{bmatrix} = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0} & P_{1,1} & P_{1,2} & P_{1,3} \\ P_{2,0} & P_{2,1} & P_{2,2} & P_{2,3} \\ P_{3,0} & P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix} \tag{2.8}$$

$$\begin{bmatrix} K_{0,0} & K_{0,1} & K_{0,2} & K_{0,3} \\ K_{1,0} & K_{1,1} & K_{1,2} & K_{1,3} \\ K_{2,0} & K_{2,1} & K_{2,2} & K_{2,3} \\ K_{3,0} & K_{3,1} & K_{3,2} & K_{3,3} \end{bmatrix} \tag{2.9}$$

Cadena de bits	Carácter hexadecimal	Cadena de bits	Carácter hexadecimal
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Tabla. 2.5 Notación hexadecimal para los byte. La tabla establece la notación hexadecimal para representar una secuencia; por ejemplo, la cadena 01011111 corresponde a $5F$ en hexadecimal.

El algoritmo AES utiliza para el proceso de cifrado dos operaciones matemáticas básicas: la suma que se realiza con la compuerta XOR que permite realizar sumas binarias como $10111010 + 01100111 = 11011101$, y la multiplicación que presenta problemas al realizarse bit a bit, ya que al multiplicarse dos byte en su mayoría se genera un resultado mayor que 8 bits, por lo tanto, es necesario sacar el módulo del polinomio resultante con el polinomio irreducible[3] de AES, que se define como $m(x) = x^8 + x^4 + x^3 + x^1 + 1$, para obtener un byte nuevamente y ser consistente con el algoritmo. Además el cifrado se ordena en cuatro fases secuenciales, 1) ronda inicial, opera el bloque AddRoundkey, combina la matriz state con las subclave, 2) la i-ésima ronda, está conformada por los bloques SubBytes, ShiftRow, MixColum, y AddRoundkey; en esta fase las rondas pueden ser 10, 12, o 14 dependiendo de la longitud de la clave, 3) la ronda final está conformada por los bloques SubBute,

[3]Un polinomio es irreducible si es divisible únicamente por la unidad y por él mismo.

ShiftRow, y AddRoundkey, 4) expansión de clave se encarga de generar las diferentes subclaves de la ronda K_i, como se muestra en la figura 2.7 [64, 65].

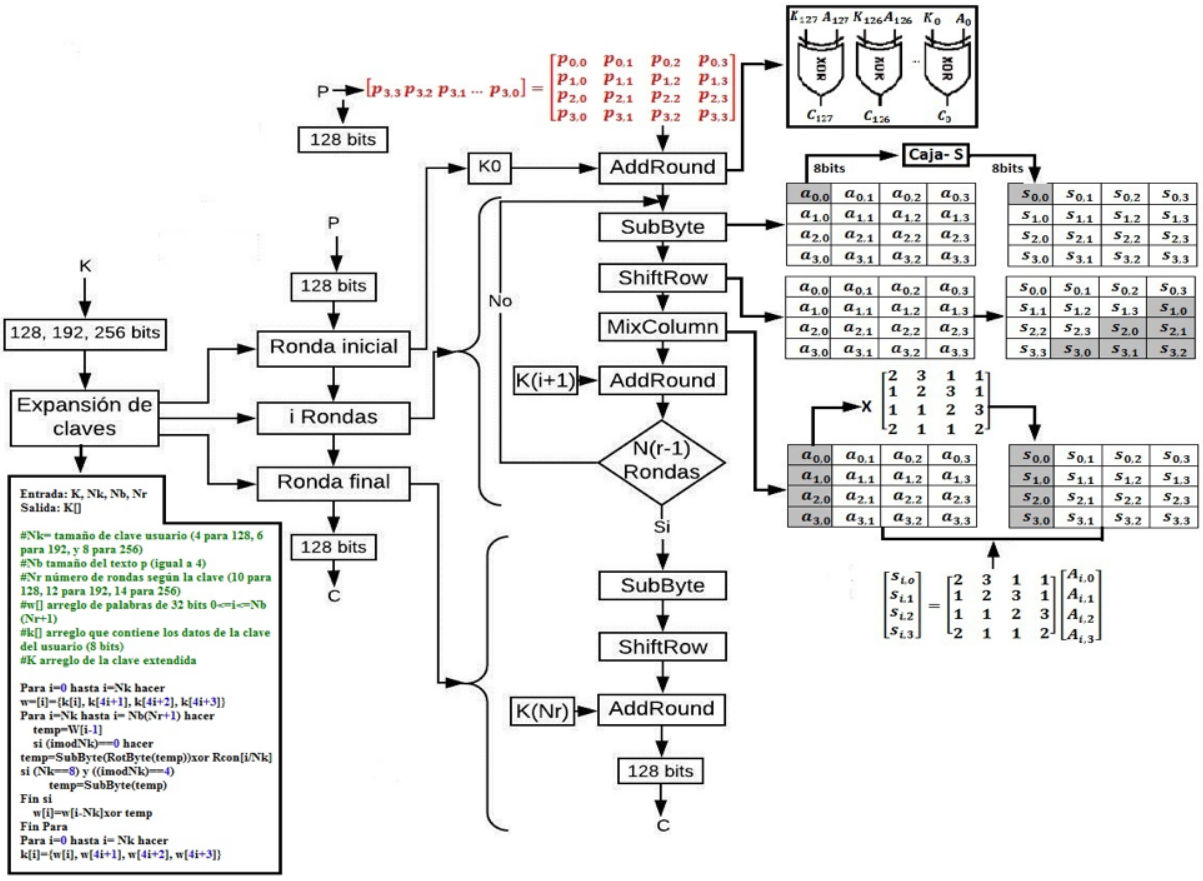

Figura. 2.7 Cifrado del Algoritmo AES: En la figura se muestra el funcionamiento de los bloques de cifrado: donde la ronda inicial, el bloque AddRoundk combina a P y K en una XOR. Seguidamente en las nueve rondas principales siguientes se aplican las cuatro operaciones de cifrado en orden: SubBytes, ShitfRows, Mix- Columns y AddRoundKey, y por último, se realiza la ronda final, en la que se aplican las operaciones SubBytes, ShiftRows y AddRoundKey, para obtener el texto cifrado. Es importante saber que en cada ronda se utilizan diferentes subclaves, que son derivadas de la clave original, de tal modo que la ronda inicial utiliza la clave original (si la clave es de 128 bits) entonces las ronda final utiliza la subclave número 10. Por otra parte el bloque SubBytes realiza sustituciones con la caja S de la tabla 2.7, además SubBytes y RotBytes, realizan una rotación cíclica de la palabra en la entrada, donde $Rcon$ es el vector de datos y $Rcon[i] = \{Rc[i], 0, 0, 0\}$, $Rconi = Rci$, es de 32 bits. Rc es un vector constante de 8 bits representado por $Rc = 0 \times 01, 0 \times 02, 0 \times 04, 0 \times 08, 0 \times 10, 0 \times 20, 0 \times 80, 0 \times 13, 0 \times 36$, figura modificada del estándar [62].

- **SubBytes**: es una operación no lineal de sustitución que transforma el byte en el bits menos significativo mediante la expresión $S'_{i,j} = M * S^{-1}_{i,j} + C_{i.j}$, donde M se representa con una matriz de 8×8 bits, C es un vector de 8×1 bits —ver la ecuación 2.10—, y $S^{-1}_{i,j}$ es el multiplicativo inverso del byte. $S'_{i,j}$ se obtiene de la tabla 2.7 de sustitución.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}. \tag{2.10}$$

- **ShiftRows**: representa una operación de rotación cíclica a la izquierda de la matriz de estados resultante, la primera fila no se modifica, la segunda fila se corre a la izquierda una posición, la tercera fila se corre a la izquierda dos posiciones, y la cuarta fila se corre tres posiciones a la izquierda como se muestra en la tabla 2.6.

a_{15}	a_{11}	a_7	a_3	$\longrightarrow$	a_{15}	a_{11}	a_7	a_3
a_{14}	a_{10}	a_6	a_2	$\longrightarrow$	a_{10}	a_6	a_2	a_{14}
a_{13}	a_9	a_5	a_1	$\longrightarrow$	a_5	a_1	a_{13}	a_9
a_{12}	a_8	a_4	a_0	$\longrightarrow$	a_0	a_{12}	a_8	a_4

Tabla. 2.6 Operación ShiftRows: En la tabla se muestra el corrimiento a la izquierda en cada una de las filas de la matriz de estados.

- **MixColumns**: opera cada columna de la matriz de estados como un polinomio $x^4 + 1$ al que se le saca el módulo con el polinomio $a(x) = (03)x^3 + (01)x^2 + (01)x + (02)$. Matematicamente la operación MixColumns se representa con el producto de las siguientes matrices:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} a_{15} & a_{11} & a_7 & a_3 \\ a_{14} & a_{10} & a_6 & a_2 \\ a_{13} & a_9 & a_5 & a_1 \\ a_{12} & a_8 & a_4 & a_0 \end{bmatrix}. \tag{2.11}$$

- **AddRoundkey**: esta operación combina en una XOR el resultado la matriz de estado que proviene de la transformación en el bloque MixColumn, con una subclave que se genera a partir de la clave principal del sistema para esa ronda, el resultado se muestra en la siguiente expresión:

$$\begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} \oplus \begin{bmatrix} k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{bmatrix} \tag{2.12}$$

$$= \begin{bmatrix} S_{0,0} \oplus k_0 & S_{0,1} \oplus k_1 & S_{0,2} \oplus k_2 & S_{0,3} \oplus k_3 \\ S_{1,0} \oplus k_4 & S_{1,1} \oplus k_5 & S_{1,2} \oplus k_6 & S_{1,3} \oplus k_7 \\ S_{2,0} \oplus k_8 & S_{2,1} \oplus k_9 & S_{2,2} \oplus k_{10} & S_{2,3} \oplus k_{11} \\ S_{3,0} \oplus k_{12} & S_{3,1} \oplus k_{13} & S_{3,2} \oplus k_{14} & S_{3,3} \oplus k_{15} \end{bmatrix}. \tag{2.13}$$

	0x	1x	2x	3x	4x	5x	6x	7x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
0x	63	7C	77	7B	F2	6B	6F	C3	30	01	67	2B	FE	D7	AB	76
1x	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2x	B7	FD	93	26	36	3F	F7	CC	3A	A5	E5	F1	71	D8	31	15
3x	04	C7	23	C3	78	96	05	9A	07	12	80	E2	EB	27	B2	73
4x	D9	81	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5x	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6x	D0	FF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7x	51	A3	40	8F	82	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8x	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9x	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
Ax	ED	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
Bx	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
Cx	BA	78	25	2E	1C	A6	B4	C6	E8	D0	74	1F	4B	BD	8B	8A
Dx	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
Ex	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
Fx	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Tabla. 2.7 Caja S de sustitución: En la tabla se muestran las operaciones de sustitución, las filas corresponden al primer dígito y las columnas al segundo. Por ejemplo, si se tiene el dato 0×61 con la caja de sustitución S se obtiene la sustitución $0 \times EF$ [62].

Para el proceso de descifrado en el algoritmo AES, se aplican en orden las operaciones inversas del cifrado, teniendo presente que como AES es simétrico entonces se utiliza la misma clave del cifrado. Por lo tanto para descifrar un mensaje se tienen nuevamente cuatro operaciones: 1) ronda inicial, operan los bloque AddRondkey, InvShiftRow, y InvSubByte que realizan una sustitución de la matriz de estado, utilizando la caja S inversa que se muestra en la tabla 2.8, 2) i rondas inversas que puede estar formada por 10, 12 o 14 rondas dependiendo del cifrado, y está formada por los bloques: AddRound, InvMixColumn, InvSubByte, 3) la ronda final que que opera el bloque AddRound, 4) y expansión de claves que es la misma del algoritmo de cifrado. En la figura 2.8 se muestran los operaciones entre los bloque del descifrado AES [66].

- **InvSubBytes**: similarmente que SubBytes, esta es una operación no lineal de sustitución que transforma el byte en el bits menos significativo mediante la ecuación de transformacional inversa $S' = (M^{-1} * (S_{i,j} + C))^{-1}$, donde $S'_{i,j}$ utiliza la tabla 2.9 para realizar las transformaciones.
- **InvShiftRows**: realiza una rotación en dirección opuesta a la operación ShiftRows, es decir, que se realiza una rotación cíclica a la derecha en las filas de la matriz de estado, de tal manera que la primera fila permanece inalterada, la segunda fila se rota una posición, la tercera fila rota dos posiciones, y la cuarta fila rota tres posiciones como se muestra en la tabla 2.8.

a_{15}	a_{11}	a_7	a_3	$\longrightarrow$	a_{15}	a_{11}	a_7	a_3
a_{14}	a_{10}	a_6	a_2	$\longrightarrow$	a_2	a_{14}	a_{10}	a_6
a_{13}	a_9	a_5	a_1	$\longrightarrow$	a_5	a_1	a_{13}	a_9
a_{12}	a_8	a_4	a_0	$\longrightarrow$	a_8	a_4	a_0	a_{12}

Tabla. 2.8 Operación InvShiftRows: En la tabla se muestra el corrimiento a la derecha de las filas en la matriz de estados.

- **InvMixColumns**: realiza la operación inversa del bloque MixColumns, cada una de las columnas se opera como un polinomio y se multiplica el módulo de $x^4 + 1$, con el polinomio $a^{-1}(x) = (OB)x^3 + (0D)x^2 + (09)x + (0E)$, como se muestra en la siguiente operación matricial:

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \cdot \begin{bmatrix} a_{15} & a_{11} & a_7 & a_3 \\ a_{14} & a_{10} & a_6 & a_2 \\ a_{13} & a_9 & a_5 & a_1 \\ a_{12} & a_8 & a_4 & a_0 \end{bmatrix} . \tag{2.14}$$

- **InvAddRoundKey**: es la operación inversa del AddRoundKey donde la clave de cifrado se adiciona a la matriz de estado mediante operaciones XOR byt a byte, y cada una de las rondas una subclave es derivada de la clave principal. Entonces el el bloque InvAddRoundKey realizan operaciones similares a las del proceso de cifrado, pero con elementos diferentes y mediante rondas en sentido inverso.

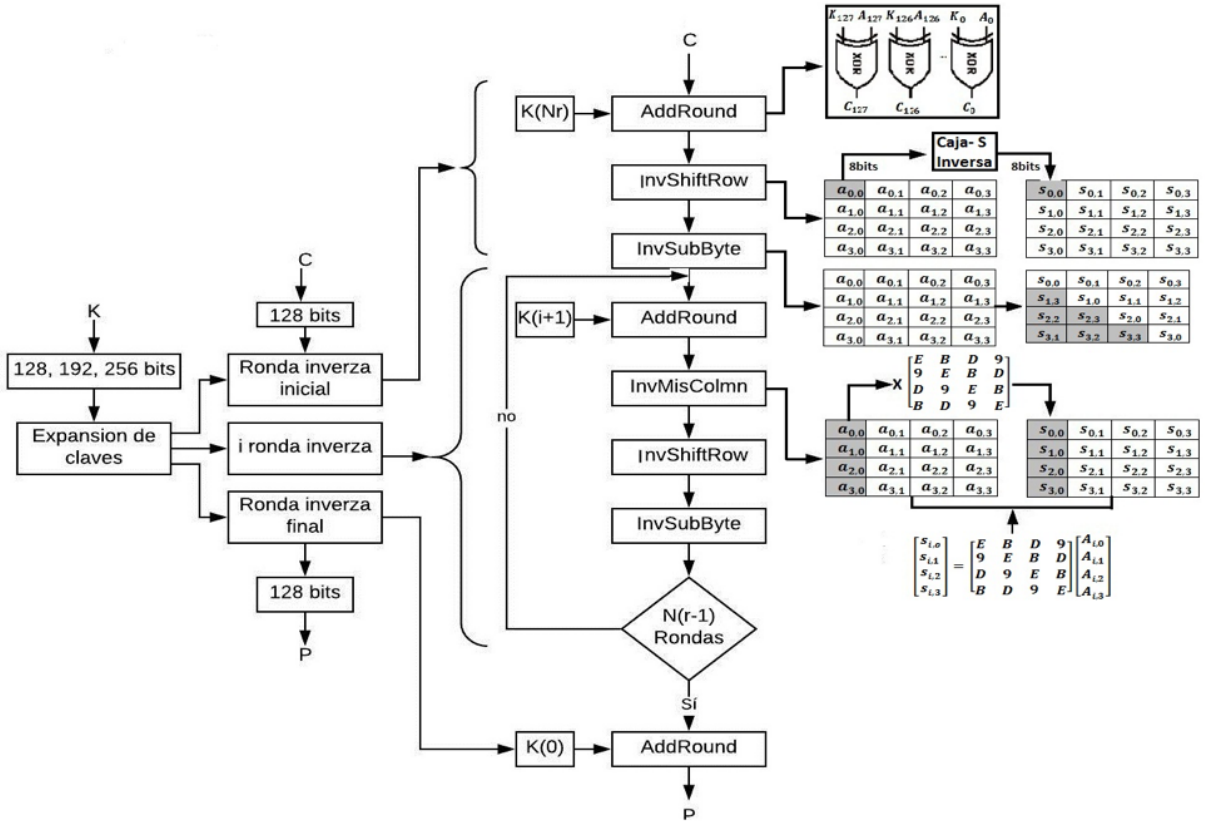

Figura. 2.8 Algoritmo de descifrado AES: En la figura se muestra la estructura ordenada del funcionamiento de descifrado que esta que opera en forma inversas los bloques de cifrado, el descifrado utiliza cuatro etapas: 1) agrupa la ronda inicial con los bloque AddRond, InvShiftRow, y InvSubByte, 2) se tienen las i rondas inversa formada por 10, 12 o 14 rondas dependiendo del cifrado, están agrupados los bloques, AddRound, InvMixColumn, InvSubByte, 3) ronda final, opera el bloque AddRound, 4) se tiene la misma expansión de claves del cifrado, figura modificada del estándar [62].

	0x	1x	2x	3x	4x	5x	6x	7x	8x	9x	Ax	Bx	Cx	Dx	xE	Fx
0x	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1x	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2x	54	7B	94	32	A6	C2	23	3D	FE	4C	95	0B	42	FA	C3	4E
3x	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4x	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5x	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	AB	8D	9B	84
6x	90	DC	AB	00	8C	BC	D3	0A	F7	F4	58	05	B8	B3	45	06
7x	D0	2C	1E	8F	CA	3F	DF	02	C1	AF	BD	03	01	13	8A	6B
8x	3A	91	11	41	AF	67	DC	EA	97	F2	CF	CE	F0	B4	F6	71
9x	95	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
Ax	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
Bx	FC	56	3E	4B	C6	D2	79	29	9A	DB	CD	FE	78	CD	5A	F4
Cx	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
Ex	A0	E0	3B	4D	AE	2A	E5	B0	C8	EB	BB	3C	83	53	99	61
Fx	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	DC	7D

Tabla. 2.9 Caja S Inversa de Sustitución: En la Tabla se muestra como funciona la operación de transposición de la caja inversa que utiliza el bloque InvSubByte para realizar las operaciones de descifrado [62].

2.4. Ataques a la Criptografía Simétrica

AES es uno de los algoritmos de criptografía simétrica más seguros para codificar información, pero partiendo del orden matemático con que se construyen sus bloques de transformación se puede realizar un criptoanálisis que puede vulnerar el algoritmo. Específicamente todos los sistemas de criptografía simétrica son susceptibles a ataques, ya que sus operaciones se basan en funciones invertibles, donde el cifrado y el descifrado dependen de un solo parámetro conocido como la clave criptográfica para mantener la seguridad del criptosistema, el que puede ser atacado con técnicas como las que se describen a continuación.

2.4.1. Criptoanálisis Diferencial

El criptoanálisis diferencial es una técnica estadística que consiste en analizar las parejas del texto cifrado que forman patrones y dan indicios de la clave criptográfica. Este ataque utiliza un conjunto de pares de criptogramas para determinar el valor de los bits de la clave principal. La información de la clave se obtiene al operar los bloques de texto cifrado sobre los bloques de texto en claro con una diferencia de bits específica entre ellos, ya que existe una cantidad de bits determinada entre cada bloque a la salida de una ronda. Con esto se crea una clave supuesta y se verifica la validez del supuesto, para continuar con el proceso para el número de claves sugeridas por un par de bytes de texto en claro, donde el criptograma no es una constante, es decir, el número de claves sugeridas puede variar según el par seleccionado, lo que permite obtener distintas claves que pueden vulnerar el sistema cifrado por bloques simétricos como el algoritmo DES [67].

2.4.2. Criptoanálisis Lineal

El criptoanálisis lineal,inventado por Mitsuru Matsui, es una técnica que trata de crear una aproximación simple al cifrado de bloques como un todo. El criptoanálisis lineal es un ataque que

se basa en pares de bytes de texto en claro para obtener información sobre la clave principal de codificación, que puede encontrarse a partir del cifrado simplificado, ya que el bit de clave del cifrado simplificado corresponde con alta probabilidad al bit de clave del cifrado real. Para realizar el ataque se trata de encontrar expresiones lineal de la forma:

$$P[i_1, i_2, i_3, \cdots, i_n] = C[j_1, j_2, j_3, \cdots, j_n] + K[k_1, k_2, k_3, \cdots, k_n], \tag{2.15}$$

donde P, C, K son vectores que representan el texto en claro, el criptograma, y la clave. En el proceso del ataque también se pueden encantar expresiones lineales como:

$$I[i_1, i_2, i_3, \cdots, i_n] = I_{15}[j_1, j_2, j_3, \cdots, j_n] + K[k_1, k_2, k_3, \cdots, k_n], \tag{2.16}$$

donde I representa los estados intermedios del algoritmo, y si se hace sobre un subconjunto de bits de la subclave en la primera y última ronda se pueden encontrar los bits I_1, I_{15}. Al iterar estos resultados se puede encontrar la clave de codificación principal mediante las siguientes ecuaciones:

$$P[i_1, i_2, i_3, \cdots, i_a] = I_{m-1}[j_1, j_2, j_3, \cdots, j_b] + K[k_1, k_2, k_3, \cdots, k_c], \tag{2.17}$$

$$I_{m-1}[j_1, j_2, j_3, \cdots, j_b] + I_m[m_1, m_2, m_3, \cdots, m_u] = K[k_1, k_2, k_3, \cdots, k_d], \tag{2.18}$$

$$P[i_1, i_2, i_3, \cdots, i_a] + I_m[m_1, m_2, m_3, \cdots, m_u] = K[k_1, k_2, k_3, \cdots, k_d]. \tag{2.19}$$

Ataque que se puede generalizar para los algoritmo DES y 3DES de la siguiente manera [67]:

$$I_{m-1}[j_1 + 32, j_2 + 32, j_3 + 32, \cdots, j_a + 32] = I_i[j_1, j_2, j_3, \cdots, j_a]. \tag{2.20}$$

2.4.3. Criptoanálisis Integral

El criptoanálisis integral, se desarrolló en principio como un ataque cuadrado para los criptosistemas que utilizan cifrado por bloques, este ataque criptográfico evoluciona al ataque conocido como saturación. El criptoanálisis integral tiene las características de un ataque de texto plano que utiliza conjuntos de textos, en donde una parte se mantiene constante y la otra parte realiza un barrido de todas las posibilidades. Un ataque puede usar 256 textos elegidos que tienen 8 de sus bits diferentes, y el resto iguales, donde el conjunto de los textos tiene una suma XOR de 0, luego las sumas XOR de los conjuntos de textos cifrados dan información sobre la operación de cifrado [68].

2.4.4. Ataque por Canal Lateral o Auxiliar

El ataque por canal lateral aprovecha las vulnerabilidades en las implementaciones del sistema criptográfico, para medir y analizar variables como la potencia de radiación de las señales radioeléctricas o la intensidad de sonido generada por la interacción ente el hardware y el software que se están utilizando para el cifrado o descifrado. Es decir, que cuando se utilizan circuitos integrados, o tarjetas inteligentes para proteger la información [69, 70], se pueden generan ataques criptográficos, pues un atacante puede monitorear el comportamiento de las señales radioeléctricas cuando se está realizando una codificación de información entre dispositivos compuestos por entradas y salidas, que al filtrar información cuando interactúan con otros canales físicos como el consumo instantáneo de energía, las emisiones electromagnéticas, o el sonido de las teclas, pueden terminar en ataques controlados en el tiempo de ejecución de los dispositivo electrónicos [71, 72].

2.4.5. Ataque por Plantilla

El ataque por plantilla, es una técnica muy sofisticada y se pueden implementar en cualquier sistema de computación, ya que depende de la capacidad de identificar y clasificar las señales generadas por las teclas de un dispositivo. Este ataque se ejecuta en dos fases: 1) Fase de construcción de plantilla. El atacante construye y almacena plantillas con una longitud de 2^r bits que pueden representar la llave de cifrado del sistema criptográfico, luego cifra m en un tiempo específico y se registra la potencia de m. Posteriormente se seleccionan n puntos como las variables más efectivas. Por ejemplo para la traza i se tiene que $T_i = t_{i,1}, t_{i,2}...t_{i,n}$ con $1 \leq i \leq m$, que es una matriz de señales de $m \times n$.

$$T = \begin{pmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m,1} & t_{m,2} & \cdots & t_{m,n} \end{pmatrix}, \tag{2.21}$$

donde el promedio del punto j se representa con la expresión $\bar{t}_{k,y} = (1/m)\sum_{i=1}^{m} t_{i,j}$ y la media del vector es $M_k = (\bar{t}_{k,1}, \bar{t}_{k,2}, ..., \bar{t}_{k,n})$, con $0 \leq k \leq 2^r - 1$. Por lo tanto la matriz de covarianza de T estará dada por:

$$T_c = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,n} \end{pmatrix}. \tag{2.22}$$

Como $c_{p,q} = (1/(m+1))\sum_{i=1}^{m}(t_{i,p} - \bar{t}_{i,p})(t_{i,q} - \bar{t}_{i,q})$, se puede tener la representación de la plantilla en función de la llave $h_k = (M_k, c_k)$. 2) Fase de coincidencia de plantillas, en esta fase se analiza el porcentaje de coincidencia entre la señal y la plantilla construida, en este proceso se verifica la máxima probabilidad de éxito de la señal desconocida de la siguiente manera: $p(x|h_k) = 1/[(2\pi)^{n/2} det(c_k)^{1/2}] \exp\big(-1/2(x - M_k)^T c_k^{-1}(x - M_k)\big)$ [71].

2.4.6. Ataque Cuadrado

El ataque cuadrado se realiza a partir de una expansión conocida de dos textos que tienen una diferencia XOR que no es cero en un byte. En este proceso se puede observar que después de una ronda, la diferencia XOR entre los estados intermedios, una de las columnas de la matriz de estado tiene una diferencia distinta de cero. Además, también se puede observar que después de la segunda ronda la diferencia XOR se propaga a todos los bytes en la matriz de estado, tal como se muestra en la ecuación 2.23, donde la primera matriz representa el texto sin formato, la segunda matriz representa la operación en la primera ronda, y la tercera matriz representa la operación en la segunda ronda:

$$\begin{bmatrix} \zeta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 2\theta & 0 & 0 & 0 \\ 3\theta & 0 & 0 & 0 \\ \theta & 0 & 0 & 0 \\ \theta & 0 & 0 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 2\alpha & \beta & \gamma & 3\delta \\ 3\alpha & 2\beta & \gamma & \delta \\ \alpha & 3\beta & 2\gamma & \delta \\ \alpha & \beta & 3\gamma & 2\delta \end{bmatrix}. \tag{2.23}$$

De lo anterior se concluye que la diferencia XOR entre dos textos, después de dos rondas, siempre será cero para cualquier byte, comportamiento que se propaga hasta el bloque MixColumns y puede ser aprovechado para construir el ataque Cuadrado. Por ejemplo, si se consideran 256 textos distintos que tienen quince bytes iguales, entonces se muestra que después de calcular las dos primeras rondas

de AES, la diferencia XOR es cero, es decir, que en los 256 estados intermedios, los bytes en cada índice contendrán uno de cada valor posible, de donde la suma XOR de los 256 bytes serán cero, propiedad que se mantendrá hasta la operación SubBytes. Con estos ataques se puede romper AES con 6 y 7 rondas y se puede extender hasta 9 rondas, con 2^{77} textos a cifrar, 2^{56} claves relacionadas, y 2^{224} procesos de cifrado [73].

2.4.7. Ataque por Colisiones

El ataque por colisiones se realiza en las columnas de la mezcla, puesto que las operaciones de transformación son lineales, y todos los cálculos pertenecen al grupo $GF(2^8)$, que se representa con polinomios sobre $GF(2)$ módulo de $m(x) = x^8 + x^4 + x^3 + x + 1$, donde las columnas de la matriz de estados se definen como polinomios sobre $GF(2^8)$ que se multiplican con el módulo de $m(x) = y^4 + 1$, para generar el siguiente polinomio de entrada fija:

$$c(y) = (03)y^3 + (01)y^2 + (01)y + (02), \tag{2.24}$$

donde $01, 02$, y 03 son los elementos de $GF(2^8)$ para $1, x, x + 1$ respectivamente. Por ejemplo, si se define la entrada como $a(y)$ y la salida como $b(y)$, entonces la trasformación de la columna de las mezclas se representa de la siguiente manera:

$$b(x) = a(y) + c(y) \quad \text{mód } (y^4 + 1), \tag{2.25}$$

y de forma matricial las columnas de mezcla se pueden representar como:

$$\begin{bmatrix} b_{00} \\ b_{10} \\ b_{20} \\ b_{30} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} a_{00} \\ a_{10} \\ a_{20} \\ a_{30}, \end{bmatrix}, \tag{2.26}$$

siendo $b_{00} = (02).a_{00} + (03).a_{10} + (01).a_{20} + (01).a_{30}$ el primer byte de la salida, y para la primera ronda $a_{00}, a_{10}, a_{20}, a_{30}$ se puede remplazar por $S(p_{00} + k_{00}), S(p_{11} + k_{11}), S(p_{22} + k_2), S(p_{33} + k_{33})$, y el byte b_{00} de la salida para esta ronda se puede escribir como:

$$b_{00} = (02).S(p_{00} + k_{00}) + (03).S(p_{11} + k_{11}) + (01).S(p_{22} + k_2) + (01).S(p_{33} + k_{33}). \tag{2.27}$$

Como la idea principal de este ataque es encontrar los pares de texto sin formato con el mismo byte b_{00} de salida, entonces solo se tienen en cuenta los textos $p_{00} = p_{11} = 0$ y $p_{22} = p_{33}$, de donde, si para dos textos en claro $p_{22} = p_{33} = \delta$ y $p'_{22} = p'_{33} = \epsilon \neq \delta$, entonces el byte b_{00} de la salida se puede representar de la siguiente forma:

$$S(\delta + k_{22}) + S(\delta + k_{33}) = S(\epsilon + k_{22}) + S(\epsilon + k_{33}). \tag{2.28}$$

Por último, para materializar el ataque, se tiene que utilizar la técnica de canal auxiliar para medir las colisiones de b_{00} o de cualquier otro byte de salida de las operaciones de transformación de las columnas de mezcla. Con este proceso se establecen los dos byte del texto sin formato p_{22} y p_{33} que tiene un valor aleatorio $\delta = p_{22} = p_{33}$. Para el ataque se cifran estos textos sin formato, se mide la energía utilizada, este proceso se repite nuevamente para nuevos valores aleatorios $\epsilon = p'_{22} = p'_{33}$ diferentes a los valores generados para δ, ϵ, y se cifra nuevamente cada texto sin formato, se vuelve a medir y almacenar la potencia correspondientes de cada texto cifrado, para coleccionarlas de forma cursada y poder detectar colisiones en el byte de salida b_{00}. Cuando se encuentre una coalición se deberá buscar información sobre k_{22} y k_{33} [74].

2.5. Criptografía Asimétrica

Los algoritmos de criptografía asimétricos son técnicas que usan en un nivel básico dos llaves para cifrar y descifrar los mensajes, una llave pública para cifrar que es compartida con todos los usuarios del sistema y una llave privada para descifrar. El funcionamiento del algoritmo asimétrico se muestra en la figura 2.9. La criptografía asimétrica ha sido desarrollada a partir de la teoría de números que proporciona teoremas y definiciones para construir los algoritmos que permiten cifrar información de forma asimétrica. Los algoritmos más representativos en criptografía asimétrica son: el de Diffie-Hellman, el RSA y el de curvas elípticas [75].

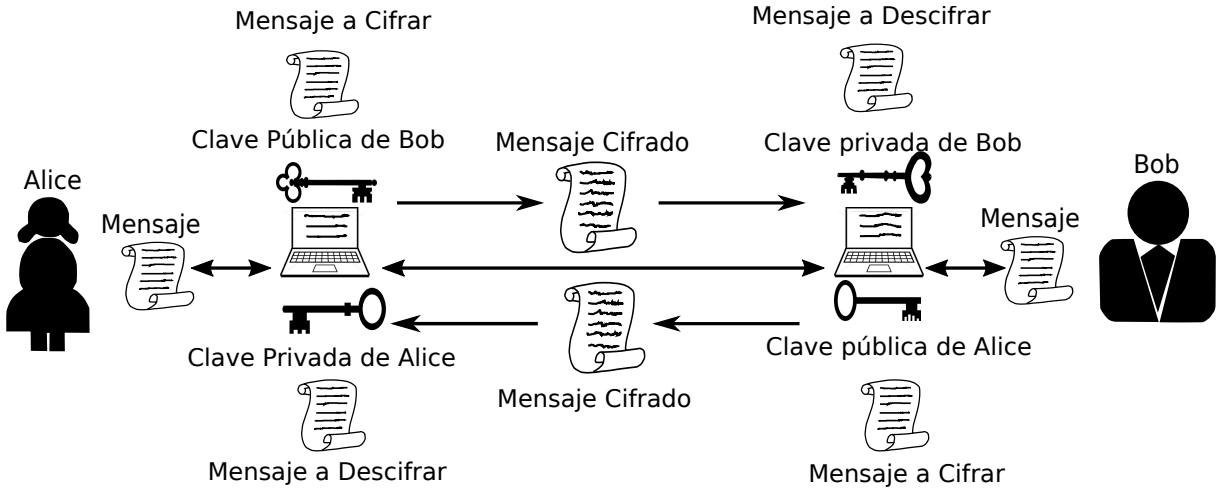

Figura. 2.9 Algoritmos asimétricos: Alice genera dos claves, una privada y una pública, del mismo modo Bob genera las dos llaves que no tienen relación con las de Alice. Alice y Bob comparten su llave pública. Si Alice quiere enviar un mensaje a Bob, cifra la información con la llave pública de Bob y envía el mensaje a través de cualquier canal, Bob recibe el mensaje y lo descifra con su llave privada que solo él conoce.

2.5.1. Algoritmo Diffie-Hellman

El algoritmo Diffie-Hellman es utilizado para realizar intercambios de claves entre usuarios de un sistema sin tener una comunicación previa ni autenticación. La efectividad de este algoritmo depende la capacidad de computar algoritmos discretos que se definen de la siguiente manera: sea a una raíz primitiva de p, las potencias de a que generan los enteros desde $1, \ldots, p-1$, donde a mód p, a^2 mód $p, \ldots, a^{p-1}$ mód p son diferentes. Luego para generar las llaves pública y privada se escoge un número q primo de tal manera que $\alpha < q$ donde α y q representan los coeficientes públicos del sistema. Por ejemplo, para un par de usuarios como Alice o Bob, Alice escoge una clave privada $x_A < q$ y calcula la clave pública con la ecuación $y_A = \alpha^{xA}$ mód q. Similarmente Bob escoge una clave privada $x_B < q$ y calcula la clave pública con $y_B = \alpha^{xB}$ mód q. Para realizar el intercambio de claves, Alice y Bob comparten sus claves públicas, Alice realiza la operación $k = (y_B)^{yB}$ mód q y Bob $k = (y_A)^{yA}$ mód q, y se puede verificar que $(y_B)^{yB}$ mód $q = (\alpha^{xB})^{xA}$ mód $q = (\alpha^{xBxA}) = (\alpha^{xA})^{xB}$ mód $q = (x_A)^{xB}$ mód q, de donde se obtienen que las dos claves k son iguales [76].

2.5.2. Algoritmo RSA

El algoritmo RSA (Rivest, Shamir y Adleman), es el algoritmo de clave pública más utilizado en internet desde hace tres décadas. Se utiliza para transmitir datos a través de canales de comunicación

públicos inseguros y está construido con conceptos matemáticos como el anillo de los enteros y la existencia y unicidad de la descomposición de factores primos de un entero, que es el problema de la factorización que permite generar claves seguras con números primos aleatorios p, q, e. Por lo tanto para entender el algoritmo se definirán algunos conceptos teóricos fundamentales [16].

- **Anillo de los enteros**: se define como la existencia de un conjunto de números enteros, en donde se definen la relación de orden (mayor que) y las operaciones aritméticas suma y producto. Por ejemplo, los enteros definidos como $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, poseen la operación suma «+» en $\mathbb{Z}$ con elemento neutro 0 y por tanto $(\mathbb{Z}, +)$ es un grupo conmutativo. También la operación producto «.» en $\mathbb{Z}$ es conmutativa, asociativa y tiene elemento neutro 1, pero solo tienen elemento inverso los enteros -1 y 1. La operación . es distributiva con respecto a la +.

- **División Entera**: si $a, b \in \mathbb{Z}$ entonces la división de a entre b tiene un cociente q y un residuo r, si y solo si, $a = b.q + r$ con $q \in \mathbb{Z}$ y $0 \leqslant r \leqslant b - 1$.

- **Máximo común divisor mcd**: si $a, b, c \in \mathbb{Z}$ tal que $c|a$ y $c|b$, entonces c es el divisor común de a, b, además c es el máximo común divisor de a, b si c es el mayor de todos los divisores comunes de a, b denotado como $c = mcd(a, b)$. También se puede decir que a y b son primos relativos o primos entre sí, si se cumple que el $mcd(a, b) = 1$.

- **Números primos entre sí**: dos números $a, b \in \mathbb{Z}$ son primos entre sí, si no tiene ningún divisor en común excepto el 1, y si no existe p, q, r tales que $a = p.q$ y $b = p.r$ con $p \neq 1$.

- **Números primos**: un número $a \in \mathbb{Z}$ es primo si sus únicos divisores son el mismo y la unidad. Por lo tanto a se puede definir como $a = \lim_{x\to\infty}(\Pi(x)/(x/\ln(x))) = 1$, donde $\Pi(x)$ es la cantidad de primos p tales que $2 \leqslant p \leqslant x$. Además un número $a \in \mathbb{Z}$ se puede representar de forma única como $a = p_1^{k_1}.p_2^{k_2}\ldots p_n^{k_n} = \prod_{i=1}^{n} p_i^{k_i}$, donde p_i son los números primos estrictamente menores que a y k_i son exponentes naturales.

- **Algoritmo de Euclides**: permite calcular el mcd de dos números sin la necesidad de descomponerlo en factores primos, es decir, dado dos enteros positivos a, b, si se divide el mayor ente el menor, y el resultado da un cociente q y un resto r, que satisface que $a = bq + r$, entonces el $mcd(a, b) = mcd(b, r)$.

- **Aritmética modular**: dado un entero n positivo, se define la relación de congruencia modular de n con dos enteros a, b y se describe $(a)_n \equiv (b)_n$ si existe un entero tal que $(a - b) = k\ n$.

- **Inverso en** $\mathbb{Z}_n$: un elemento $[a]_n$ de $\mathbb{Z}_n$, es invertible si y solo si q y n son primos relativos, es decir que $mcd(q, n) = 1$.

- **Orden de un elemento** $\mathbb{Z}_n$: el orden de un elemento $a \in \mathbb{Z}_n$ se define como el mínimo número natural e no nulo tal que $a^e = (1)_n$.

- **Función multiplicativa de Euler** $\phi(r)$: es el número entero tal que $\phi(r) = \#\{S/S \geq 1\}$ con $S < r$, y $mcd(S, r) = 1$, siendo # la representación del cardinal del conjunto. Entonces dado r se puede se puede calcular $\phi(r)$ de la siguiente forma: si r es primo $\phi(r) = r - 1$, si r es el resultado del producto de primos $r = pq$ con $mcd(p, q) = 1$ entonces $\phi(r) = \phi(p).\phi(q)$, y se $r = p^k$ con p primo, entonces $\phi(r) = p^k - p^{k-1} = p^k(p - 1)$.

- **Proposición sobre clases de equivalencia con aritmética modular**: sea n un número natural que es el producto de dos números primos p, q, y sea t un número natural perteneciente a la clase $(1)_{\phi(n)}$, es decir donde se cumple que $(1)_{\phi(n)} = (t)_{\phi(n)}$, entonces para cualquier número x se tiene que $(x^t)_n = (x)_n$, entonces el módulo n de x^t y x pertenecen a la misma clase [77].

Con estas definiciones y propiedades matemáticas se tiene todo el marco conceptual para desarrollar el algoritmo de criptografía asimétrica RSA, que permite compartir información en forma segura entre Alice y Bob. La idea principal del algoritmo es que, tanto el emisor como el receptor puedan efectuar operaciones en forma independiente y se guarden sin que se conozca su contenido. Para la trasmisión se debe cumplir que si el canal de comunicación está interceptado por un tercero, éste no pueda entender el contenido de información compartida por el emisor y el receptor, los cuales no saben si el canal está siendo vulnerado. Para desarrollar el algoritmo RSA se utiliza un secuencia como la que se describe a continuación:

- El receptor Bob selecciona dos números p, q primos grandes y los multiplica para obtener $n = pq$.
- Bob en forma privada calcula el valor de la función multiplicativa de Euler $\phi(r) = \phi(pq) = \phi(p)\phi(q) = (p-1)(q-1)$ siendo p, q números primos y primos entre sí.
- Bob en forma privada selecciona un número e primo tal que $1 \leq e \leq \phi(n)$ y sea primo relativo con $\phi(n)$, luego calcula el inverso del módulo $\phi(n)$ que representa el coeficiente privado $d = (e^{-1})_{\phi(n)}$.
- Bob guarda en secreto los números (d, n) que forman la clave privada y comparte las números (e, n) que forman la clave pública.
- Si Alice quiere enviarle un mensaje x codificado a Bob, entonces cifra el mensaje con $Cif(x) = (x^e)_n$, puesto que conoce los números e, n que Bob compartió, y envía el número $Cif(x)$.
- El receptor Bob recibe el número $y = Cif(x)$, y realiza la operación de descifrado $Des(y) = (y^d)_n$, pues conoce el coeficiente privado de la clave d. Con la operación de descifrado Bob obtiene $Des(y) = ((x^e)^d)_n = (x)^(ed)_n = (x)_n$, puesto que d, e son inversos módulo $\phi(n)$ donde $d\ e$ es un número t que representa la clase $(1)_{\phi(n)}$, por lo que se cumple que $(x^t)_n = (x)_n$. Por lo tanto Bob puede conocer el mensaje x enviado por Alice [78, 79].

Por entender mejor el funcionamiento del algoritmo de criptografía asimétrica RSA, examinemos el siguiente ejemplo. Supongamos que $p = 11$ y $q = 23$ de donde $n = pq = 11 * 23 = 254$ y $\phi = (p-1)(q-1) = (11-1) * (23-1) = 220$, y como e no puede tener múltiplos comunes con ϕ, es decir que $1 < e < \phi$ donde se cumple que $mcd(\phi, e) = 1$. En particular vemos que si $e = 3$, el $mcd(220, 3) = 1$, entonces para calcular d que es el coeficiente privado de la clave se utiliza la aritmética modular $e\ d = 1 \mod \phi(n)$, siendo d el inverso multiplicativo de $e \mod \phi(n)$ y $d\ e - 1$ que es divisible exactamente por $\phi(n) = (p-1)(q-1)$. Este proceso se realiza con el algoritmo de Euler para obtener $d = inv(e, \phi) = inv(2, 220) = 147$. Entonces, la clave pública se representa con $(e, n) = (3, 253)$ y la privada con $(d, n) = (147, 253)$, por lo tanto para realizar el cifrado, primero se divide el texto en bloques que se numeran de 0 a n como se muestra en la tabla 2.10 [80, 81].

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z

Tabla. 2.10 Asignación de un número para cada letra del alfabeto: En el cuadro se muestra la asignación alfanumérica que permite cifrar un mensaje con el algoritmo RSA.

Para cifrar un mensaje como $M = QUBIT$, se escogen los números que le corresponden a cada carácter en la tabla 2.10, de donde $M = \{17, 21, 1, 8, 20\}$, y se utiliza $C = M^e \mod n$ para codificar cada M como se muestra en la parte derecha de la tabla 2.11. similarmente para descifrar $C =$

$\{106, 153, 1, 6, 157\}$ se escojen los números $(d, n) = (147, 253)$ y se utiliza $M = C^d$ mód n, de donde se obtiene nuevamente el mensaje $M = QUBIT$ tal como se muestra en la parte derecha de la tabla 2.11.

Cifrado con Llave Pública	Descifrado con Llave privada
$(e, n) = (3, 253)$	$(d, n) = (147, 253)$
$C = M^e(Mod(n))$	$C = M^d(Mod(n))$
$17^3 Mod(253) = 106$	$106^{147} Mod(253) = 17$
$21^3 Mod(253) = 153$	$153^{147} Mod(253) = 21$
$1^3 Mod(253) = 1$	$1^{147} Mod(253) = 1$
$8^3 Mod(253) = 6$	$6^{147} Mod(253) = 8$
$20^3 Mod(253) = 157$	$157^{147} Mod(253) = 20$

Tabla. 2.11 Ejemplo de cifrado y descifrado RSA: En la tabla se muestran los procesos para cifrado y descifrado de la palabra $QUBIT$ usando el algoritmo RSA.

2.5.3. Curvas Elípticas ECC

Se pude definir una curva elíptica sobre un cuerpo $\mathbb{K}$ como una curva sin puntos singulares representada mediante la ecuación general de Weierstrass [82, 83]:

$$y^2 + a_1xy + a_3x + b = x^3 + a_2x^2 + a_4x + a_6,\ a_i \in \mathbb{K}, \tag{2.29}$$

y realizando transformaciones lineales se llega a la ecuación reducidas de Weierstrass.

$$y^2 = x^3 + ax + b : a, b\epsilon\mathbb{K}, \tag{2.30}$$

también es posible definir una curva elíptica sobre un cuerpo finito $\mathbb{F}_q$, y si $\mathbb{K} = \mathbb{F}_q$, con $q = p^m$ y p primo, se genera un grupo finito con cardinal $\#E(\mathbb{F}_q) = q + 1 - t$ donde t es la traza del endomorfismo de Frobenius $\varphi : E(\mathbb{F}_q) \longrightarrow E(\mathbb{F}_q)$, el cual le asigna a cada (x, y) el punto (x^q, y^q). Lo anterior, permite realizar la criptografía, a partir de la complejidad del problema de los logaritmos discretos, como se expone a continuación.

- **Teorema de Hasse**: Si $E/\mathbb{F}_q$ es una curva elíptica sobre un campo finito, entonces $\#E(\mathbb{F}_q) = q + 1 - t$, donde $t = \mathrm{Tr}\{\pi_E\}$ representa la traza del endomorfismo de Frobenius para π_E y $|t| \leq 2\sqrt{q}$ [84].
- **Problema del logaritmo discreto**: Si se tiene un grupo finito y cíclico G, con un generador g de G y un elemento h de G, entonces se puede encontrar el entero x tal que $h = g^x$ que es la representación del logaritmo discreto de h en la base g [85].

Del problema del logaritmo discreto se puede observar que si se conoce g y x, se puede calcular de forma eficiente $h = g^x$, pero por el contrario si se tiene g y h, encontrar x se convierte en un problema computacionalmente difícil de resolver. Similarmente del problema del logaritmo discreto se puede calcular $Q = d.P$, con solo conocer P y d, pero en caso contrario si se conoce P y Q y se quiere calcular d, ya no es fácil, se tiene un problema computacionalmente difícil de resolver.

La criptografía de curvas elípticas (Elliptic curve cryptography), fue desarrollada en 1985 por Neal Koblitz y Victor Miller, y son utilizadas sobre cuerpos finitos para codificar información [86, 87]. Los principales algoritmos con curvas elípticas son: El algoritmo ECIES, que utiliza la función KDF como derivación de claves construida a partir de hash, la función ENC (DEC) que cifra y descifra en un sistema de clave compartida como el AES, y la función MAC que es un código de autenticación [88]. El algoritmo DSA es la base del estándar de firma digital DSS, que utiliza funciones hash que le asignan a cada mensaje una representación numérica de tamaño fijo resistente a colisiones, es decir, que la firma digital es un hash del mensaje y de la clave privada, de tal manera que, cualquier entidad podrá comprobar la veracidad de la firma a partir de la clave pública [89]. Uno de los primeros modelos criptográficos con estas ideas de curvas elípticas es el algoritmo de ElGamal que utiliza el problema de los logaritmos discretos para generar un p primo perteneciente al cuerpo $\mathbb{F}_p$ y un elemento $g \in \mathbb{F}_p^*$ con orden q primo con la misma longitud que p operados en el subgrupo G de orden q generado por g, y la clave privada d perteneciente a $[1, q-1]$ que se hace pública en el proceso $h = g^d$ perteneciente a G. El mensaje a cifrar se representa con m tal que $0 \leq m \leq p$, por lo tanto para cifrar un mensaje se tiene una entrada representada por los parámetros (p, q, g), una clave pública h, y un mensaje m, que se operan en una secuencia como la siguiente [90]:

- Se escoge un entero r aleatorio perteneciente a $[1, p-1]$.
- Se calcula $C_1 = g^r$ y $C_2 = m * h^r$.
- Se obtiene el resultado (C_1, C_2).

Para el proceso de descifrado se tiene una entrada representada por (p, q, g), una clave privada d y un mensaje cifrado (C_1, C_2), que se operan en de la siguiente manera para obtener nuevamente el mensaje m [91]:

- Se calcula $m = C_2 * (C_1^d)^1$
- Se obtiene m.

Por otra parte, el algoritmo de ElGamal se puede construir a partir del grupo de puntos de una curva elíptica que pueden generar un p primo para definir el cuerpo $\mathbb{F}_p$ donde a, b son parámetros de la curva $E_{a,b}$ sobre $\mathbb{F}_p$. Con esta caracterización se puede escoger un punto P de la curva, de tal manera que exista un n primo y del mismo orden que p, por lo tanto las operaciones se realizan en el grupo $E_{a,b}(\mathbb{F}_p)$ de orden n generado por p. Para este modelo la clave privada se define como un entero d perteneciente a $[1, n-1]$, la clave pública está representada por el punto $Q = d * P$ de la curva, y el mensaje m con $0 \leq m \leq p$ estará dado por M que es la abscisa en la curva. Para cifrar un mensaje se tiene una entrada representada por (p, a, b, P, n), una clave pública Q y un mensaje m que se operan de en la siguiente secuencia lógica.

- Se representa m con M de $E_{a,b}\mathbb{F}_p$.
- Se escogé un entero aleatorio r perteneciente a $[1, n-1]$.
- Se calcula $C_1 = r * P$ y $C_2 = M + r * Q$ para obtener el mensaje cifrado.

Para el proceso de descifrado se tiene una entrada (p, a, b, P, n), una clave privada d y un mensaje cifrado C_1, C_2 que se operan para obtener nuevamente el mensaje original [90].

- Se calcula $M = C_2 - d * C_1$.
- Se obtiene m a partir de M.

2.6. Ataques a la Criptografía Asimétrica

En esta sección se discutirá la dificultad computacional para encontrar los factores de un número entero compuesto n, construido a partir del producto de dos números primos grandes con una longitud mínima de 512 bits, o 155 dígitos, por lo que n contiene 310 dígitos que corresponden a 1024 bits de longitud, condición necesaria para tener una seguridad promedio en las transferencias de información con cifrado asimétrico. El algoritmo RSA se puede vulnerar con una baja probabilidad de éxito con métodos de factorización entera como [92]: 1) Directo o de Eratóstenes, 2) de Euler [93], 3) de Fermat [94], 4) de Dixon, 5) de Williams $p+1$, 6) de Pollar ρ, 7) de Pollar $p-1$, 8) de fracciones continuas, 9) de curvas elípticas y 10) finalmente por el método de cifrado numérico.

2.6.1. Ataque por Cifrado Cíclico

El ataque por cifrado cíclico es una técnica que utiliza operaciones matemáticas de la forma $C = N^e \mod n$, donde N corresponde a un valor secreto. Entonces, el ataque realiza múltiples operaciones en el cifrado C_i con la llave pública e, si se llega a conocer C, entonces se encontró el secreto N ya que RSA es un grupo multiplicativo, y después de obtener el criptograma C se realiza la operación $C_i = N_{i-1}^e \mod n$ para $i = 1, 2, \ldots$ con $C_0 = C$. Por lo tanto, si la i-ésima cifra se encuentra en el criptograma C entonces el cifrado $i-1$ corresponde al secreto [72, 95].

2.6.2. Ataque por Paradoja del Cumpleañero

Los ataques por paradoja del cumpleañero se fundamentan en la siguiente idea. Supóngase que en un salón se encuentran marcados los 365 días de un año e ingresa un grupo de persona de uno en uno. El que entra marca el día de su cumpleaños. Se pregunta: £cuántas personas deben entrar para que exista una probabilidad mayor al 50 % de que se encuentre una día repetido?. La respuesta es que con entrar 23 personas ya existe una probabilidad mayor del 50 % para que el siguiente en entrar encuentre el día de su cumpleaños ocupado. Esto se entiende de la siguiente forma: cuando la primera persona entra encuentra 365 posibilidades para para marcar su día de cumpleaños, que representa una probabilidad de $n/n = 1$. El segundo en entrar tiene una probabilidad de $(n-1)/n$, la tercera de $(n-2)/n$, por lo tanto la probabilidad de no encontrar su día de cumpleaños ocupado es de $p_{NC} = n!/(n^k(n-k)!)$ donde k es el número de personas que han entrado antes. Por lo tanto, si $k = 23$, entonces $p_{nc} = 0{,}493$ y la probabilidad de que dos personas coincidan en la misma casilla es $p_c = (1 - p_{NC}) = 1 - 0{,}493 = 0{,}507$ que es mayor que el 50 %.

Aplicando la paradoja del cumpleañero se puede atacar el algoritmo de cifrado RSA, ya que se puede encontrar d partiendo de una pareja d' construida con los valores públicos de la víctima. Entonces, si se conoce el módulo n y la clave e, el ataque toma un N cualquiera y se seleccionan dos números aleatorios $(i, j) \in n$ para realizar la operaciones $N^i \mod n$ y $N^j \mod n$. Si el resultado es diferente entonces se incrementa en una unidad los números i y j y se realizan de nuevo las operaciones: $N^{i+1} \mod n$ y $N^{j+1} \mod n$, que se repiten sucesivamente hasta obtener una coincidencia donde $N^i \mod n = N^j \mod n$, de donde se obtiene d [95].

2.6.3. Ataque por Fuerza Bruta

El ataque de fuerza bruta, es una técnica que prueba una serie de contraseñas almacenadas en listas llamadas diccionarios o directamente construida con palabras claves. El objetivo del ataque es obtener la clave de ingreso al sistema sin tener una autorización, es decir, que un ataque por fuerza bruta es una suposición aleatoria de encontrar una respuesta probando combinaciones sucesivas hasta

encontrar la verdadera. Por ejemplo para adivinar una clave de 4 caracteres construida a partir de una alfabeto de 27 caracteres, se tiene una probabilidad de éxito aproximada de $P = (1/27)^4 = 0{,}000002$, y dependiendo de la cantidad de palabras baldías en el diccionario la probabilidad puede disminuir. Este ataque es lento, porque a medida que la longitud de la clave crece el arreglo de permutación $A_m^n = m!/(m-n)!$ también lo hace. Por ejemplo, si $n = 4$ la longitud de la clave tiene $A_{27}^4 = 27!/(27-4)! = 421200$ posibilidades, y si $n = 8$ crece entonces a $A_{27}^8 = 27!/(27-8)! = 89513424000$, lo que quiere decir que a medida que la longitud de la clave crece es menos probable que el ataque por fuerza bruta funcione [96].

2.6.4. Ataque MITM: man-in-the-middle

El ataque man-in-the-middle MITM, de forma similar que el ataque de fuerza bruta, se puede construir para cualquier sistema criptográfico. Esta técnica permite que un atacante capture todo el tráfico entre las víctimas usando conexiones independientes e inyecte una nueva información. Así, el atacante le hace creer al transmisor y al receptor de la comunicación que están hablando a través de una conexión privada, cuando éste es el que controla todo el flujo de información [97, 98]. Por ejemplo, el ataque MITM es la amenaza más crítica para el estándar WPA2-Enterprise, que es el encargado de mantener las comunicaciones inalámbricas protegidas, por la robustez de los algoritmos criptográficos y los certificados de seguridad que utiliza. Para realizar una incursión a las redes inalámbricas protegidas con WPA2-Enterprise, se pueden crear puntos de Wi-Fi falsos utilizando un servidor RADIUS para obtener el inicio de sesión, luego se hace una solicitud al servidor y se obtiene una respuesta que utiliza MS-CHAPv2. Esta información obtenida es suficiente para aumentar la fuerza bruta de la contraseña [97, 99].

2.7. Funciones Hash

Las funciones hash, matemáticamente son funciones que no tienen inversa, y pueden mapear datos de un tamaño arbitrario en datos de un tamaño fijo, es decir, que las operaciones matemáticas realizadas con las funciones hash tienen un carácter de irreversibilidad funcional. Por ejemplo, una función hash transforma una secuencia binaria $M = 100101111000101\ldots01010$, en una secuencia binaria $C = g(M) = 0100100\ldots110$ con una longitud menor que M, lo que permite la verificación de integridad [100, 101], la generación de claves de seguridad [102], la transmisión de certificados, firmas digitales [103], y la construcción de las cadenas bloques [104, 105] encargadas de mantener la seguridad en la transacciones electrónicas con criptomonedas. Las funciones hash criptografía poseen propiedades matemáticas como las siguientes:

- De $g(M)$ no se permite conocer M.
- La longitud de $g(M)$ es menor que la de M.
- La capacidad de cómputo para calcular $g(M)$ es mínima.
- Lasa funciones hash son unidireccionales, es decir que dado un W en la imagen de g, es extremadamente difícil encontrar un mensaje M tal que $g(M) = W$, Y dada un M y $g(M)$, es extremadamente difícil encontrar un mensaje $M' \neq M$ tal que $g(M') = g(M)$ [106, 107]. En la figura 2.10 se muestra como se realiza la verificación de integridad utilizando una función hash.

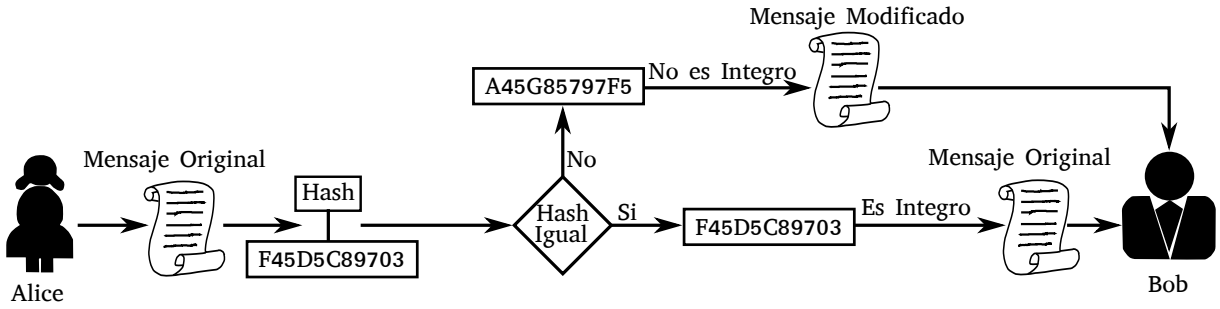

Figura. 2.10 Estructura de una función hash criptográfica: En la figura se muestra la verificación de integridad en un documento. Alice obtiene un hash de un documento, lo comparte con Bob que nuevamente le saca el hash con la misma función, y si el hash de Alice y el de Bob coinciden entonces el documento conserva la integridad, si los hash son diferentes el documento fue modificado.

2.7.1. Algoritmo MD5

MD5 es un algoritmo resumen utilizado en sistemas criptográficos para comprobar integridad, fue desarrollado por RSA Data Security, y no permite el cifrado, ya que destruye completamente la información en el proceso de codificación. Esta función hash divide el mensaje en bloque de 512 bits y genera una salida resumen de 128 bits, tiene un búfer estándar de 128 bits conformado por 4 palabras de 32 bits, y una función de compresión que opera en cuatro rondas y en cada ronda el bloque de mensaje y el búfer son combinados en el cálculo, mediante el uso de sumas modulares, XOR, AND, OR y operaciones de rotaciones sobre palabras de 32 bits. Después que cada bloque de 512 bits se combina con el búfer estado 4 veces, el búfer estado y el resultado son operados en una sumas módulo 232 para obtener la salida [108, 109].

2.7.2. Algoritmo SHA-1

Similarmente que MD5 la función hash SHA-1(Secure Hash Algorithm) es un algoritmo resumen que genera una salida de 20 bytes, a partir de un mensaje con un tamaño máximo de 2^{64} bits. Este algoritmo criptográfico fue desarrollado por el NIST en 1995, y su funcionamiento se estructura a partir de bloques de 512 bits operados en 80 rondas de procesamiento con un vector de inicialización conformado por 5 palabras de 32 bits, y textos de 160 bits que generan una salida resumen de 160 bits con una complejidad algorítmica de 2^{80} combinaciones estructuradas en la siguiente secuencia.

- Si el documento tiene una longitud menor de 512 bits, se adicionan bits 0 hasta completar el tamaño del bloque.
- Se inicializa con el vector formado por las 5 palabras en el sistema de numérico hexadecimal, utilizando octetos de orden superior A, B, C, D, E de la siguiente manera $A16 = 67452301,\ B16 = EFCDAB89,\ C16 = 98BADCFE,\ D16 = 10325476$, y $E16 = C3D2E1F0$.
- Para obtener el resumen se utilizan las funciones F, G, H, I, las cuales son una secuencia lógica que va de 0 a 79 representada con las funciones $F(t)(b, c, d) = b \wedge c \vee \neg tb \wedge d$ para $t = 0 \ldots 19$, $F(t)(b, c, d) = b \veebar c \veebar d$ para $t = 20 \ldots 39$; $F(t)(b, c, d) = b \wedge c \vee b \wedge d \vee c \wedge d$ para $t = 40 \ldots 59$, donde t representa el número de vueltas.
- Finalmente se calcula $F(t)(b, c, d) = b \veebar c \veebar d$, para $t = 60 \ldots 79$. Cada una de las funciones opera con tres palabras de 32 bits que dan una salida de 32 bits, como se muestra en la figura 2.11 [110].

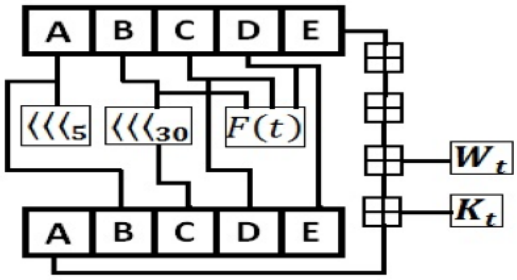

Figura. 2.11 Algoritmo SHA-1: En la figura A, B, C, D, E representan las palabras con una longitud de 32 bits, F es una función no lineal que varía con una rotación del bit a izquierda por n lugares, n también varía para cada operación, W_t representa el mensaje expandido de la ronda t, K_t es la constante de rotación de la ronda t que denota el módulo de adición 2^{32}. Figura tomada de [110].

2.7.3. Algoritmo SHA-2

El algoritmo SHA-2 está compuesto por un conjunto de funciones hash: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 y SHA-512/256. El algoritmo fue desarrollado por la NSA (National Security Agency) y publicado por el NIST como FIPS (Federal Information Processing Standard). Las funciones SHA-224 y SHA-256 cifran utilizando 64 iteraciones, donde se mezclan las operaciones $(\oplus, \wedge, \vee, \ulcorner)$ que son las representaciones de las compuertas lógicas XOR, AND, OR, NOT, con textos máximos de $(2^{64} - 1)$ bits, y están formadas por palabras de 32 bits agrupadas en bloques de 512 bits que generan una salida de 224 o 256 bits. Similarmente las funciones SHA-384, SHA-512, SHA-512/224 y SHA-512/256 cifran utilizando 80 iteraciones mediante las mismas operaciones anteriores, con textos máximos de $2^{128} - 1$ bits formados por palabras de 64 bits agrupadas en bloques de 1024 bits con salidas de $384, 512, 224, 256$ bits, en la figura 2.12 se muestra una iteración de SHA-2 [111].

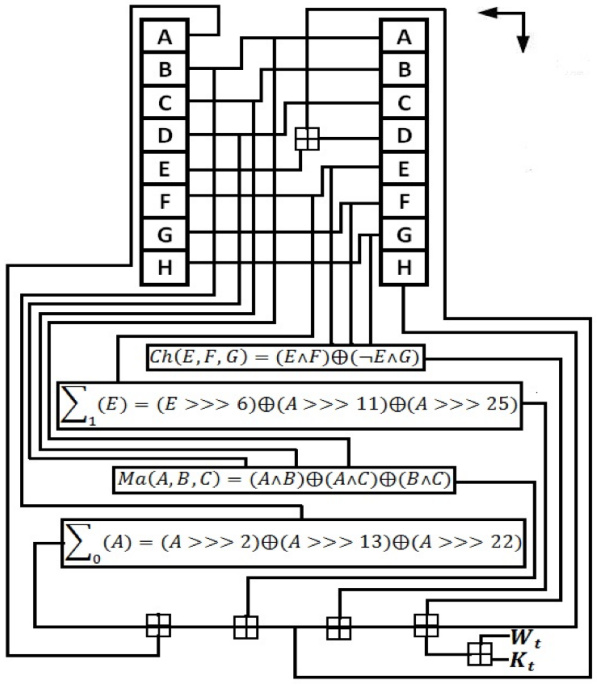

Figura. 2.12 Algoritmo SHA-2: En la figura se muestra el funcionamiento de las operaciones $Ch, \sum_1, Ma_j, \sum_0 0$, en una ronda de cifrado. Figura modificada del trabajo [111].

2.7.4. Algoritmo SHA-3

Este algoritmo fue desarrollado por Guido Bertoni, Joan Daemen, Michaël Peeters y Gilles Van Assche y fue publicado por NIST en 2015. Su funcionamiento se basa en permutaciones aleatorias $KECCAK - f$, que están compuestas por secuencias de cinco transformaciones con un rango de b bits, donde las operaciones de índice son de mód 5 para las dos primeras dimensiones y módulo mód w para la tercera. El bloque de permutación utiliza una $XOR(\veebar)$ con palabras formadas por potencias de dos: $w = 2^l$ bits. Por ejemplo, para palabras de 64 bits $l = 6$. Para SHA-3 el estado de permutación se representa con una matriz de $5 \times w$ bits, con $[i], [j], [k]$ el bit de la entrada definido por $5i + j \times w + k$. Además, las funciones de permutación están conformadas por $12 + 2l$ rondas mezcladas en cinco fases. En la figura 2.13 se muestra el funcionamiento de una ronda del algoritmo SHA-3 y a continuación se describen sus principales fases [112, 113].

- Se calcula la paridad de las columnas $5w$ que son de 5 bits. En general la paridad se calcula de la forma $a[i][j][k] \longleftarrow a[i][j][k] \oplus paridad(a[0, \ldots, 4][j-1][k]) \oplus paridad(a[0 \ldots 4][j+1][k-1])$.
- Fase ρ. Se giran bit a bit las 25 palabras en una dirección diferente a la secuencia $0, 1, 3, 6, 10, 13 \ldots$. Es decir, no se rota en las combinaciones $a[0][0]$ y para todo $0 \leq t \leq 24$, $a[i][j][k] \longleftarrow a[i][j][k-(t+1)(t+2)/2]$ donde $\binom{i}{j} = \left(\begin{smallmatrix} 3 & 2 \\ 1 & 0 \end{smallmatrix}\right)^t \binom{0}{1}$.
- Fase π. Se permutan las 25 palabras utilizando un patrón fijo definido por $a[j][2i + 3j][k] \longleftarrow a[i][j]$.
- Fase $\aleph$. Se mezclan las filas bit a bit con la siguiente operación $x \longleftarrow \oplus(\neg y \wedge z)$ y en forma general se utiliza $a[i][j][k] \longleftarrow a[i][j] \oplus (\neg a[i][j+1][k] \wedge a[i][j+2][k])$, la cual es una operación no lineal.
- Fase l, se aplica en la ronda n definida con $0 \leq m \leq l$, $a[0][0][2^m - 1]$ una XOR con el bit $m + 7n$ de una secuencia LFSR de grado 8, para romper la simetría de las anteriores fases.

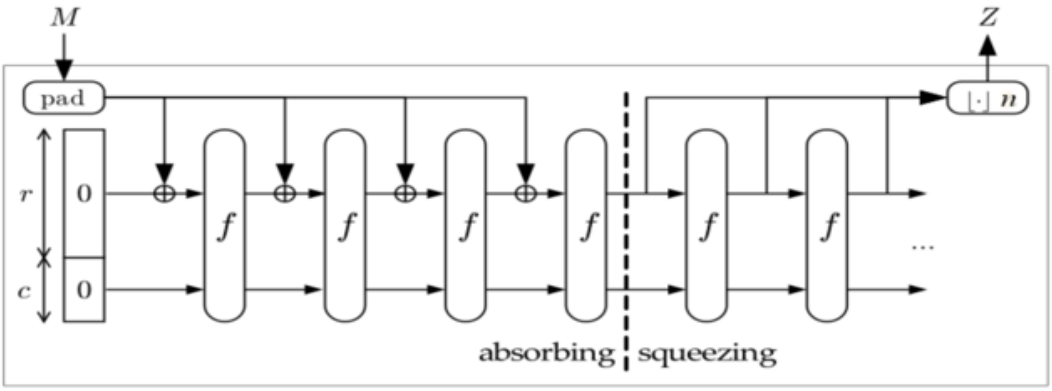

Figura. 2.13 Algoritmo SHA-3: En la figura se muestra la construcción de SHA-3 en forma de esponja. Los datos absorbidos son procesados para mostrar una salida de longitud deseada. En la fase de absorción, se utiliza una operación XOR para procesar los bloques, que luego se transforman con la función de permutación f.SHA-3. Figura tomada del estándar de la NIST [112].

2.8. Ataques a las Funciones Hash

Por las características matemáticas de las funciones hash, los ataques a estas funciones son difíciles de realizar, sin embargo, existen estrategias informáticas que ayudan a romper su seguridad. Es

posible atacar la no aceptación de colisiones y la unidireccionalidad de las funciones hash, estos ataques se pueden realizar en forma general si con la misma técnica se vulneran diferentes funciones hash o en forma particular si el ataque va dirigido a una función en particular. Por ejemplo, los ataques más generalmente realizados a las funciones hash, son los que utilizan las técnicas de fuerza bruta integrados con la paradoja del cumpleañero, en donde se busca generar colisiones para que un par de entradas tengan la misma salida, y de esta manera se pueda obtener la información. Además, se pueden desarrollar ataques con criptoanálisis diferencial, crioptoanálisis integral, y por canal auxiliar, técnicas que se explicaron en las secciones anteriores.

2.8.1. Ataque Chino

El ataque chino fue desarrollado por un grupo de matemáticos que logro romper la seguridad del algoritmo resumen MD5 y SHA-1. Es una modificación sofisticada del ataque por fuerza bruta en donde se eligen de manera aleatoria dos textos M y M' que tengan el mismo hash, lo que permite encontrar colisiones en el hash más rápido. Con esta técnica se ha demostrado que se puede romper SHA-1 con menos de 2^{69} operaciones, y con una velocidad de 2000 veces más rápida que un ataque de fuerza bruta que necesita para ser efectivo 2^{80} operaciones. El ataque más afectivo a SHA-1 se realizó con 2^{63} operaciones, que está dentro de los límites de la capacidad de cálculo computacional actual [114].

2.8.2. Ataque Multicolisión

Una colisión se produce cuando a partir de dos mensajes diferentes operados con la misma función hash, se obtiene la misma salida, el mismo resumen. Por lo tanto una función hash es susceptible de ataques por coalición, si dado un x es computacionalmente viable encontrar un $y \neq x$ tal que $g(x) = g(y)$, por el contrario, si computacionalmente es difícil encontrar un par (x, y) tal que $g(x) = g(y)$, entonces la función hash es segura a ataques por coalición. [115]

2.8.3. Ataque Multicolisión de Joux

Joux demostró que se pueden realizar ataques por colisión a las funciones hash interactivas clásicas con una complejidad computacional de orden $\mathcal{O}\left(r^{2^{n/2}}\right)$ que es mucho menor que $\Omega(2^{n(n^r-1)/2r})$ orden computacional del modelo de oráculo aleatorio [115]. Joux también demostró que H^f no es una función aleatoria y se puede atacar controlando las primeras colisiones sucesivas $f(h_{i-1}, m_i^1) = f(h_{i-1}, m_i^2) = h_i$ con $1 \leq i \leq r$, entonces $H(m_1^{i_1} \| \ldots \| m_r^{i_r}) = h_r$ para $i_1, \ldots, i_r \in \{1, 2\}$, donde se tienen colisiones de 2^r vías, con una complejidad computacional de orden $\mathcal{O}\left(r^{2^{n/2}}\right)$ [116].

En general todos los sistemas criptográficos clásicos pueden ser vulnerables con la potencia de la computación cuántica, la cual ha demostrado que se pueden realizar búsquedas computacionalmente eficientes en listas desordenadas o descomponer cualquier número en sus factores primos con los algoritmos de Grover y Shor. Es decir que la computación cuántica puede vulnerar la criptografía clásica, que es la encargada en la actualidad de codificar la información para realizar transmisiones seguras. En el siguiente capítulo desarrollaremos los elementos mínimos de la teoría cuántica para tener un marco conceptual completo de la teoría de la información y computación cuántica. En el capítulo 4 se estudiará la criptografía cuatica y se examinarán los algoritmos de Grover y Shor [117, 118, 119].

CAPÍTULO 3

MECÁNICA CUÁNTICA: INFORMACIÓN Y COMPUTACIÓN CUÁNTICA. PROPIEDADES, PRINCIPIOS Y DEFINICIONES

La mecánica cuántica ha cambiado la forma de entender el entorno en que vivimos, esta teoría física predice fenómenos naturales que no hubiesen sido esperados en las teorías clásicas, y en general, se escapa a la intuición de cómo debería ser la naturaleza. Por lo tanto, en este capítulo resumiremos algunos conceptos fundamentales de la mecánica cuántica necesarios para la comprensión de la criptografía cuántica. En particular, se realizará una breve introducción a los postulados, propiedades y definiciones a partir de un enfoque de la teoría de la información y la computación cuántica. Se realizará la definición de qubit y la representación de algunas compuertas lógicas cuánticas. Se revisarán algunos modelos físicos de computadora cuántica, y de canal cuántico de comunicación ruidoso, lo que permitirá tener una nueva perspectiva sobre el procesamiento, seguridad, y almacenamiento de la información a nivel cuántico.

3.1. Teoría de la Información Cuántica

La teoría de la información cuántica se fundamenta en las ideas clásicas de información, las cuales definen un conjunto de leyes matemáticas que modelan el procesamiento, la transmisión y la representación de la información. Estas ideas propuestas por Claude E. Shannon en su trabajo titulado *La teoría matemática de la comunicación*, modelan todo proceso de comunicación como un sistema compuesto por un emisor, un transmisor y un canal ruidoso a través del cual se transmiten los mensajes. Esta idea es retomada en la teoría cuántica de la información, ya que al igual que la información clásica, ésta puede procesarse, transmitirse de un lugar a otro, manipularse y analizarse con algoritmos computacionales. Adicionalmente, la teoría de la información cuántica necesita de un sistema físico controlado por las propiedades de la mecánica cuántica como medio de comunicación y procesamiento. A continuación expondremos los principales postulados de la teoría cuántica escritos desde la perspectiva de la teoría de la información y computación cuántica. [120, 121, 122].

- ***Postulado 1:*** *Asociado a cualquier sistema físico aislado hay un espacio vectorial complejo con un producto interno (es decir, un espacio de Hilbert) conocido como el espacio de estados cuántico $\mathcal{H}$ del sistema. El sistema se describe completamente por su vector de estado, el cual*

es un vector unitario en el espacio de estados del sistema. Por ejemplo, el sistema mecánico cuántico más simple, el qubit, tiene un espacio de estados bidimensional. Si $|0\rangle$ y $|1\rangle$ forman una base ortonormal para el espacio de estados del qubit, entonces se puede escribir cualquier vector de estado arbitrario en el espacio de estados del qubit como $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$.

- ***Postulado 2:*** *La evolución de un sistema cuántico cerrado se describe mediante una transformación unitaria. Es decir, el estado $|\psi\rangle$ del sistema en el tiempo t_1 está relacionado con el estado $|\psi'\rangle$ del sistema en el tiempo $t_2 > t_1$ por un operador unitario $\hat{U}$ que depende solo de la diferencia entre los tiempos $t_2 - t_1$. En otras palabras, la mecánica cuántica simplemente asegura que la evolución de cualquier sistema cuántico cerrado se puede describir como $|\psi'\rangle = \hat{U}(t_2 - t_1)\,|\psi\rangle$.*

- ***Postulado 2′:*** *La evolución temporal del estado de un sistema cuántico cerrado se describe mediante la ecuación de Schrödinger, $i\hbar\,\mathrm{d}\,|\psi\rangle/\mathrm{d}t = \hat{H}\,|\psi\rangle$,* donde $\hat{H}$ es un operador hermítico conocido como el hamiltoniano del sistema cerrado (el operador de energía), y $\hbar$, es la constante física con unidades de acción conocida como constante de Planck y cuyo valor debe determinarse experimentalmente[1]. Es usual usar las llamadas unidades naturales donde $\hbar = 1$.

- ***Postulado 3:*** *Las mediciones cuánticas se describen mediante una colección $\{\hat{M}_m\}$ de operadores de medición. Estos son operadores actúan sobre el espacio de estados del sistema que se está midiendo. El índice m se refiere a los posibles resultados de la medición que pueden ocurrir en el experimento. Si el estado del sistema cuántico es $|\psi\rangle$ inmediatamente antes de la medición, entonces la probabilidad de que se produzca el resultado m es $p(m) = \langle\psi|\,\hat{M}_m^\dagger\hat{M}_m\,|\psi\rangle$, y el estado del sistema después de la medición es $|\psi'\rangle = \hat{M}_m\,|\psi\rangle\,/\sqrt{p(m)}$. Los operadores de medición deben satisfacer la relación de completitud $\sum_m \hat{M}_m^\dagger\hat{M}_m = I$, de tal forma que: $1 = \sum_m p(m) = \sum_m \langle\psi|\,\hat{M}_m^\dagger\hat{M}_m|\psi\rangle = \langle\psi|\psi\rangle$.*

- ***Postulado 4:*** *El espacio de estados de un sistema físico compuesto es el producto tensorial de los espacios de estados de los sistemas físicos de componentes. Además, si tenemos sistemas numerados del 1 al n, y el sistema i-ésimo está preparado en el estado $|\psi_i\rangle$, entonces el estado conjunto del sistema total es $|\psi_1\rangle \bigotimes |\psi_2\rangle \bigotimes ... \bigotimes |\psi_n\rangle$.*

3.2. Operadores en Mecánica Cuántica

Se puede decir que $\widehat{A}$ es un operador en mecánica cuántica si realiza una transformación de un estado del sistema $|\psi\rangle$ a otro estado $|\phi\rangle$ de tal manera que $\widehat{A}\,|\psi\rangle = |\phi\rangle$. En forma general, los operadores se pueden definir como una función de operadores $f(\widehat{A})$:

$$f(\widehat{A}) = \sum_{i=0}^{\infty} a_n \widehat{A}^n, \tag{3.1}$$

siendo a_n el coeficiente de expansión que corresponde a la serie de potencias de la función real $f(x)$. Un operador lineal entre espacios vectoriales V y W se puede definir como un mapa $\widehat{A} : V \longrightarrow W$ tal que:

$$\widehat{A}\left(\sum_i a_i\,|\psi\rangle\right) = \sum_i a_n \widehat{A}(|\psi\rangle), \tag{3.2}$$

[1] En el SI de unidades $\hbar = 1{,}054 \times 10^{-34}\ J\,s$.

y si V, W y X son espacios vectoriales tales que $\widehat{A} : V \longrightarrow W$, y $\widehat{B} : V \longrightarrow X$ son operaciones lineales, entonces se puede definir la composición de operadores $\widehat{A}$ con $\widehat{B}$ como $\widehat{A}\widehat{B}$.

La relación de completez en el espacio vectorial V se puede escribir como:

$$\widehat{I} = \sum_j |j\rangle \langle j| , \tag{3.3}$$

siendo $\widehat{I}$ el operador identidad, tal que $\widehat{I}|\psi\rangle = |\psi\rangle$ y con $\{|j\rangle\}$ una base ortonormal del espacio vectorial V. De la anterior relación, podemos escribir:

$$|\psi\rangle = \left(\sum_j |j\rangle \langle j| \right) |\psi\rangle = \sum_j \langle j|\psi\rangle \ |j\rangle . \tag{3.4}$$

Los operadores también se pueden representar de forma matricial usando la base del espacio vectorial. Es decir que dado un operador $\widehat{A}$ y una base $\{|j\rangle\}$, se le puede asociar una matriz cuadrada de números A_{jk} tales que:

$$A_{jk} = \langle j| \widehat{A} |k\rangle , \tag{3.5}$$

y como la base $\{|j\rangle\}$ ortonormal se tiene que:

$$\widehat{A}|k\rangle = \widehat{I}|k\rangle = \left(\sum_j |j\rangle \langle j| \right) \widehat{A}|k\rangle = \sum \langle j| \widehat{A} |k\rangle |j\rangle , \tag{3.6}$$

en donde se usó la relación de completez. Lo anterior implica que $A_{jk} = \langle j| \widehat{A} |k\rangle$, como se mostró en la ecuación 3.5. La traza $\mathrm{Tr}\left(\widehat{A}\right)$ de un operador se define como la suma de sus elementos diagonales:

$$\mathrm{Tr}\left(\widehat{A}\right) = \sum_j \langle j| \widehat{A} |j\rangle = \sum \widehat{A}_{jj}. \tag{3.7}$$

Adicionalmente se tienen las siguientes definiciones y propiedades sobre operadores $\widehat{A}$:

- $\widehat{A}$ es un operador lineal si $\widehat{A}(\alpha_1|\psi_1\rangle + \alpha_2|\psi_2\rangle) = \alpha_1\widehat{A}|\psi_1\rangle + \alpha_2\widehat{A}|\psi_2\rangle$.
- $\widehat{A}$ es el operador identidad si $\widehat{A}|\psi\rangle = |\psi\rangle$ para todo vector $\psi \in \mathcal{H}$.
- $\widehat{0}$ es el operador nulo si $\widehat{0}|\psi\rangle = |0\rangle \in \mathcal{H}$ para todo $|\psi\rangle \in \mathcal{H}$, donde $|0\rangle$ es el vector nulo del espacio.

Además se pueden realizar las siguientes operaciones algebraicas y definicione utilizando las propiedades de los operadores cuánticos:

- Conmutador: $[\widehat{A}, \widehat{B}] = \widehat{A}\widehat{B} - \widehat{B}\widehat{A}$. Si $[\widehat{A}, \widehat{B}] = 0$, quiere decir que $\widehat{A}$ y $\widehat{B}$ conmutan.
- Potencia: para todo m entero positivo $\widehat{A}^m = \widehat{A}\widehat{A}\widehat{A}...\widehat{A}$, m-veces.
- Norma de un operador: La norma de un operador $\widehat{A}$ se escribe como $\|\widehat{A}\|$. Si se cumple la siguiente desigualdad $\sqrt{\langle\psi|\widehat{A}^\dagger\widehat{A}|\psi\rangle} \leq C\sqrt{\langle\psi|\psi\rangle}$, para todo estado $|\psi\rangle \in \mathcal{H}$, entonces la norma del operador será $\|\widehat{A}\| = C$.

- Operador adjunto: El operador adjunto $\widehat{A}^\dagger$ de un operador $\widehat{A}$ se define mediante la relación $\langle\phi|\widehat{A}|\psi\rangle = \langle\psi|\widehat{A}^\dagger|\phi\rangle$, con $|\psi\rangle, |\phi\rangle \in \mathcal{H}$. Adicionalmente se cumple que $(\widehat{A}\widehat{B}\widehat{C})^\dagger = \widehat{C}^\dagger\widehat{B}^\dagger\widehat{A}^\dagger$.

- Operador hermítico: Un operador es hermítico si $\widehat{A} = \widehat{A}^\dagger$, lo que significa que el operador es autoadjunto. Los operadores hermíticos tienen gran importancia en la teoría cuántica porque con ellos se describen los observables del sistema, es decir las variables dinámicas como: energía, momentum, momentum lineal, espín, etc. Los operadores hermíticos satisfacen el importante teorema espectral el cual permite afirmar que, para cualquier operador hermítico se satisfacen: i) sus autovalores o valores propios son reales y ii) sus autovectores forman una base completa ortonormal.

- Operador unitario: Un operador $\widehat{U}$ es unitario si $\widehat{U}\widehat{U}^\dagger = \widehat{U}^\dagger\widehat{U} = I$, es decir, que $|\psi\rangle$ y $\widehat{U}|\psi\rangle$ conservan la norma, y por lo tanto $\langle\psi|\psi\rangle = \langle\psi|\widehat{U}^\dagger\widehat{U}|\psi\rangle$. Los operadores unitarios son importantes en teoría cuántica porque con ellos se describen las transformaciones de estados que preservan la norma de los estados (es decir la probabilidad cuántica).

- Operador normal: Un operador normal se define como aquel que conmuta con su adjunto $[\widehat{A}, \widehat{A}^\dagger] = 0$.

3.2.1. Valor Esperado

El valor esperado de un operador[2] $\widehat{A}$ en un estado $|\alpha\rangle$ se define como

$$\left\langle \widehat{A} \right\rangle_\alpha = \langle\alpha|\widehat{A}|\alpha\rangle = \sum a_i |\alpha_i|^2. \tag{3.8}$$

El valor esperado se puede interpretar como el valor promedio ponderado al medir el observable $\widehat{A}$ en el estado $|\alpha\rangle$, ya que es la suma de las probabilidades $|\alpha_i|^2$ de obtener a_i al medir $\widehat{A}$ en el estado $|\alpha\rangle$. Note que a_i es el valor medido de la propiedad A y es uno de los autovalores del operador $\widehat{A}$. Usando identidaes algebraicas podemos reescribir el valor esperado en la forma:

$$\left\langle \widehat{A} \right\rangle_\alpha = \langle\alpha|\,\widehat{A}\,|\alpha\rangle = \langle\alpha|\left(\sum a_i\,|a_i\rangle\,\langle a_i|\right)|\alpha\rangle = \sum a_i\,\langle\alpha|a_i\rangle\,\langle a_i|\alpha\rangle\,, \tag{3.9}$$

en donde hemos usado la forma diagonal del operador $\widehat{A}$ en su base propia $\widehat{A} = \sum a_i\,|a_i\rangle\,\langle a_i|$. Note que adicionalmente se cumple que:

- Si $|\alpha\rangle = |a_i\rangle$, con $|a_i\rangle$ el autovector de $\widehat{A}$ con valor propio a_i, entonces $\left\langle \widehat{A} \right\rangle_\alpha = a_i$.

- Si un valor es propio de $\widehat{A}$ también es propio para cualquier potencia del operador $\widehat{A^m}$ con valores propios a_i^m. Por ejemplo, $\widehat{A^2}\,|a_i\rangle = \widehat{A}\widehat{A}\,|a_i\rangle = \widehat{A}a_i\,|a_i\rangle = a_i^2\,|a_i\rangle$.

3.2.2. Operadores Compatibles

$\widehat{A}, \widehat{B}$ son operadores compatibles si y solo si conmutan, es decir, si satisfacen la siguiente igualdad $[\widehat{A}, \widehat{B}] \equiv \widehat{A}\widehat{B} - \widehat{B}\widehat{A} = 0$. Otra forma de expresar la compatibilidad de dos operadores es que $\widehat{A}, \widehat{B}$ se pueden diagonalizar utilizando una misma base, lo que significa que los dos operadores tienen una base común de autovectores.

[2]En general $\widehat{A}$ puede ser cualquier operador, pero por el teorema espectral los valores esperados más útiles son los de operadores hermíticos. Desde el punto de vista físico, los valores esperados importantes son los de los observables del sistema, ya que esta cantidad es la que se determina experimentalmente.

3.3. Relaciones de Incertidumbre

Aunque de manera general es posible establecer las relaciones de incertidumbre para cualquier par de observables, las más interesantes se presentan cuando aparecen cantidades físicas conjugadas como: posición y momentum, energía y tiempo, momentum angular y ángulo, o cuando los observables no conmutan. Desde 1927 el principio de incertidumbre asociado a Heisenberg se convirtió en uno de los principios fundamentales de la mecánica cuántica. Sin embargo, los desarrollos recientes en los fundamentos de la teoría cuántica ha permitido definir tres tipos diferentes de relaciones de incertidumbre.

3.3.1. Relación de Incertidumbre de Robertson-Schrödinger

La relación de incertidumbre de Robertson-Schrödinger (en una parte de la literatura se menciona solo como Robertson) se puede escribir de la siguiente manera:

$$\sigma(\widehat{A},\psi)\sigma(\widehat{B},\psi) \geq \frac{\left|\langle\psi|\left[\widehat{A},\widehat{B}\right]|\psi\rangle\right|}{2}, \tag{3.10}$$

donde $\sigma(\widehat{X},\psi)$ es la desviación estándar del observable $\widehat{X}$ en el estado $|\psi\rangle$. Para cualquier operador $\widehat{X}$ su desviación estándar se define como $\sigma(\widehat{X},\psi)^2 = \langle\psi|\,\widehat{X}^2\,|\psi\rangle - \langle\psi|\,\widehat{X}\,|\psi\rangle^2$. Esta relación de incertidumbre nos habla acerca de las dispersiones estadísticas inherentes al estado cuántico $|\psi\rangle$ cuando hacemos mediciones de los observables $\widehat{A}$ y $\widehat{B}$, en particular estas relaciones no nos dicen nada acerca del proceso de medición y de los posibles errores y/o perturbaciones experimentales cometidos durante este proceso. En el anterior sentido, las relaciones de incertidumbre de Robertson-Schrödinger son muy diferentes conceptualmente a las propuestas por Heisenberg, cuya generalización se enuncia en los apartados siguientes. Finalmente, debemos hacer notar que esta forma de las relaciones de incertidumbre es la que aparece de manera estándar en los textos de teoría cuántica ya que se deriva matemáticamente de la desigualdad de CauchySchwarz para el espacio de Hilbert de estados cuánticos [6, 120, 122, 123].

3.3.2. Relación de Incertidumbre Ruido-Perturbación generalizada

La relación de incertidumbre generalizada ruido-perturbación se desarrolla en los trabajos pioneros de Ozawa [124, 125] a principios de este siglo. En ellos se intenta darle un marco formal a la idea original de Heisenberg de que si se mide con un error $\epsilon\Big(\widehat{A},\widehat{\psi}\Big)$ una variable dinámica A asociada a un observable $\widehat{A}$ en el estado[3] $\widehat{\psi}$ entonces se produce una perturbación $\eta\Big(\widehat{B},\widehat{\psi}\Big)$ del observable $\widehat{B}$ en el mismo estado. La relación de incertidumbre generalizada se escribe:

$$\epsilon\Big(\widehat{A},\widehat{\psi}\Big)\eta\Big(\widehat{B},\widehat{\psi}\Big) + \epsilon\Big(\widehat{A},\widehat{\psi}\Big)\sigma\Big(\widehat{B},\widehat{\psi}\Big) + \sigma\Big(\widehat{A},\widehat{\psi}\Big)\eta\Big(\widehat{B},\widehat{\psi}\Big) \geq \frac{1}{2}\Big|\mathrm{Tr}\Big\{\Big[\widehat{A},\widehat{B}\Big]\widehat{\psi}\Big\}\Big|, \tag{3.11}$$

para cualquier estado $\widehat{\psi}$. Note que en la anterior expresión intervienen las dispersiones estadísticas definidas para la relación de Robertson-Scrödinger. Para el caso de observables conjugados $\widehat{Q},\widehat{P}$ en un estado de mínima incertidumbre se tiene que $\sigma(\widehat{Q}) = \sigma(\widehat{P}) = \sqrt{(\hbar/2)}$, entonces se tiene la siguiente

[3]En esta sección se usará para describir el estado cuántico el llamado formalismo del operador densidad $\hat{\rho}$. Por esta razón los estados aparecen con notación de operadores $\widehat{\psi}$. El formalismo del operador densidad se desarrollará en la próxima sección 3.4.

desigualdad:

$$\epsilon\Big(\widehat{Q}\Big)\eta\Big(\widehat{P}\Big) + \sqrt{\frac{\hbar}{2}}\Big[\epsilon\Big(\widehat{Q}\Big) + \eta\Big(\widehat{P}\Big)\Big] \geq \frac{\hbar}{2}. \tag{3.12}$$

De la desigualad se observa que pueden ocurrir varios casos. Por ejemplo, que $\epsilon(\widehat{Q})\eta(\widehat{P}) = 0$ con $\epsilon(\widehat{Q}) = 0$ y $\eta(\widehat{P}) \geq \sqrt{(\hbar/2)}$ o que $\eta(\widehat{P}) = 0$ y $\epsilon(\widehat{Q}) \geq \sqrt{(\hbar/2)}$. Para el caso general de una medida precisa de $\widehat{A}$ y una medida sin perturbación de $\widehat{B}$ se utilizan los siguientes teoremas:

- Teorema 1: Para cualquier aparato $A(x)$ y observables $\widehat{A}, \widehat{B}$, si $A(x)$ no perturba $\widehat{B}$, es decir que si $\eta(\widehat{B}, \widehat{\psi}) = 0$, entonces se tiene que.

$$\epsilon\Big(\widehat{A}, \widehat{\psi}\Big)\sigma\Big(\widehat{B}, \widehat{\psi}\Big) \geq \frac{1}{2}\Big|\mathrm{Tr}\Big\{\Big[\widehat{A}, \widehat{B}\Big]\widehat{\psi}\Big\}\Big|, \tag{3.13}$$

  para cualquier estado $\widehat{\psi}$.

- Teorema 2: Para cualquier aparato $A(x)$ y observables $\widehat{A}, \widehat{B}$, si $A(x)$ mide precisamente a $\widehat{A}$, es decir que $\epsilon(\widehat{B}, \widehat{\psi}) = 0$, entonces se tiene que.

$$\sigma\Big(\widehat{A}, \widehat{\psi}\Big)\eta\Big(\widehat{B}, \widehat{\psi}\Big) \geq \frac{1}{2}\Big|\mathrm{Tr}\Big\{\Big[\widehat{A}, \widehat{B}\Big]\widehat{\psi}\Big\}\Big|, \tag{3.14}$$

  para cualquier estado $\widehat{\psi}$ [124, 125].

3.3.3. Relación de Incertidumbre para medidas conjuntas

La relación de incertidumbre para medidas conjuntas arbitrarias de los observables $\widehat{A}, \widehat{B}$ en un estado $|\psi\rangle$ se puede definir mediante la siguiente expresión [126]:

$$\epsilon(\widehat{A}, \psi)\epsilon(\widehat{B}, \psi) \geq \frac{\Big|\langle\psi|\,[\widehat{A}, \widehat{B}\,|\psi\rangle]\Big|}{2}. \tag{3.15}$$

3.4. Operador Densidad

En mecánica cuántica el operador densidad u operador de estado es el operador lineal que codifica todas las propiedades cuánticas de un sistema en su forma más general posible. Es decir, el operador densidad representa el estado cuántico del sistema, en particular cuando no es posible describir el sistema mediante un vector de estado $|\psi\rangle$ o estado puro. El operador densidad permite describir los estados de un sistema cuántico abierto, o el estado de equilibrio termodinámico de un sistema cuántico, o el estado reducido de un subsistema que pertenece a un sistema más grande que de manera general se describen mediante los llamados estados mezclados. El operador densidad de forma general, se puede escribir en la forma:

$$\widehat{\rho} = \sum_j p_j\,|\psi_j\rangle\,\langle\psi_j|\,, \tag{3.16}$$

donde los p_j son los pesos estadístico del estado $|\psi_j\rangle$ en el estado $\widehat{\rho}$ del sistema, que satisfacen $\sum_j p_j = 1$ con $0 \leq p_j \leq 1$. Si se tiene un estado puro $|\psi\rangle$, su operador densidad es $\widehat{\rho} = |\psi\rangle\langle\psi|$, que es también un operador de proyección. Para estados mezclados estrictos, se cumple que $\widehat{\rho^2} \neq \widehat{\rho}$, y para estados puros $\widehat{\rho^2} = \widehat{\rho}$. Para un sistema descrito por un operador densidad (3.16), el valor esperado de cualquier observable A es la media de los los valores esperados en cada estado ψ_j ponderados por

p_j y se puede representar como la traza del observable por el operador densidad:

$$\left\langle \widehat{A} \right\rangle = \sum_j p_j \langle \psi_j | \widehat{A} | \psi_j \rangle = \sum_j p_j \, \mathrm{Tr}\Big\{ |\psi_j\rangle\langle\psi_j| \widehat{A} \Big\} = \mathrm{Tr}\left\{ \sum_j p_j |\psi_j\rangle\langle\psi_j| \widehat{A} \right\} = \mathrm{Tr}\Big\{ \widehat{\rho}\widehat{A} \Big\}, \tag{3.17}$$

donde se uso la propiedad lineal de la traza.

3.5. Qubits

En la teoría de información cuántica se describe el estado cuántico del sistema cuántico más elemental –el sistema de dos niveles— como un qubit. En la llamada base computacional, que representa simplemente dos estados característicos ortogonales, el estado del qubit se escribe como la superposición de los estados $|0\rangle \doteq \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right), |1\rangle \doteq \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$. Los anteriores estados son abstracciones de por ejemplo, los estados de polarización de un fotón individual, los estados de espín para un sistema de espín 1/2, los estados de energía de un átomo de dos niveles, o los estados de excitación de sistemas de estado sólido, entre otros [127]. Si se compara el qubit con el bit, el último representa el contenido de información en un sistema de respuesta binaria cero, que lo podríamos representar por $|0\rangle$, o uno $|1\rangle$, mientras que el qubit gracias a la superposición de estados cuánticos puede representar un contenido de información más alto $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \doteq \left(\begin{smallmatrix}\alpha\\\beta\end{smallmatrix}\right)$, donde α y β son amplitudes de probabilidad complejas que satisfacen la condición de normalización$|\alpha|^2 + |\beta|^2 = 1$ [128].

3.5.1. Representación de Qubits en la esfera de Bloch

La esfera de Bloch es una representación geométrica del espacio de estados de un sistema cuántico de dos niveles, donde se presenta una correspondencia uno a uno entre estados puros de un qubit y los puntos en la esfera de radio 1 en $\mathbb{R}^3$. En esta representación geométrica, cada punto de la superficie de la esfera corresponde a un estado puro del espacio de Hitbert de dimensión compleja 2, y utilizando una reparametrización adecuada de la ligadura $|\alpha|^2 + |\beta|^2 = 1$ se tiene una representación geométrica del estado de un qubit:

$$|\psi\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2)e^{i\phi}|1\rangle, \tag{3.18}$$

donde $(\theta, \phi) \in \mathbb{R}$, $0 \leq \theta \leq \pi$ y $0 \leq \phi \leq 2\pi$ son los ángulos que definen todos los puntos en la superficie de la esfera para todos los estados $|\psi\rangle$ existentes. Note sin embargo, que los estados $|0\rangle, |1\rangle$ correspondientes a $\theta = 0$ y $\theta = \pi$, son casos en los que ϕ no define el estado físico. De esta manera se tiene una corresponencia uno-a-uno entre puntos de la superficie de la esfera y estados del qubit, lo que nos permite la visualización geométrica de los estados puros de un qubit en la representación de Bloch.

Si el qubit se encuentra en un estado mezclado, con cierta probabilidad de estar en el estado $|1\rangle$ y en el estado $|0\rangle$, entonces el qubit se describe mediante una matriz densidad de dimensión 2×2. Una forma elegante de escribir estos operadores densidad es usar el conjunto de operadores $I, \sigma_1, \sigma_2, \sigma_3$, que forman una base completa para las matrices hermíticas 2×2, donde las σ_i, son las matrices de Pauli: $\sigma_1 = \left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right)$, $\sigma_2 = \left(\begin{smallmatrix}0&-i\\i&0\end{smallmatrix}\right)$, $\sigma_3 = \left(\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\right)$. Por lo tanto, los estados mezclados de un qubit pueden expresarse como:

$$\widehat{\rho} = \frac{1}{2}(I_2 + \overrightarrow{n}.\overrightarrow{\sigma}), \tag{3.19}$$

con $\overrightarrow{n} = (n_1, n_2, n_3) \in \mathbb{R}^3$ el vector de Bloch que cumple que: $|\overrightarrow{n}|^2 \leq 1$ y $\overrightarrow{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$.

Si se calcula la traza de ρ^2 usando la anterior expresión, se tiene que:

$$\mathrm{Tr}\Big\{\widehat{\rho^2}\Big\} = \frac{1+|\overrightarrow{n}|^2}{2}, \tag{3.20}$$

de donde se puede concluir que si $|\overrightarrow{n}|^2 = 1$, entonces el punto en la esfera de Bloch está sobre la superficie y por lo tanto $\widehat{\rho}$ es un estado puro, mientras que si $|\overrightarrow{n}|^2 < 1$, entonces el punto que representa el estado está dentro de la esfera y en este $\widehat{\rho}$ es un estado mezclado. En la figura 3.1 se muestra un representación geométrica en la esfera de Bloch de los estados $|0\rangle$ y $|1\rangle$ [129].

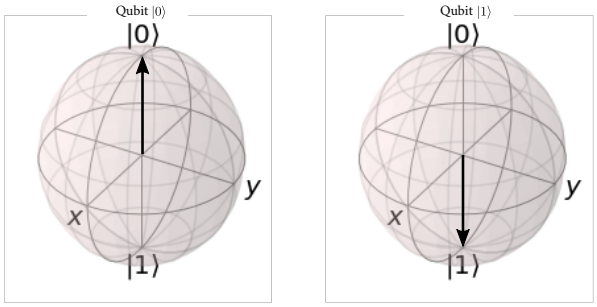

Figura. 3.1 Representación de los qubit $|0\rangle, |1\rangle$ en la esfera de Bloch: figura generada con el Qiskit

3.6. Entropía de Von Neumann

La entropía de Von Neumann es el equivalente cuántico de la entropía de Shannon en computación clásica, simplemente en vez de utilizar una distribución de probabilidad se usa el operador densidad $\widehat{\rho}$ que caracteriza el sistema cuántico A. Se puede definir la entropía de Von Neumann como:

$$S(\widehat{\rho}) = -\,\mathrm{Tr}\{\widehat{\rho}\log_2\widehat{\rho}\}. \tag{3.21}$$

La entropía se puede calcular en términos de los autovalores del operador densidad como se muestra a continuación:

$$\widehat{\rho} = \sum_i \lambda_i \left|\lambda_i\right\rangle \left\langle\lambda_j\right|, \tag{3.22}$$

Como la base propia del operador densidad también satisface condiciones de ortonormalización —por el teorema espectral— $\langle\lambda_i|\lambda_j\rangle = \delta_{ij}$, por lo tanto la entropía de Von Neumann queda:

$$S(\widehat{\rho}) = -\sum_i \lambda_i \log(\lambda_i), \tag{3.23}$$

y además se cumplen las siguientes propiedades matemáticas.

- La entropía $S(\widehat{\rho})$ siempre es positiva para todo operador densidad $\widehat{\rho}$.
- El valor máximo de $S(\widehat{\rho})$ es $\log_2(d)$, siendo d la dimensión del espacio de estados. Esta máxima entropía se obtiene para el estado máximamente mixto $\widehat{\rho} = (1/d)I_{d\times d}$.
- Si el operador $\widehat{\rho}$ es un estado puro arbitrario la entropía $S(\widehat{\rho})$ es cero, ya que si se conoce el estado $|\psi\rangle$ siempre se puede realizar una medición con operadores de medición $\{|\psi\rangle\langle\psi|, 1-|\psi\rangle\langle\psi|\}$, y tener certeza absoluta del resultado de la medición.

- La entropía de un operador densidad es invariante bajo transformaciones unitarias, es decir que:

$$S(\widehat{\rho}) = S\Big(\widehat{U}\,\widehat{\rho}\,\widehat{U^{\dagger}}\Big), \tag{3.24}$$

donde $\widehat{U}$ es un operador unitario que representa cualquier transformación unitaria de estados cuánticos [120].

3.7. Clasificación de los Estados Cuánticos

Si un sistema cuántico está conformado estructuralmente por N subsistemas distinguibles $A, B, C, \ldots$, donde el operador densidad del sistema completo se puede representar por $\widehat{\rho}_{A,B,C,\ldots}$, entonces el operador densidad del i-ésimo sub-sistema se puede obtener al tomar la traza parcial sobre todos los otros subsistemas. En este contexto, se pueden clasificar los estados cuánticos de un sistema físico como se describe a continuación.

3.7.1. Estados Puros

Cómo se menciono anteriormente, un estado puro describe a un sistema físico aislado que puede ser descrito por un vector de estado único, que representa toda la información disponible sobre las propiedades del sistema. La evolución de un estado puro está descrita de forma determinista por medio de la ecuación de Schrödinger.

3.7.2. Estados Separables

Se dice que un sistema cuántico compuesto está en un estado separable si cada uno de los subsistemas se puede encontrar en un estado bien definido y el estado general no es sino el producto tensorial de cada uno de los estados de los subsistemas. De forma general un sistema compuesto $\widehat{\rho}_{A,B,C,\ldots}$ es separable si se puede representar como la suma convexa de operadores densidad:

$$\widehat{\rho}_{A,B,C,\ldots} = \sum_i p_i \widehat{\rho_A^i} \otimes \widehat{\rho_B^i} \otimes \widehat{\rho_C^i} \otimes \ldots, \tag{3.25}$$

donde $\widehat{\rho_A^i}$, $\widehat{\rho_B^i}, \ldots$ son los operadores densidad de cada uno de los subsistemas cuánticos $A, B, C \ldots$, y además se cumple que $\sum_i p_i = 1$ [130].

3.7.3. Estados Entrelazados

Para un sistema cuántico compuesto si un estado no es separable se dice entonces que es entrelazado. La consecuencia más importante de ser entrelazado es que de alguna manera los subsistemas pierden su identidad particular y solo pueden definirse en relación el sistema completo.Para un sistema bipartito con un espacio de Hilbert con dimensión d, se puede decir que el estado es máximamente entrelazado si se puede representar como:

$$|\psi_{AB}\rangle = \frac{1}{\sqrt{d}} \sum_{i=o}^{d-1} |i, i\rangle\,, \tag{3.26}$$

construido a partir de una base separable $|j\rangle \otimes |k\rangle$ con $j, k = 0, \ldots, d-1$.

Note que un estado máximamente entrelazado no se puede construir a partir de cualquier estado de la base utilizando operaciones unitarias locales en cada uno de los espacios de estados de los

subsistemas. Para obtener un estado entrelazado es necesaria una operación unitaria global que actúe en el espacio de Hilbert completo del sistema.

3.7.4. Estados de Bell

Los estados de Bell son los estados máximamente entrelazados para un sistema de 2 qubits que además forman una base ortonormal del espacio de Hilbert de los 2 qubits. Los estados de Bell se escriben como [9, 131]:

$$\left|\phi^{\pm}\right\rangle = \frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle \pm |1\rangle\,|1\rangle), \tag{3.27}$$

$$\left|\psi^{\pm}\right\rangle = \frac{1}{\sqrt{2}}(|0\rangle\,|1\rangle \pm |1\rangle\,|0\rangle). \tag{3.28}$$

3.8. Operaciones con Qubits en Circuitos Cuánticos

Las puertas lógicas cuánticas son circuitos cuánticos básicos que operan sobre qubits, análogas a las puertas lógicas digitales clásicas. Sin embargo, se diferencian en que i) son reversibles, y ii) deben preservar la norma del estado cuántico. Esta última propiedad implica que las puertas lógicas sobre n-qubits se pueden representar mediante matrices unitarias $2^n \times 2^n$. Entre las puertas lógicas más importantes encontramos las matrices de Pauli, que actúan sobre 1-qubit, que además de ser unitarias son hermíticas:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},\ Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{3.29}$$

La compuerta cuántica de Hadamard transforma los estados de la base del qubit $\{|0\rangle, |1\rangle\}$ en una superposición de estos. Se puede generalizar para que opere sobre n-qubits con un comportamiento similar, es decir, la compuerta de Hadamard sobre un n-qubits transforma los elementos de la base $\{|0\rangle, \ldots, |2n-1\rangle\}$ en una superposición de estos. Para 1-qubit La transformación que realiza la compuerta de Hadamard está dada por:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),\ \ H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{3.30}$$

es decir que la formar matricial para la compuerta de Hadamard está dada por, usando por supuesto la base computacional o canónica $\{|0\rangle, |1\rangle\}$:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{3.31}$$

Para 1-qubit también se tienen otras puertas cuánticas importantes como la puerta de fase (denotada por S), y la puerta $\pi/8$ (denotada por T), que se representan mediante las siguientes matrices:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix},\ \ T = \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{bmatrix}. \tag{3.32}$$

Por otra parte, cuando se aplican los operadores de rotación sobre los ejes $\mathbf{x}, \mathbf{y}, \mathbf{z}$, usando las matrices de Pauli como generadoras de la rotación, se obtiene que las rotaciones están representadas

por las siguientes matrices unitarias:

$$R_x(\theta) \equiv e^{-i\theta X/2} = \cos(\theta/2)I - i\sin(\theta/2)X = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \tag{3.33}$$

$$R_y(\theta) \equiv e^{-i\theta Y/2} = \cos(\theta/2)I - i\sin(\theta/2)Y = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \tag{3.34}$$

$$R_z(\theta) \equiv e^{-i\theta Z/2} = \cos(\theta/2)I - i\sin(\theta/2)Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \tag{3.35}$$

A partir de estas expresiones se tiene que la operación de rotación generalizada en un ángulo θ alrededor de un eje cuya dirección está dada por el vector unitario $\boldsymbol{n}$, se puede escribir de la siguiente manera

$$R_{\boldsymbol{n}}(\theta) = e^{(-i\theta/2)\boldsymbol{n}\cdot\widehat{\sigma}} = \cos(\theta/2)I - i\sin(\theta/2)(\boldsymbol{n}\cdot\widehat{\sigma}). \tag{3.36}$$

A modo de ejemplo se puede mostrar que una matriz unitaria parametrizada determina una compuerta cuántica parametrizada

$$U(\theta) = R_y(-\theta) = \begin{bmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \tag{3.37}$$

si actuamos la matriz $U(\theta)$ sobre los elementos de la base $\{|0\rangle, |1\rangle\}$ se obtiene el siguiente comportamiento:

$$U(\theta)|0\rangle = \begin{bmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ -\sin(\theta/2) \end{bmatrix} = \cos(\theta/2)|0\rangle - \sin(\theta/2)|1\rangle, \tag{3.38}$$

$$U(\theta)|1\rangle = \begin{bmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} = \sin(\theta/2)|0\rangle + \cos(\theta/2)|1\rangle. \tag{3.39}$$

Siguiendo el hecho de que una rotación generalizada se puede descomponer en tres rotaciones sucesivas a través de los llamados ángulos de Euler, entonces se puede demostrar que para un qubit, cualquier operación U unitaria, se puede descomponer en términos de 3 rotaciones y una fase global. Existen entonces números reales $\alpha, \beta, \gamma, \delta$ tales que U se puede escribir como, en la llamada descomposición Z-Y [120]:

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta). \tag{3.40}$$

Usando las definiciones de las matrices de rotación y realizando la multiplicación de matrices, obtenemos

$$U = \begin{bmatrix} e^{i(\alpha-\beta/2-\delta/2)}\cos(\gamma/2) & -e^{i(\alpha-\beta/2+\delta/2)}\sin(\gamma/2) \\ e^{i(\alpha+\beta/2-\delta/2)}\sin(\gamma/2) & e^{i(\alpha+\beta/2+\delta/2)}\cos(\gamma/2) \end{bmatrix}. \tag{3.41}$$

Note que como U es unitaria, sus filas y columnas son ortonormales.

Otra descomposición útil de U es la siguiente. Se pueden definir operadores unitarios A, B, C, de tal manera que $ABC = I$ y $U = e^{i\alpha}AXBXC$, donde α es un factor de fase global. Entonces, por comparación con la expresión (3.40) se tiene que [120]:

$$A \equiv R_z(\beta)R_y(\gamma/2),\ B \equiv R_y(\gamma/2)R_z(-((\delta+\beta)/2),\ C \equiv R_z((\delta+\beta)/2). \tag{3.42}$$

3.8.1. Operación Controladas CNOT

Se pueden implementar operaciones controladas complejas utilizando circuitos cuánticos construidos a partir de operaciones elementales. Las operaciones controladas más elementales se pueden definir para dos qubits de entrada, el llamado qubit de control y el qubit blanco, objetivo o de destino. Entre estas puertas una de las más importantes es la llamada puerta *CNOT*, cuya acción sobre la base computacional es $|c\rangle |t\rangle \longmapsto |c\rangle |t \oplus c\rangle$, donde $|c\rangle$ es el qubit de control y $|t\rangle$ el qubit objetivo. La acción de la puerta *CNOT* se puede interpretar de la siguiente manera: si el qubit de control se establece en $|1\rangle$ entonces el qubit objetivo se invierte, de lo contrario el qubit objetivo no se altera. La puerta *CNOT* se puede escribir como la compuerta controlada más general $C(U)$ (que se describirá más adelante). En la base computacional $|c, t\rangle$ la forma matricial de la compuerta es:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{3.43}$$

En la figura 3.2 se muestra la convención gráfica para el circuito que representa la compuerta *CNOT*.

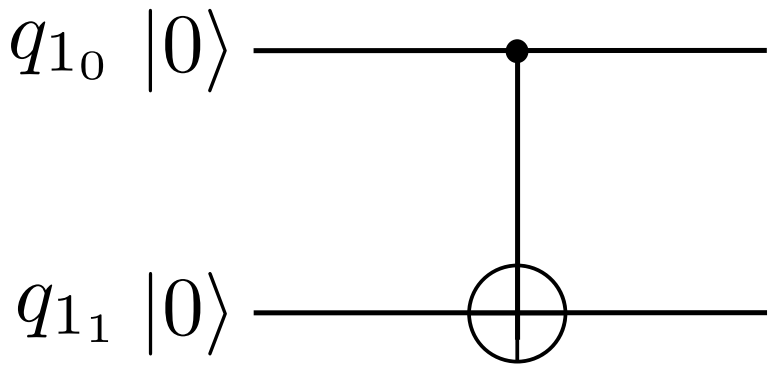

Figura. 3.2 Circuito para la puerta controlada *CNOT*. La línea superior representa el qubit de control, la línea inferior el qubit objetivo, figura generada con el Qiskit

En forma más general, la compuerta $C(U)$ controlada actúa de tal forma que si se establece el qubit de control en el estado $|1\rangle$ entonces se aplica la operación U al qubit objetivo, de lo contrario, se deja el qubit de destino inalterado. En otras palabras, se realiza la operación $|c\rangle |t\rangle \to |c\rangle U^c |t\rangle$. Matricialmente la puerta $C(U)$ controlada tiene la forma:

$$C(U) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{00} & U_{01} \\ 0 & 0 & U_{10} & U_{11} \end{pmatrix}. \tag{3.44}$$

La operación $C(U)$ controlada está representada por el circuito que se muestra en la figura 3.3, en donde la ejemplificamos con la puerta Y.

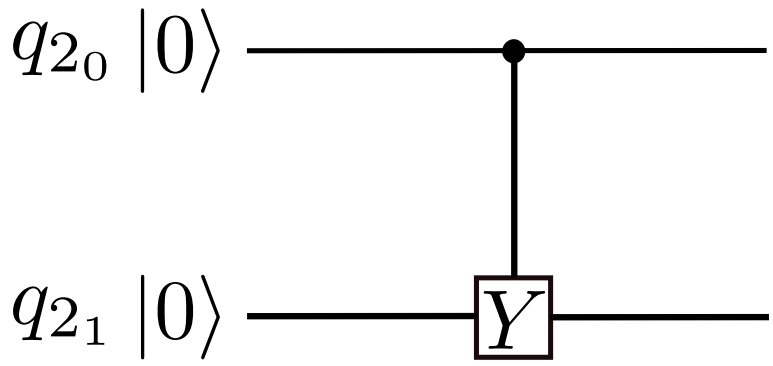

Figura. 3.3 Operación controlada $C(U)$: La línea superior es el qubit de control y la línea inferior es el qubit de destino. Si se establece el qubit de control, entonces se aplica U, de lo contrario el qubit objetivo queda inalterado. Se ejemplifica con la puerta de Pauli Y, figura generada con el Qiskit

La implementación de la operación $C(U)$ controlada para un qubit se basa en la descomposición $U = e^{i\alpha}AXBXC$ descrita anteriormente. El primer paso aplica un cambio de fase $e^{i\alpha}$ controlado en el qubit objetivo de la forma:

$$|00\rangle \to |00\rangle,\ |01\rangle \to |01\rangle,\ |10\rangle \to e^{i\alpha}|10\rangle,\ |11\rangle \to e^{i\alpha}|11\rangle. \tag{3.45}$$

De esta manera la operación $C(U)$ controlada se puede escribir como $C(U) = C(e^{i\alpha})AXBXC$ siendo A, B, C operaciones de un qubit que satisfacen la condición $ABC = I$. Si el control del qubit está establecido, la operación $U = AXBXC$ es aplicada al segundo qubit, y si el qubit de control no se establece la operación $ABC = I$ se aplica al segundo qubit y no se realiza ningún cambio. Esta compuerta puede generalizarse a más qubits de entrada y de salida. Por ejemplo, si se tienen $n + k$-qubits y U es un operador unitario sobre los últimos k-qubits, entonces la operación controlada $C_n(U)$ se define a través de la siguiente expresión:

$$C_n(U)\,|x_1x_2\ldots x_n\rangle\,|\psi\rangle = |x_1x_2\ldots x_n\rangle\,U^{x_1x_2\ldots x_n}\,|\psi\rangle \tag{3.46}$$

donde $x_1x_2\ldots x_n$ es el producto binario de los bits, es decir que el operador U solo se aplica a los últimos k-qubits, si los primeros n-qubits son iguales a uno, de lo contrario, no se ejecuta nada.

3.8.2. Puertas Cuánticas Universales

Se dice que un conjunto de compuertas es universal si cualquier operación unitaria puede aproximarse con precisión arbitraria mediante la aplicación sucesiva de un número determinado de las compuertas universales. En otras palabras, un conjunto universal de compuertas permite escribir cualquier operación unitaria como un circuito cuántico hecho por un único conjunto de módulos. Para la computación cuántica, se puede demostrar que cualquier operación unitaria se puede aproximar con precisión arbitraria utilizando únicamente las puertas cuántica: $Hadamard = H$, $Phase = S$, $CNOT$, y $\pi/8 = T$. Los siguientes proposiciones son válidas en este caso [120]:

- Si U es un operador unitario en que actúa en el espacio de n-qubits, entonces se puede descomponer en máximo $n(n-1)/2$ operaciones unitarias de un solo qubit.
- Cualquier operación unitaria arbitraria de un qubit se puede expresar como una rotación de un qubit más una $CNOT$.
- La rotación de un qubit se puede aproximar mediante un conjunto de puertas H, S, y T.

3.9. Medidas Proyectivas

Como se mencionó anteriormente, en la mecánica cuántica a cada propiedad del sistema físico se le asocia un operador lineal hermítico, autoadjunto, que actúa sobre el espacio de estados $\mathcal{H}$. Debido al carácter inherentemente probabilístico y no unitario de la medición cuántica, el proceso de medición de propiedades físicas sobre un sistema cuántico debe establecer con cuidado las posibles formas mediante las cuales se puede extraer la información del sistema. La forma más sencilla del proceso de medición cuántica son las llamadas mediciones proyectivas que discutiremos a continuación. Usando la descomposición espectral de un operador autoadjunto que representa un observable $\widehat{M} \in \beta(\mathcal{H})$ [4]:

$$\widehat{M} = \sum_i \lambda_i P_i, \tag{3.47}$$

donde $P_i = \sum_j |\phi_i^j\rangle\langle\phi_j^i|$ son operadores de proyección en el sub-espacio vectorial asociado al autovalor λ_i, que satisfacen $P_i P_j = P_i \delta_{ij}$. Note que los λ_i son únicos porque corresponden a los autovalores de $\widehat{M}$. Entonces, las posibles medidas de $\widehat{M}$ sobre un estado cuántico $|\psi\rangle$ son sus autovalores $\{\lambda_i\}$ con una probabilidad asociada de obtener λ_i igual a:

$$p(i) = \langle\psi|P_i|\psi\rangle. \tag{3.48}$$

Inmediatamente después de la medición, el estado del sistema es proyectado al estado correspondiente al resultado de la medida, es decir se ha obtenido el i-ésimo resultado posible, y el estado del sistema pasa a ser –de manera no continua y no unitaria–

$$\left|\psi'\right\rangle = \frac{P_i\,|\psi\rangle}{\sqrt{\langle\psi|P_i|\psi\rangle}}. \tag{3.49}$$

Si en lugar de un estado puro $|\psi\rangle$ el sistema está descrito por un estado mezclado $\widehat{\rho}$, entonces la probabilidad de obtener el resultado λ_i es

$$p(i) = \mathrm{Tr}(P_i\,\widehat{\rho}), \tag{3.50}$$

y el estado después del proceso de medición será:

$$\widehat{\rho'} = \frac{P_i \widehat{\rho} P_i}{\mathrm{Tr}(P_i\,\widehat{\rho})}. \tag{3.51}$$

La anterior descripción del proceso de medición proyectiva en mecánica cuántica se puede generalizar mediante las llamadas mediciones de operadores valuados positivos o POVM por sus siglas en inglés *Positive Operator Valued Measure (POVM)*. Estas medidas se describen mediante una colección de operadores $\{A_i\}$ de $\mathcal{L}(\mathcal{H})$ [5], denominados operadores de Kraus, que satisfacen $\sum_i A_i^\dagger A_i = I$, donde I es el operador identidad. Entonces la probabilidad de obtener el i-ésimo resultado en la medida está dado por:

$$p(i) = \mathrm{Tr}\Big(A_i \widehat{\rho} A_i^\dagger\Big). \tag{3.52}$$

[4] $\beta(\mathcal{H})$ representa el espacio de todos los operadores lineales hermíticos que actúan sobre $\mathcal{H}$.
[5] $\mathcal{L}(\mathcal{H})$ representa el espacio de todos los operadores lineales que actúan sobre $\mathcal{H}$.

y el estado del sistema después de la medida es

$$\widehat{\rho'} = \frac{A_i \widehat{\rho} A_i^\dagger}{\mathrm{Tr}\Big(A_i \widehat{\rho} A_i^\dagger\Big)}. \tag{3.53}$$

Se puede ver que el esquema de mediciones proyectivas son un caso especial de las POVM, cuando los operadores de Kraus coinciden con los operadores de proyección $A_i = A_i^\dagger = P_i$. La diferencia principal entre medidas proyectivas y las POVM, es que en las POVM los operadores de Kraus que definen la medición no tienen porque ser necesariamente ortogonales [132].

3.10. Entrelazamiento Cuántico

El entrelazamiento es un fenómeno inherentemente cuántico que no tiene un equivalente clásico, incluso Schrödinger lo llamó el *rasgo cuántico por excelencia*. Esta correlación cuántica aparece en sistemas compuestos cuando el estado del sistema no es separable —ver sección 3.7— e implica que el estado de un sistema físico se describe mediante un estado global que involucra a todas las partes del sistema sin importar que tan separados espacialmente están estas partes. Esta característica no-local del entrelazamiento, que tanto molestaba a Einstein, implica que en el sistema físico existen correlaciones que permiten modificar el estado de una de las partes del sistema modificando el estado de otra de las partes, independientemente de la posición en la cual se encuentren localizadas. El ejemplo de estados entrelazados más estudiado son los estados máximamente entrelazados de Bell para un sistema cuántico de dos qubits. En este caso, si se conoce el resultado del estado en la medición de uno de los qubits entonces se puede predecir de forma exacta el estado del segundo qubit sin ser observado. Del hecho de que los estados de Bell sean máximamente entrelazados se sigue que los operadores densidad reducidos de un qubit son los de máxima mezcla, es decir $\widehat{\rho} = \frac{1}{2} I$ [133, 134].

3.11. Teleportación de un Estados Cuánticos

El protocolo de teleportación de estados cuánticos permite transferir un estado cuántico desconocido a una localización alejada usando entrelazamiento cuántico distribuido. Este protocolo de deportación nos permite transmitir fielmente un estado $|\phi\rangle_X$ de un sistema cuántico X entre dos partes espacialmente separadas. El protocolo de teleportación se puede describir como sigue: Sea Alice un transmisor que envía un mensaje a un lugar alejado, y un receptor Bob que recibe dicha información y además Alice y Bob comparten un par de fotones máximamente entrelazados. Ahora para realzar la teleoperación, supongamos que Alice quiere teleportar un estado general desconocido $|\alpha\rangle = a\,|0\rangle + b\,|1\rangle$ a Bob, utilizando un canal cuántico, un par EPR de la forma $|\beta\rangle = 1/\sqrt{2}(a\,|0\rangle\,|0\rangle + b\,|1\rangle\,|1\rangle)$, por lo tanto se puede notar que el estado inicial del sistema queda representado por.

$$\begin{aligned} |\alpha\rangle \otimes |\beta\rangle &= (a\,|0\rangle + b\,|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle + |1\rangle\,|1\rangle) && (3.54)\\ &= \frac{1}{\sqrt{2}}(a\,|0\rangle\,(|0\rangle\,|0\rangle + |1\rangle\,|1\rangle) + b\,|1\rangle\,(|0\rangle\,|0\rangle + |1\rangle\,|1\rangle)) && (3.55)\\ &= \frac{1}{\sqrt{2}}(a\,|000\rangle + a\,|011\rangle + b\,|100\rangle + b\,|111\rangle) && (3.56) \end{aligned}$$

posteriormente Alice aplica la compuerta *Cnot* al qubit de donde obtiene la expresión.

$$C_{not}\left|\alpha\right\rangle \otimes \left|\beta\right\rangle \quad = \quad \frac{1}{\sqrt{2}} C_{not}(a\left|000\right\rangle + a\left|011\right\rangle + b\left|100\right\rangle + b\left|111\right\rangle), \tag{3.57}$$

y luego aplica la compuerta hadamard al qubit que desea teleportar.

$$\begin{aligned}(H)(C_{not})\left|\alpha\right\rangle \otimes \left|\beta\right\rangle \quad &= \quad \frac{1}{2}[\left|00\right\rangle (a\left|0\right\rangle + b\left|1\right\rangle) + \left|10\right\rangle (a\left|0\right\rangle - b\left|1\right\rangle) + \left|01\right\rangle (a\left|1\right\rangle + b\left|0\right\rangle) \\ &+ \quad \left|11\right\rangle (a\left|1\right\rangle - b\left|0\right\rangle)].\end{aligned} \tag{3.58}$$

De la expresión se deduce que si Alice mide sobre sus qubit obtendrá con igual probabilidad uno de los estados $\left|00\right\rangle, \left|01\right\rangle, \left|10\right\rangle, \left|11\right\rangle$, y el qubit de Bob es proyectado a uno de los estados $a\left|0\right\rangle + b\left|1\right\rangle, a\left|1\right\rangle + b\left|0\right\rangle, a\left|0\right\rangle - b\left|1\right\rangle, a\left|1\right\rangle - b\left|0\right\rangle$. Por ultimo en el protocolo de teleoperación Alice le comunica a Bob el resultado de su mediciones a través de dos bits clásicos de información y utilizando un canal clásico, seguidamente Bob aplica una de las compuertas I, X, Z o Y en función de los bits clásicos recibidos, y luego estados de los qubits de Bob son transformados e el estado $\left|\alpha\right\rangle$ exactamente [135].

3.12. Teorema de no Clonación

Una de las características más importantes de la mecánica cuántica aplicada a computación es la propiedad de no clonación de estados cuánticos [136]. Esta propiedad establece que no se puede copiar un estado cuántico desconocido arbitrario, ya que no se puede medir el estado $\left|\psi\right\rangle$ de un sistema cuántico a través de una sola medición. Como se discutió en la sección 3.9 el resultado de toda medición de un observable A es uno de sus valores propios, lo cual da muy poca información acerca del sistema $\left|\psi\right\rangle$. Entonces para reconstruir $\left|\psi\right\rangle$, por ejemplo a través de un protocolo de tomografía de estado cuántico, se deben medir los valores esperados de varios observables, lo que implica calcular promedios estadísticos sobre estados del sistema idénticamente preparados. Un razonamiento para intentar evitar esta imposibilidad de los sistemas cuánticos es tomar el sistema en el estado desconocido $\left|\psi\right\rangle$ y dejarlo interactuar con N sistemas diferentes preparados previamente en un estado en blanco de referencia $\left|S\right\rangle$ para preparar $N+1$ copias del estado inicial, es decir que:

$$\left|\psi\right\rangle \otimes \left|S\right\rangle \otimes ... \otimes \left|S\right\rangle \longrightarrow \left|\psi\right\rangle \otimes \left|\psi\right\rangle \otimes ... \otimes \left|\psi\right\rangle, \tag{3.59}$$

El anterior razonamiento permitiría determinar el estado cuántico de un sistema sin siquiera medirlo, ya que podría medir las N copias y dejar el original intacto. El teorema de no clonación indica que, *"No existen operaciones cuánticas que puedan duplicar perfectamente un estado cuántico arbitrario."* Supongamos por un momento, que si existe una máquina copiadora de estados cuánticos con dos sub-sistemas A, B, donde A son los datos que inician en un estado $\left|\psi\right\rangle$ puro desconocido, y B es la copia que también inicia en un estado $\left|S\right\rangle$ puro de referencia, es decir que el estado inicial de la máquina es $\left|\psi\right\rangle \otimes \left|S\right\rangle$. Por lo tanto para que la máquina copiadora funcione es necesario efectuar una operación,una evolución unitaria U, que satisfaga:

$$\left|\psi\right\rangle \otimes \left|S\right\rangle \xrightarrow{U} (\left|\psi\right\rangle \otimes \left|S\right\rangle) = \left|\psi\right\rangle \otimes \left|\psi\right\rangle. \tag{3.60}$$

Es decir que la operación unitaria U copia el estados desconocido de los datos $\left|\psi\right\rangle$ en el estado del sub-sistema B. Una condición razonable que se le puede exiguir a esta máquina copiadora de estados cuánticos es que funcione para cualquier estado desconocido que quiera copiar, así si la aplicamos

sobres los estados $|\psi\rangle$ y $|\varphi\rangle$, tenemos:

$$U(|\psi\rangle \otimes |S\rangle) = |\psi\rangle \otimes |\psi\rangle\,, \tag{3.61}$$

$$U(|\varphi\rangle \otimes |S\rangle) = |\varphi\rangle \otimes |\varphi\rangle\,. \tag{3.62}$$

Si se toma el producto interno de las anteriores ecuaciones, se tiene que:

$$\langle\psi|\varphi\rangle = \langle\psi|\varphi\rangle^2\,. \tag{3.63}$$

De la anterior igualdad se puede ver que como $x = x^2$ sólo tiene dos soluciones $x = 0$ y $x = 1$, entonces $|\psi\rangle = |\varphi\rangle$ o $|\psi\rangle$ y $|\varphi\rangle$ son ortogonales, lo que significa que la maquina copiadora solo funciona con estados específicos, y por lo tanto no puede copiar un estado cuánticos general. Es decir no existe un protocolo universal para clonar estados cuánticos arbitrarios, resultado que es aprovechado aumentar la seguridad de los sistemas de encriptación cuánticos, por ejemplo para generar las claves criptográficas.

3.13. Polarización de la Luz

La polarización es un fenómeno que puede producirse en las ondas electromagnéticas, en el cual el campo eléctrico oscila sólo en un plano determinado. Este plano puede definirse por dos vectores, uno de ellos paralelo a la dirección de propagación de la onda y otro perpendicular a esa dirección que representa el campo eléctrico. En la naturaleza se encuentran diferentes tipos de polarización que dependen de la dirección de oscilación del campo eléctrico con respecto al eje de propagación de la señal como se muestra en la figura 3.4 [137].

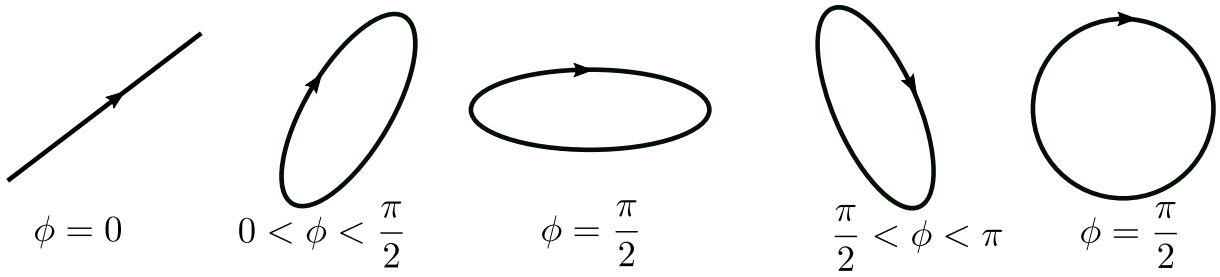

Figura. 3.4 Modos de polarización de la luz: Diferentes tipos de polarización, cada uno de los estados de polarización en la figura están denotados por la dirección de la oscilación (polarización lineal $\phi = 0$) o por el sentido de rotación (polarización circular o elíptica). Figura tomada de [138].

Para el caso practico más común de la codificación de información usando la polarización lineal de fotones se tienen cuatro formas de polarización posibles: Polarización horizontal $\leftrightarrow$, polarización vertical $\updownarrow$, polarización diagonal a derecha $\nearrow$, y polarización diagonal a izquierda $\nwarrow$. Por lo tanto los estados de polarización de fotones se pueden describir a partir de vectores en un espacio vectorial complejo de dimensión 2, donde se puede definir las bases $\{|x\rangle\,, |y\rangle\}$ y $\{|R\rangle\,, |L\rangle\}$ que corresponden a polarización lineal en x, y. Si se realiza una rotación alrededor del eje de propagación de los estados lineales $\{|x\rangle\,, |y\rangle\}$ tenemos estados de polarización lineal

$$|x\rangle \longrightarrow \cos\theta\,|x\rangle + \sin\theta\,|y\rangle\,; \tag{3.64}$$

$$|y\rangle \longrightarrow -\sin\theta\,|x\rangle + \sin\theta\,|y\rangle\,, \tag{3.65}$$

Como ejemplo del anterior formalismo, consideremos una fuente de pares de fotones entrelazados en polarización que emergen de un proceso de desexcitación de un átomo excitado que emite dos fotones que se propagan a lo largo del eje z en direcciones opuestas [13]:

$$|\phi\rangle_{AB}^{\pm} = \frac{1}{\sqrt{2}}(|x\rangle_A |x\rangle_B \pm |y\rangle_A |y\rangle_B). \tag{3.66}$$

3.14. Concurrencia

Mediante la concurrencia de Wootters [139, 140], es posible mostrar que para cualquier sistema bipartito de dos qubits, su entrelazamiento es una función monótona de la concurrencia. Es decir, se obtienen las mismas condiciones de separabilidad o no de de los estados bipartitos. La concurrencia puede variar de $C = 0$ para los estados separables, hasta $C = 1$ para estados máximamente entrelazados. La concurrencia para un estado de un par de qubits se puede calcular mediante la siguiente expresión [141]

$$C(\rho) = max\{0, \sqrt{\lambda_1} - \sqrt{\lambda_2} - \sqrt{\lambda_3} - \sqrt{\lambda_4}\}, \tag{3.67}$$

donde λ_i son los autovalores en orden decreciente del operador ξ

$$\xi = \rho(\sigma_y \otimes \sigma_y)\rho^*(\sigma_y \otimes \sigma_y). \tag{3.68}$$

En la anterior expresión ρ^* representa el complejo conjugado de ρ en la base estándar $|++\rangle, |+-\rangle, |-+\rangle, |--\rangle$, y σ_y es la matriz de Pauli para un qubit.

Uno de los aspectos importantes y necesarios para caracterizar las correlaciones cuánticas en procesos de información, es analizar su dinámica en presencia de ruido. En un llamado estado X, la matriz densidad bipartita ρ^{AB} solo contiene elementos a lo largo de sus dos diagonales principales:

$$\rho^{AB} = \begin{pmatrix} a & 0 & 0 & w \\ 0 & b & z & 0 \\ 0 & z^* & c & 0 \\ w^* & 0 & 0 & d \end{pmatrix}; \text{ con } a+b+c+d = 1. \tag{3.69}$$

En la literatura se ha demostrado que este tipo de estados aparecen naturalmente en muchos sistemas físicos [142], como por ejemplo los estados puros de Bell. Los estados X tienen la particularidad que en muchos casos, bajo evolución con ruido retienen la forma X del estado [143]. Esto es muy conveniente, porque la concurrencia para el estado X se puede calcular con la sencilla expresión [144]:

$$C(\rho^{AB}) = 2\ max\{0, |z| - \sqrt{ad}, |w| - \sqrt{bc}\}. \tag{3.70}$$

3.15. Discordia Cuántica

La discordia es otra de las correlaciones cuánticas que permiten cuantificar la no clasicalidad de las correlaciones que se presentan en los sistemas físicos. La discordia cuántica se presenta cuando se extiende la información mutua de un sistema clásico compuesto por sistemas A y B con una correlación[6]definida por una distribución de probabilidad $p(A, B)$, la cual es obtenida como una

[6]Las correlaciones se pueden ver como el cambio esperado en las características físicas a medida que un sistema cuántico interactúa con un sistema de medida, según la visión del estado relativo de la física cuántica, una medición cuántica nunca da resultados concluyentes, sino que establece correlaciones entre el sistema y el dispositivo de medición. Una de las correlaciones más estudias en es el entrelazamiento cuántico.

medida de información mutua de la siguiente manera:

$$I(A,B) = H(A) + H(B) - H(A,B). \tag{3.71}$$

Siendo $H(\cdot)$ la entropía de Shannon definida por $H(p) = \sum_i p_i \log_2 p_i$ [145]. La información mutua clásica se puede expresar de una forma equivalente utilizando la regla de Bayes $J(A:B) = H(A) - H(A|B)$ [146], donde la entropía condicional $H(A|B)$ cuantifica la ignorancia sobre el sistema A dado que se conoce B. Para un sistema cuántico representado por el operador densidad ρ, la entropía de Shannon se reemplaza por la entropía de von Neumann $s(\rho) = -\operatorname{Tr}(\rho \log_2 \rho)$. Note que clásicamente tanto $I(A:B)$ como $J(A:B)$ son exactamente iguales. Sin embargo, cuánticamente puede existir una diferencia entre estas dos expresiones, ya que la segunda definición implica una medición local sobre el subsistema B [146, 147].

Si cuánticamente se quiere calcular la expresión $J(A:B)$ podemos restringir a medidas realizadas localmente sobre el subsistema B descritas por un conjunto de mediciones proyectivas $\{\prod_k\}$. Con esta suposición, se puede observar que el estado cuántico post-medición es

$$\frac{\rho_k}{p_k} = \frac{(I \bigotimes \prod_k)\rho(I \bigotimes \prod_k)}{\operatorname{Tr}((I \bigotimes \prod_k)\rho(I \bigotimes \prod_k))}, \tag{3.72}$$

donde I es el operador identidad del subsistema A. Lo anterior permite definir un operador densidad condicional y por lo tanto se puede definir un análogo de la entropía condicional $S(\rho|\prod_k) = \sum_k p_k S(\rho_k)$. Entonces, la información mutua se puede escribir de la siguiente manera, $J(A:B) = J(\rho|\prod_k) = S(\rho^A) - S(\rho|\prod_k)$. De esta segunda definición de la información mutua, se tiene que las medidas proyectivas en el sistema B remueven todas las correlaciones no clásicas entre A y B, por lo tanto el valor de $J(\rho|\prod_k)$ depende de la elección de $\{\prod_k\}$. Como se quiere eliminar todas las correlaciones no-clásicas del sistema, se maximiza J sobre todas las posibles mediciones proyectivas $\{\prod_k\}$, y se puede escribir la correlación clásica como $Q(\rho) = \sup_{\{\prod_k\}} J(\rho|\prod_k)$. Por lo tanto la discordia cuántica se puede definir como

$$D(\rho) = I(\rho) - Q(\rho). \tag{3.73}$$

En la actualidad, la discordia cuántica es la más «basica» de las correlaciones cuánticas y ofrece la información suficiente para saber la naturaleza cuántica de las correlaciones entre dos sistemas. Es decir que si la discordia es cero, se está en un dominio clásico en donde solo hay correlaciones clásicas, pero si la discordia es distinta de cero se tienen correlaciones no-clásicas en el sistema [148].

3.16. Medidas de Canal Cuántico Ruidoso

Una de las características principales de la mecánica cuántica es describir medidas, como el cambio de estado des un sistema mediante operaciones básicas, y poder obtener un resultado con una probabilidad, entre las medidas cuánticas. La medida mas simple que se tiene es la traza $\rho \longrightarrow tr(\rho)$, que se define como una operación si H_a pertenece al espacio de Hitert, y tiene una base otonormal $|1\rangle, ..., |d\rangle$, ademas $H_a^{'}$ es el espacio de salida para el estado $|0\rangle$, por lo tanto se puede definir una operación cuántica como

$$\varepsilon(\rho) = \sum_{i=1}^{d} |0\rangle\langle i|\rho|i\rangle\langle 0|, \tag{3.74}$$

siendo ε se define como una operación cuántica, donde $\varepsilon(\rho) = tr(\rho)|0\rangle\langle 0|$ que representa la función traza.

Utilizando la definición de traza parcial se pueden obtener resultados mas significativos en las operaciones cuánticas. Por ejemplo, si se tiene un sistema QR y se quiere trazar sabre R, con $|j\rangle$ base del sistema R, entonces una operación lineal $E^j : H_{QR} \to H_R$ se define mediante la expresión

$$E_i(\sum_j \lambda_i|q_i\rangle|j\rangle) \equiv \lambda_i|q_i\rangle, \varepsilon(\rho) = \sum_i E_i \rho E_i^\dagger, \tag{3.75}$$

$$\varepsilon(\rho \bigotimes |i\rangle\langle j^{'}|) = \rho\delta_{jj'} \equiv tr(\rho \bigotimes |j\rangle\langle j^{'}|), \tag{3.76}$$

donde ρ es cualquier operación hermítica en el espacio de estados del sistema Q y $|j\rangle$ y $|j^{'}\rangle$ pertenecen a la base ortonormal del sistema R.

3.16.1. Dos Qubits Sujetos al Canal de *Phase Damping*

Partiendo de un análisis geométrico se puede caracterizar un sistema bipartito de dos qubits A y B, interactuando con su entorno local de forma independiente. Se pueden estudiar estos sistemas y construir una relación dinámica entre entrelazamiento y discordia geométrica, utilizando la norma traza y la norma de Hilbert-Schmidt. por lo tanto para describir la evolución del sistema, se tiene dos qubits sujetos cada uno a una dinámica de canal markoviano de phase damping que es un entorno simétricos, donde la probabilidad de que p de cambio es la misma. Bajo estas condiciones y partiendo de estados diagonal de Bell, se puede mostrar que el sistema evolucionado conserva la misma estructura, es decir que si se parte de un estado inicial $\rho_{AB} = \frac{1}{4}(I_4 + \sum_i r_i\sigma_i \otimes \sigma_i)$, con $\overrightarrow{r} = (r_1(0), r_2(0), r_3(0))$ y operadores de Kraus M_i^A, M_j^B $i, j = 0, 1, 2$, dados por $M_0^{pd} = \sqrt{1-p}I, M_1^{pd} = \sqrt{P}\left(\begin{smallmatrix}1 & 0\\ 0 & 0\end{smallmatrix}\right), M_2^{pd} = \sqrt{P}\left(\begin{smallmatrix}0 & 0\\ 0 & 1\end{smallmatrix}\right)$, por lo tanto el vector correlación evolucionado se expresa como

$$\overrightarrow{r}_{pd} = r_1(0)(1-p)^2\widehat{i} + r_2(0)(1-p)^2\widehat{j} + r_2(0)(1-p)^2\widehat{k}. \tag{3.77}$$

3.16.2. Canales Cuánticos bit flip y Phase flip para un Qubit

En la teoría de canales cuánticos con ruido, se define el canal bit blip como una operación fundamental que cambia un qubit en estado $|0\rangle$ a estado $|1\rangle$, y similarmente cambia el estado $|1\rangle$ a $|0\rangle$, con una probabilidad de $1-p$. Los operadores de canal se representan como:

$$E_0 = \sqrt{p}I = \sqrt{p}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, E_1 = \sqrt{1-p}X = \sqrt{1-p}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}. \tag{3.78}$$

por otra parte, para el canal cuántico fase flip las operaciones elementales para un solo qubit se pueden representar:

$$E_0 = \sqrt{p}I = \sqrt{p}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, E_1 = \sqrt{1-p}Z = \sqrt{1-p}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}, \tag{3.79}$$

y si se considera $p = 1/2$, se tiene un caso especial de la operación inversora de fase donde $\rho \longrightarrow \varepsilon(\rho) = p_0\rho p_0 + p_1\rho p_1$ con $p_0 = |0\rangle\langle 0|, p_1 = |1\rangle\langle 1|$, medida del qubit en la $|0\rangle, |1\rangle$.

Para el canal bit-phase flip las operaciones elementales representan un cambio de fase y cambio de bit

$$E_0 = \sqrt{p}I = \sqrt{p}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, E_1 = \sqrt{1-p}Y = \sqrt{1-p}\begin{pmatrix}0 & -I\\ I & 0\end{pmatrix}. \tag{3.80}$$

El canal de polarización se entiende como un ruido cuántico, en donde por ejemplo, si se tiene un sistema de un qubit con una probabilidad p de que este polarizado, y además se tiene el estado mixto $\frac{I}{2}$ con una probabilidad $1-p$, entonces el estado cuántico después de evolucionar con el rudo se define como. $\varepsilon(\rho) = \frac{pI}{2} + (1-p)\rho$, y para un ρ arbitrario se puede generalizar como $\frac{I}{2} = \frac{\rho + X\rho X + Y\rho Y + Z\rho Z}{4}$, de donde se puede representar el estado cuántico en función del ruido

$$\varepsilon(\rho) = (1 - \frac{3p}{4})\rho + \frac{p}{4}(X\rho X + Y\rho Y + Z\rho Z). \tag{3.81}$$

Con el modelo de amplitud damping se puede hacer una descripción de la disipación de la energía, se modelan los efectos del sistema cuántico debido a perdidas de energía. Por ejemplo si se tiene la transformada unitaria $B = exp[\theta(a^\dagger b - ab^\dagger)]$ donde $a, a^\dagger y b, b^\dagger$ son operadores de aniquilación y creación de fotones

$$\varepsilon_{AD}(\rho) = E_0\rho E_0^\dagger + E_1\rho E_1^\dagger, con E_k = \langle k|B|0\rangle, E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, E_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}. \tag{3.82}$$

donde $\gamma = sin2\theta$ se considera como la probabilidad de perder un fotón. Y con el modelo phase damping se puede describir la perdida de información en un sistema cuántico si tener perdidas de energía. físicamente este efecto se puede entender observando la dispersión aleatoria que sufre un fotón cuan des desplaza en un guía de onda, en este sistema los estados de la energía no cambian en función del tiempo, solamente acumulan una fase que es proporcional al valor propio, es decir que cuando un sistema cuántico evoluciona en el tiempo se pierde información de las fases relativas de los estados propios de la energía. Por ejemplo, qubits $|\Psi\rangle = a|0\rangle + b|1\rangle$ al que se le aplica una rotación $R^z(\theta)$, con θ aleatorio, y representado con una distribución gaussiana con media 0 y varianza 2λ, entonces la respuesta al sistema se puede modelar con la matriz densidad [149]

$$\rho = \frac{1}{\sqrt{4\pi\lambda}} \int_{-\infty}^{\infty} R_z(\theta)|\Psi\rangle\langle\Psi|R_z^\dagger(\theta) e^{\frac{-\theta^2}{4d}} d\theta = \begin{pmatrix} |a|^2 & ab^* e^{-\lambda} \\ a^* b e^{-\lambda} & |b|^2 \end{pmatrix}. \tag{3.83}$$

CAPÍTULO 4

CRIPTOGRAFÍA EN EL ESPACIO DE ESTADO CUÁNTICO

La criptografía cuántica es un proceso de codificación, y decodificación de información que utiliza como base teórica las propiedades de la mecánica cuántica, se proyecta como el futuro de la seguridad ideal, ya que no está sujeta a las limitaciones computacionales del procesamiento clásico de la información. Los protocolos de criptografía cuántica permiten mantener la seguridad en la trasmisión de información que circulan en redes de comunicación, ya que sus modelos se basan en propiedades cuánticas de la materia y no en la incapacidad temporal de resolver problemas matemáticos complicados, que pueden también ser vulnerables por la potencia de cómputo cuántico ejemplificadas en los algoritmos de Shor y Grover. En este capítulo, se definen los algoritmos cuánticos que pueden vulnerar la criptografía clásica, y además se muestran los principales protocolos de transmisión de claves cuánticas que permiten mantener la seguridad de la información.

4.1. Algoritmos Cuánticos

Un algoritmo cuántico se puede modelar con una serie de circuitos formados por compuertas cuánticas que actúan sobre un conjunto de qubits de entrada, a los que se les realizan operaciones y mediciones para obtener una salida. Se pueden construir modelos como el oráculo de Hamilton [150], el retroceso de fase, la estimación de fase, la transformación cuántica de Fourier, las caminatas cuánticas, la amplificación de amplitud y teoría cuántica de campos topológicos [151]. En general se puede decir que un algoritmo cuántico es un modelo que ejecución de operaciones descritas por las propiedades de la mecánica cuántica, como la superposición o el entrelazamiento cuántico. Esta idea se fortalece a partir de los trabajos de Feynman [152] y Deutsch [153] en década de los ochenta, en donde los algoritmos empiezan a aprovechar los fenómenos cuánticos para mejorar la velocidad de procesamiento de información y reducir los tiempos de operación de exponenciales a polinomiales. Clásicamente el tiempo de descomposición de un números primo N cualquiera crece en forma logarítmica con respecto a su longitud, y el mejor algoritmo clásico para realizar estos procesos necesita computar $\mathcal{O}\Big(\exp\Big((69/9b)^{1/2}\log(b)^{2/3}\Big)\Big)$ operaciones [154], mientras que un algoritmo cuántico como el Shor puede descomponer en sus factores un número N primo cualesquiera en tiempos polinómico con un orden $\mathcal{O}\Big(\log(N)^3\Big)$, lo que permitiría quebrantar los sistemas criptográficos basados en el problema de la factorización como el sistema de clave pública RSA. Por otra parte, el algoritmo de Grover es otro de los algoritmos cuánticos importantes y utiliza el principio fundamental de superposición, para potenciar búsquedas sobre un conjunto de datos no estructurados. Por ejemplo, si se necesita encontrar un dato W especifico, realizando una búsqueda en un conjunto con N datos,

entonces con un algoritmo clásico se necesitan hacer una búsqueda como mínimo de $N/2$ operaciones, mientras que con un sistema cuántico y aprovechando la superposición de estados, este mismo problema se puede resolver examinando simultáneamente todas las posibles combinaciones. Por lo tanto, el proceso de búsqueda se puede reducir a un orden de $\mathcal{O}\left(\sqrt{N}\right)$ operaciones, hecho que puede vulnerar la criptografía simétrica y las funciones hash.

4.1.1. Algoritmo de Peter Shor

El algoritmo de Peter Shor utiliza la transformada cuántica de Fourier[1], la estimación de la fase[2], y el problema del orden[3] [155], para encontrar los factores primos n_1, n_2 de un número compuesto $N = n_1 n_2$ de donde $(n_1, n_2) = 1$, en este proceso se selecciona aleatoriamente un número x primo relativo con N tal que $(x, N) = 1$ y que satisfaga las igualdades $\exp\{k, N\} = x^k \mod N, \exp(k + r, N), x^r = 1 \mod N$, siendo r es el orden de la función $x \mod n$, si r es par se define una función $y = x^{r/2}$ solución de los sistemas de ecuaciones:

$$y_1 = 1 \mod n_1, 1 \mod n_2 \tag{4.1}$$
$$y_2 = -1 \mod n_1, -1 \mod n_2 \tag{4.2}$$
$$y_3 = 1 \mod n_1, -1 \mod n_2 \tag{4.3}$$
$$y_4 = -1 \mod n_1, 1 \mod n_2. \tag{4.4}$$

Del sistema de ecuaciones 4,1 y 4,2 se tiene la solución trivial $y_1 = 1, y_2 = -1$, la solución no triviales $y_3 = a, y_4 = -a$ se obtiene de sistema de ecuaciones 4,3, 4,4, luego se puede notar que por el teorema chino del residuo para un sistema con módulo comprimido $y = a_i \mod m_i$ y números pares se generan las soluciones $0 \leq y \leq m_1 m_2 ... m_i$, si r es par se tiene la solución $y = \pm a$, y con $a \neq 1, a \neq N - 1$ entonces se puede descir que $a + 1, a - 1$ tienen un divisor común con N, debido a que $a^2 = 1 \mod N$ y $a^2 = cN + 1$ con $c \in N$ siendo $a^2 - 1 = (a + 1)(a - 1) = cN$, de donde se puede encontrar que $N, a + 1$ y $N, a - 1$ son los factores de N, y se puede utilizando el algoritmo de Euclides en la siguiente expresión [156]

$$(a, b) = \begin{cases} b & si \quad a \mod b = 0 \\ (a, a \mod b) & si \quad a \mod b \neq a, para a > b \end{cases}, , \tag{4.5}$$

de donde para cada x aleatorio se obtiene una coincidencia con una probabilidad de $p = 1/2$ si $N \neq p^\alpha, o, N \neq 2p^\alpha$, pero si $N = p^\alpha, o, N = 2p^\alpha$ que representan las potencias puras de primos que no se pueden factorizar con algoritmos clásicos, pero se pueden construir algoritmos probabilísticos que encuentra un r en tiempos polinomiales mediante la definición de una función cuántica entera $F : |x, 0\rangle \longrightarrow F : |x, f(x)\rangle . f : Z \longrightarrow Z_{2^n}$ con un periodo $r \leq 2^n$, de donde para encontrar r se necesitan dos registros de tamaño $2n$ y m qubits que se inicializan en el estado $|0, 0\rangle$ y son aplicados al operador unitario U en los estados de la base, para generar una superposición homogénea $U|x, 0\rangle =$

[1]Teniendo un vector complejo $x_0, x_1, ..., x_{N-1}$ con N componentes, la trasformada discreta de Fourier operando sobre el vector y lo transforma en $y_0, y_1, ..., y_{N-1}$ vector complejo con N entrados.

[2]El problema de estimación de fase se reduce a calcular el valor de partiendo de un operador unitario U con auto estados $|u\rangle$ y autovalores $\exp\{2\pi\psi\}$ con norma 1. El algoritmo se divide en dos etapas: en la primera se prepara el estado $|u\rangle$ y se le aplica el operador unitario n veces. En la segunda etapa se aplica la transformada inversa de Fourier a un conjunto de t qubits al primer registro del circuito que inicia en el estado $|o\rangle$.

[3]El problema del orden aparece cuando se tiene dos números enteros (x, N) y se quiere encontrar el orden r de $x \mod n$. Clásicamente no se ha podido desarrollar un algoritmo capaz de resolver este problema, cuánticamente la estimación de fase desarrolla una idea que realiza $\mathcal{O}\left(L^3\right)$ operaciones con alta probabilidad de éxito, donde L corresponde a los bits necesarios para encontrar el orden.

$\sum_{i=0}^{N-1} c_1|1,0\rangle$ con $|c_1| = 1/\sqrt{N}$ y $N = 2^{2n}$ correspondientes a una transformación Hadamard, y si se opera $F|\psi\rangle = FH|0,0\rangle$ se obtiene $1/2^n = \sum_{i=0}^{N-1} |i,0\rangle = 1/2^n = \sum_{i=0}^{N-1} |i,f(x)\rangle$, y al medir en el segundo registro se obtiene $k = f(s)$ con $s < r$, lo que reduce el estado $|\psi^{'}\rangle = \sum_{i=0}^{N-1} c^j|rj+s,k\rangle$ con $c_j^{'} = |N/r^{-1/2}|$ que es el espectro homogéneo por todos los vectores de la base $|rj+s\rangle$ debido a que $f(rj+s) = f(s)$ para todo j.

Con la medición en el primer registro aparece un desfase S que no permite que se calcule r o uno de sus múltiplos, entonces para mantener la probabilidad espectral invariante con respecto al desfase S se utilizar la transformada cuántica discreta de Fourier $TDF : |x\rangle = 1/\sqrt{N}\sum_{y=0}^{N-1} \exp\{(2\pi i/N)xy\}$ aplicada al estado $|\psi^{'}\rangle$ se tiene $TDF : |\psi^{'}\rangle = \sum_{y=0}^{N-1} c_j^{'}\,|i,k\rangle$ con $c_j^{'} = \sqrt{r}/N\sum_{j=0}^{p-1}\exp\{(2\pi i/N)i(jr+s)\} = \sqrt{r}/N\sum_{j=0}^{p-1}\exp\{\phi_i\}\exp\{(2\pi i/w)ijk\}$ donde $\phi_i = 2\pi(s/N), P = |N/r|$ y si $N = 2^n$ múltiplo de r entonces $c_j^{'} = e_i^{\phi}/\sqrt{r}$, pero si i es múltiplo de N/r o tiene otra forma, el espectro de $|\psi^{'}\rangle$ tiene un periodo N/r dado que:

$$lim_{n\to\infty}(\sum_{n\longrightarrow\infty}^{N-1} e^{2\pi k\alpha}) = \begin{cases} 1 & si \quad \alpha \in \mathbb{Z} \\ \\ 0 & si \quad \alpha \notin \mathbb{Z} \end{cases}, \tag{4.6}$$

lo que garantiza 2^n elementos de orden $\mathcal{O}(1)$ para el primer registro con $2n$ qubits donde $r < 2^n$, por lo tanto la fase S se puede calcular con un circuito como el que se muestra en la figura 4.1 [154].

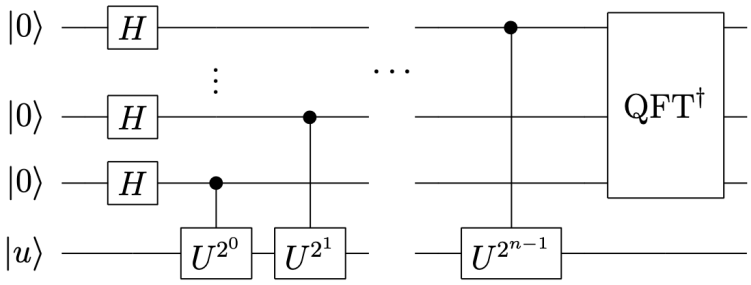

Figura. 4.1 Circuito cuántico para calcular la fase: Superposición uniforme de la forma $1/2^{t/2}\sum_{k=0}^{2^t-1}|k\rangle$ donde el operador U^{2^j} es $|j\rangle\,|U\rangle \longrightarrow |j\rangle\,U^j\,|k\rangle$, por lo tanto, la primera etapa queda descrita por $|0\rangle\,|U\rangle = 1/2^{t/2}\sum_{k=0}^{2^b-1}\exp\{2\pi i\phi k\}\,|k\rangle$, figura tomada del trabajo [157].

Con la transformada discreta de Fourier en el primer registro se obtiene c en la vecindad de $\lambda N/r$ con $\lambda \in \mathbb{Z}_r$ tal que $c/N = c * 2^{-2n} \cong \lambda/r$ aproximaciones racionales de la forma a/b con $a, b \leq 2^n$ para $c * 2^{-2n}$, de tal manera que λ y r se puedan determinar cómo $a/b = \lambda/r$ siempre y cuando se cumpla que $(\lambda, r) = 1$, de donde la probabilidad de que $\lambda + 1$ sea coprimo es mayor que $1/\ln r$, y por lo tanto se necesitan $\mathcal{O}(n)$ intentos para tener una probabilidad de éxito cercana a 1 [158]. La DTF en mecánica cuántica es una representación del momentum y la posición de una función de onda que tiene un vector f de N números complejos f_k con $k \in \{0, 1, \ldots, N\}$, es decir que la la transformada discreta de Fourier DFT representa el mata de estados de N números complejos a N números complejos donde los coeficientes de la transformada de Fourier son $\tilde{f}_j$ tal que [159].

$$\tilde{f}_j = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} w^{-jk} f_k, \tag{4.7}$$

donde $w = \exp\{2\pi i/N\}$, de donde la trasformada inversa de fourier se define como

$$f_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} w^{-jk} \tilde{f}_k. \tag{4.8}$$

Para implementar la transformada cuántica de Fourier QFT en un circuito se utiliza la compuerta Hadamard, que para $N = 2$ adopta la forma

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{4.9}$$

Aplica la Hadamard y al estado unitario del qubit $a_0 |0\rangle + a_1 |1\rangle$, se obtiene $1/\sqrt{2}(a_0 + a_1) |0\rangle + (a_0 + a_1) |1\rangle = \tilde{a}_0 |0\rangle + \tilde{a}_1 |1\rangle$, de donde se observa que la Hadamard realiza una operaciones de amplitud en los estados de la DFT con $N = 2$ que representan una nueva amplitud para la base computacional de la QFT que se puede escribir como:

$$\sum_{x=0}^{N-1} a_x |x\rangle \longrightarrow \sum_{x=0}^{N-1} \tilde{a}_x |x\rangle = \sum_{x=0}^{N-1} \frac{1}{\sqrt{N}} = \sum_{y=0}^{N-1} w_N^{-xy} a_y |x\rangle , \tag{4.10}$$

de donde se puede ver que la QFT realiza una transformacion de los estados de la base $|x\rangle = 1/\sqrt{N} \sum_{y=0}^{N-1} w_N^{-xy} |y\rangle$, por lo tanto la QFT se puede representar con la matriz unitaria

$$U_{QFT} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} w_N^{-xy} |y\rangle \langle x| , \tag{4.11}$$

La QFT permite realizar las operaciones en el algoritmo de Shor, y se puede implementar en un circuito cuántico con $N = 2^n$ que generar la siguiente transformación

$$|x\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} w_N^{-xy} |y\rangle , \tag{4.12}$$

de donde la suma se puede expandir como

$$|x\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_{y_1,y_2,\ldots,y_n \in \{0,1\}} w_N^{-x \sum_{k=0}^{n} y_k} |y_1, y_2, \ldots, y_n\rangle , \tag{4.13}$$

si se expande la suma como un producto de exponenciales, y se reorganizan los productos se tiene la expresión

$$|x\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \bigotimes_{k=0}^{n} \left[\sum_{y_k \in \{0,1\}} w_N^{-x 2^{n-k} y_k} |y_k\rangle \right], \tag{4.14}$$

y al expandir nuevamente la suma se puede reescribir $|x\rangle$ como

$$|x\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \bigotimes_{k=0}^{n} \left[|0\rangle + w_N^{-x 2^{n-k} y_k} |1\rangle \right], \tag{4.15}$$

pero como en la ecuación 4,15 $w_N^{-x 2^{n-k} y_k}$ no dependen de los bits de orden superior de x, por lo tanto, al utilizar la espacian $0.x_l x_{l+1} \ldots x_n = x_1/2 + x_{l+1}/4 + \ldots + x_n/2^{n-l+1}$, de donde $|x\rangle$ se puede

representar mediante

$$|x\rangle \longrightarrow \frac{1}{\sqrt{2^n}}\left[|0\rangle + e^{-2\pi i 0.x_n}\,|1\rangle\right] \oplus \left[|0\rangle + e^{-2\pi i 0.x_{n-1}x_n}\,|1\rangle\right] \oplus \ldots \oplus \left[|0\rangle + e^{-2\pi i 0.x_1 x_2 \ldots x_n}\,|1\rangle\right], \quad (4.16)$$

en la QFT para $N = 2^n$, el ultimo qubit depende de todos los qubit de la entrada, entonces si se toma el primer qubit como $|x_1 \ldots x_n\rangle$, y se aplica una operación Hadamard se produce la siguiente transformación

$$|x\rangle \longrightarrow \frac{1}{\sqrt{2}}\left[|0\rangle + e^{-2\pi i 0.x_1}\,|1\rangle\right] \oplus |x_2, x_3, \ldots, x_n\rangle\,. \quad (4.17)$$

Ahora, utilizando una compuerta cuántica de rotación

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\{(-2\pi i/2)\} \end{bmatrix}, \quad (4.18)$$

que representa una operación controlada que se le puede aplicar a los qubits R_2, R_3 para que generen la siguiente transformación

$$|x\rangle \longrightarrow \frac{1}{\sqrt{2}}\left[|0\rangle + e^{-2\pi i 0.x_1 x_2 \ldots x_n}\,|1\rangle\right] \oplus |x_2, x_3, \ldots, x_n\rangle\,, \quad (4.19)$$

expresión que representa el último término del estado de la QFT, y si se realiza una Hadamard a la compuerta controlada R_k se obtiene el penúltimo qubits, y al final este proceso la transformación [159]

$$|x\rangle \longrightarrow \frac{1}{\sqrt{2}}\left[|0\rangle + e^{-2\pi i 0.x_1 x_2 \ldots x_n}\,|1\rangle\right] \oplus \left[|0\rangle + e^{-2\pi i 0.x_1 x_2 \ldots x_{n-1}}\,|1\rangle\right] \oplus \ldots \oplus \left[|0\rangle + e^{-2\pi i 0.x_n}\,|1\rangle\right], \quad (4.20)$$

de donde se obtiene la QFT, Y en la figura 4.2 se muestra el circuito.

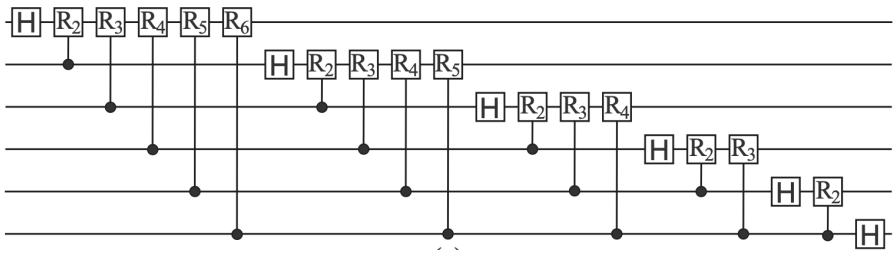

Figura. 4.2 Circuito para calcular la transformada cuántica de Fourier: En la figura se muestra el circuito cuántico que calcular la transformada cuántica de Fourier que se comporta como un operador unitario, y H es el espacio de estados de los n qubit con $N = 2^n - 1$ y $j = (j_1, j_2 \ldots j_n) = j_1 2^{n-1} +_1 2^{n-2} + \ldots +_1 2^0$ y $j_k \in 0, 1$, para $k \in |1, n|$ de donde $|j\rangle \longleftrightarrow |j_1, j_2 \ldots j_n\rangle \in H$, por tanto TF opera en la base $0 * j_1 j_2 \ldots j_n = j_1 2^{-1} + j_2 2^{-2} + \ldots + j_k 2^{-k}$, entonces $TF\,|j\rangle = (|0\rangle + \exp\{2\pi i(0 * j_n)\}\,|1\rangle)(|0\rangle + \exp\{2\pi i(0 * j_n) - 1 * j_n\}\,|1\rangle) \ldots (|0\rangle + \exp\{2\pi i 0 * j_1 j_2 \ldots j_n\}\,|1\rangle)/(2^{n/2})$. Figura tomada de [160].

El algoritmo cuántico de Shor puede resolver el problema de la factorización y el problema del logaritmo discreto, que son los encargados de brindar seguridad a la criptografía asimétrica. Este algoritmo encuentra los factores de un número con alta probabilidad de acierto utilizando la QFT, pero hasta el momento y por motivos tecnológicos solo se han podido implementar un número pequeño de qubit para realizar cómputos cuánticos, lo que a echo posible solo la factorización de números pequeños, es decir, que si se quiere factorizar un número N, se escoge un n tola que $N^2 \leq n \leq 2N^2$,

m tal que $N \leq m \leq 2N$, y $Q = 2^n$, entonces para implementar el algoritmo de Shor se desarrolla una secuencia como la siguiente:

- Se encojé un entero aleatorio a perteneciente a $[1, n-1]$.
- Si $\quad$ mód $(a, N) \neq 1$, se vuelve a calcular $\quad$ mód (a, N).
- Se determina el perdió T de la función $f(k) = a^k \mod N$.
  - Se inicializa el sistema en (n, m) qubit, $|0\rangle \otimes |0\rangle$
  - Se aplica la QFT, F_n al primer registro.
  - Se aplica el operador U_f al asociada a la función f.
  - Se aplica nuevamente F_n al primer registro.
  - Se obtiene la medida ky, y se calcula la función continua k/Q.
  - Se toma como posible valor de de T los denominadores los resultados convergentes de la función continua.
- Para cada T se realizan las siguientes operaciones:
  - Si T es impar el algoritmo arroja un follo.
  - Si T es par y $\quad$ mód $(a^{t/2} + 1, N) \neq N$ se vuelve a calcular $\quad$ mód $(a^{t/2} + 1, N)$.
  - Para cualquier otro caso se obtiene un fallo.

4.1.2. Algoritmo de Grover

El algoritmo cuántico de Grover es un problema de búsqueda no estructurada desarrollado en 1996, por Lov K. Grover. La idea principal del problema es hallar x en un conjunto de posibles soluciones tal que la hipótesis $f(x) = 1$ sea cierta. Este algoritmo reduce los tiempos de búsqueda en una lista desordenada en $\mathcal{O}\left(\sqrt{N}\right)$ evaluaciones de f, pues clásicamente para un espacio de búsqueda de tamaño N, se necesita evaluar a f un promedio de $N/2$ veces y en el peor de los casos N veces, estos algoritmos clásicos de búsqueda no estructurada requieren por lo tanto $\mathcal{O}(N)$ evaluaciones de f. Es decir que para un N grande con el algoritmo de Grover el tiempo decrece significativamente, por ejemplo, si $N = 10^6$ clásicamente se necesitan realizar como mínimo 500×10^3 búsquedas, mientras que cuánticamente solo se necesitan 10^3 búsquedas para obtener el mismo resultado. Esta aceleración del algoritmo cuántico se debe a que el estado inicial es una superposición uniforme de todos los $N = 2^n$ en un problema que tiene M soluciones con $1 \leq M \leq N$, de donde la superposición uniforme de estados se escribe como $|\phi\rangle = 1/\sqrt{(}N(|0\rangle + \ldots + |q1\rangle + \ldots + |q_M\rangle + \ldots + |N-1\rangle)$ de los cuales q_1, q_M representan los estados mezclados del sistema, y para el resultado sea mas efectivo se repiten subrutinas, [161].

Para utilizar el algoritmo de Grover, se tiene una lista de de tamaño N que puede tener un incremento de $N = 2^N$, y se busca un x ente $[0, 2^n - 1]$ tal que $f(x) = 1$, este problema se puede realizar con computación cuántica, ya que esta teoría permite evaluar una función f simultáneamente sobre todas las posibles entradas utilizando una base otonormal B_n, que se obtiene a partir del estado $|0\rangle$ y aplicándole la trasformación de Wolsh-Hadamard se obtiene la expresión.

$$\psi = W_n(|0\rangle) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \,, \tag{4.21}$$

que representa el incremento de la amplitudes en la solución, y disminuye los x que no verifican la igualdad $f(x) = 1$, para cuando se mida el registro de lo resultados se tenga una probabilidad de acierto. el algoritmo de Grover se puede implementar en tres etapas operacionales:

- Partiendo de una lista $N = 2^n$ datos, con $x = 0 \ldots N-1$, de tal manera que solo un de las datos verifica la condición $f(x) = 1$, entonces de construye el estado de superposición de todas las palabras de n bits como $W_n(|0\rangle)$.
- Oráculo: se da un cambio de signa en la amplitud de los x tales que $f(x) = 1$ al aplicar $U(|x\rangle) = (-1)^{f(x)} |x\rangle$, que se implementa con $U_f(|x\rangle \otimes |b\rangle) = |x\rangle \otimes |b\rangle \otimes |f(x)\rangle$ con $b = 1/\sqrt{2}(|0\rangle - |1\rangle)$.
- Inserción sobre el promedio: si A es el promedio de las operaciones en las amplitudes se realizan las trasformaciones $\sum_{x=0}^{N-1} a_x |x\rangle \longrightarrow \sum_{x=0}^{N-1} (2A - a_x) |x\rangle$ con el operador G.

$$G = \begin{pmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \cdots & \frac{2}{N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} - 1 \end{pmatrix}. \tag{4.22}$$

Por ejemplo si S el número de búsqueda y se han realizado k iteraciones, donde el estado inicial $W_n(|0\rangle)$ se a modificada con el cambio de las amplitudes de cada estado $|x\rangle$, por lo tanto para todos los estados $|x\rangle$ con $x \neq S$ se tendrá una amplitud m_k, y para el estado$|S\rangle$ se tendrá un amplitud b_k de tal manera que el estado resultante se puede escribir como:

$$b_k |S\rangle + m_k \sum_{k \neq S} |x\rangle, \tag{4.23}$$

de donde las amplitudes serán.

$$\begin{cases} m_0 = \frac{1}{\sqrt{N}} & b_0 = \frac{1}{\sqrt{N}} \\ m_{k+1} = 2A_k - m_k & b_{k+1} = 2A_k + b_k \end{cases} \quad con, A_k = \frac{(N-1)m_k - b_k}{N}, \tag{4.24}$$

$$\begin{bmatrix} m_{k+1} \\ b_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{N-2}{N} & \frac{-2}{N} \\ \frac{2N-2}{N} & \frac{N-2}{N} \end{bmatrix} \begin{bmatrix} m_k \\ b_k \end{bmatrix} \tag{4.25}$$

por lo tanto cuando se mide el estado después de las k iteraciones, la probabilidad de obtener S es $|b_k|^2$ y la probabilidad de fallo es $(N-1)|m_k|^2$, entonces con un número de iteraciones adecuado se llega a que la probabilidad de fallo es menor que $1/N$ [162].

Con los desarrollos de Shor, Grover y la velocidad de procesamiento de la computación cuántica se demuestra que es posible vulnerar los algoritmos clásicos de criptografía, y se genera la necesidad de implementar nuevas técnicas para proteger la información, como la criptografía cuántica que depende de las propiedades físicas de la materia, el entrelazamiento, la teleportación, y la superposición de los estados cuánticos, donde un estado desconocido no se puede ser copiado, clonado o medido sin perturbarlo (teorema de no clonación) [163].

4.2. Protocolos Criptográficos Cuánticos

La criptografía cuántica surge como una respuesta a la velocidad de procesamiento de la computación cuántica, que es la implementación física de la información pasar del bit al qubit, el cual

puede representar una superposición de estados $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ descritos por las propiedades de la mecánica cuántica. Con esto, la criptografía cuántica se puede definir como un conjunto de técnicas que aprovecha las propiedades cuánticas de la materia para transmitir claves criptográficas a través de canales como fibra óptica o espacio libre; en el proceso de generación de claves se utiliza una secuencia aleatoria de fotones polarizados o entrelazados en donde se codifican los bits clásicos 0 o 1, y se involucran los actores activos de un sistema de comunicación donde Alice es el transmisor y Bob el receptor como se describe a continuación [164].

- Alice y Bob acuerdan una polarización de espín (x, z) para realizar las mediciones.
- Alice controla la fuente S y prepara N estados de espín $N \gg n$ con n el número de bits del mensaje, y le envía a Bob la secuencia b.
- Para cada espín enviado que son eventos no correlacionados, Alice y Bob miden de forma aleatoria el estado y registran la polarización z o x, que tiene una probabilidad de ocurrencia $p = 1/2$.
- Bob de forma aleatoria escoge un subconjunto del conjunto de estados medidos, y se comunica con Alice utilizando un canal inseguro para mostrar el resultado de las medidas realizadas.
- Alice compara sus resultados con los enviados por Bob, para saber las coincidencias que se tienen con las medidas hechas por.
- Bob, en este punto si las medidas coinciden el proceso para, pero si no, entonces Alice polariza nuevamente una secuencia de espín y se repite el proceso de medición.
- Alice comunica por el canal inseguro las medidas que ha tenido éxito, y luego Bob comunica todos sus resultados con los ejes de polarización, pero no comunica la correspondencia de los espín para generar una clave segura .

La seguridad en términos de la protección de la información con estas técnicas, se fundamenta en la capacidad de Alice y Bob para detectar a un intruso Eve; por ejemplo si se supone que Eve quiere interceptar la clave, está tiene que realiza mediciones sobre los estados de los fotones ya transmitidos, y como se están utilizando propiedades de la mecánica cuántica, al realizar un medición se introducen errores en el sistema por el teorema de no clonación, que serían detectados por Alice y Bob como un error por fuera de umbral aceptado por el sistema de transmisión de claves, y por lo tanto se abortaría la comunicación [165, 166]. Con las anteriores características de han desarrollada bario protocolos de criptografía cuántica que permiten generar claves para cifrar información de modo cuántico.

4.2.1. Protocolo BB84

El protocolo BB84, fue desarrollado en 1984 por Charles H. Bennett y Gilles Brassard, en este protocolo la generación y trasmisión de la clave utiliza cuatro estados cuánticos, horizontal $|\leftrightarrow\rangle$, vertical $|\updownarrow\rangle$, oblicuo a derecha $|\nearrow\rangle$, oblicuo a izquierda $|\nwarrow\rangle$, y dos bases de polarización $+$ y X de tal manera que Alice puede enviar a Bob un cero 0 o un 1 de forma aleatoria codificado en cualquiera de los estados $|\leftrightarrow\rangle, |\updownarrow\rangle, |\nearrow\rangle, |\nwarrow\rangle$, por medio de un canal cuántico que puede ser una fibra óptica o espacio libre. Cuando Alice envía un 0 codificado en el estado horizontal del sistema $|\leftrightarrow\rangle$, Bob pude medir el estado cuántico $|\leftrightarrow\rangle$ en la base $+$, o el estado cuántico $|\nearrow\rangle$ en la base X, similarmente, si Alice envía un 1 codificado en el estado $|\updownarrow\rangle$, Bob puede medir el estado $|\leftrightarrow\rangle$ o $|\updownarrow\rangle$ dependiendo de cuál base utilice $+$ o X. Después que Bob mide toda la cadena de estados enviada se comunica con Alice utilizando un canal clásico y le dice en qué bases midió cada una de los estados, estos comparan

las mediciones realizadas y escogen los estados que son iguales para generar la clave de cifrado [167]. luego de haber comparado las mediciones hechas por Alice y Bob están en el umbral aceptable de error del sistema, entonces se tienen los bits que forman la clave para cifrar y descifrar los mensaje, utilizando una técnica de cifrado de información de un único uso, Alice y Bob pueden codificar un mensaje y enviarlo a través de un canal clásico inseguro con alta seguridad, como se muestra en la figura 4.3 [168, 169].

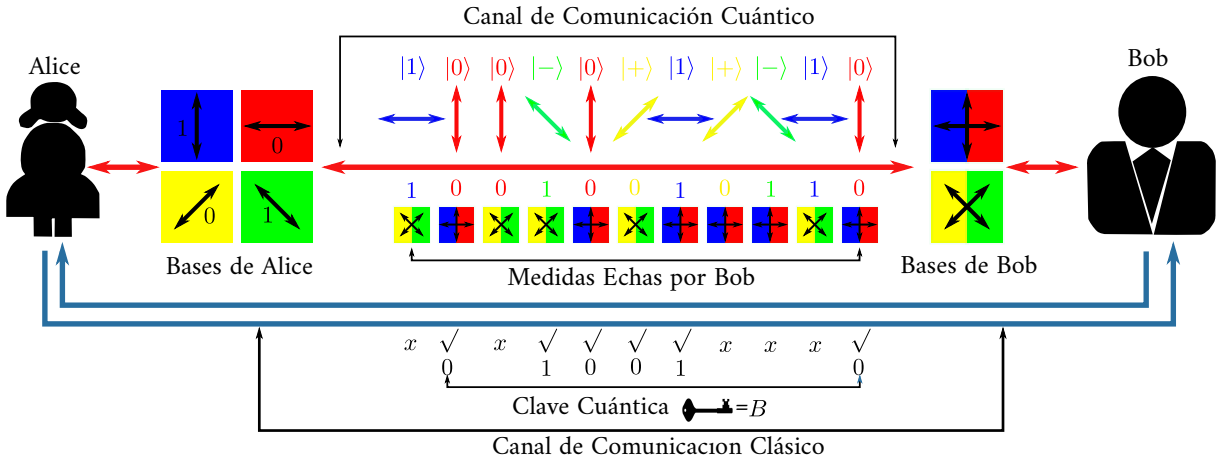

Figura. 4.3 Protocolo BB84: En la figura se muestra una implementación del protocolo de transmisión de claves cuánticas BB84, el cual consiste en enviar polarizaciones lineales de 0°, 45°, 90° y 135° desde el transmisor Alice, al receptor Bob utilizando una fibra óptica como canal de comunicación por la la modulación de polarización, efecto que modifica los estados trasmitidos. Por lo tanto, antes de que Bob realice las midas, los estados se deben transformar al sistema de coordenadas original, donde se debe controlar el cambio de fase entre el estado polarizado y su ortogonal. Después de la transmisión y medición de los estados cuánticas, Bob se comunica con Alice para compara los bits y los iguales formaran la clave cuántica.

4.2.2. Protocolo K05

En el protocolo K05, Alice tiene la posibilidad de codificar más de dos estados no ortogonales en cada uno de los bits 1 o 0, construidos a partir de un subconjuntos de estados de polarización, por lo tanto K05 es una generalización del protocolo criptográfico BB84, funciona mendicante la generación, trasmisión y codificación de cadenas binarias estructuradas de la siguiente manera.

- Alice prepara cadenas de bits 00110101 y la pasa a través de un filtro de polarización aleatorio. En ese proceso, la polarización de estados se representa con 0 y 1 que solo son conocidos por Alice y desconocidos por cualquier otro, incluido Bob.

- Bob recibe la cadena de fotones polarizados los cuales pasan por su filtro de polarización para ser medidos de forma independiente y aleatoria, en donde la polarización aleatoria del filtro deja pasar o rechaza los fotones recibidos, con lo que se genera una cadena binaria nueva de 0 y 1, que tiene algunos bits con el valor lógico correcto de la cadena generada por Alice, y como Bob ni Alice conoce la asociación entre el valor lógico y el estado de polarización, y tampoco Bob saben que bits son comunes en las cadenas binarias.

- Después que Bob realiza las medidas pertinentes, se comunica con Alice por un canal público (llamada telefónica) y le dice la secuencia de polarización que ha usado mientras recibía los

fotones polarizados de Alice. Pero Bob no le revela a Alice la secuencia lógica que ha generado, como se muestra en la figura 4.4 [170].

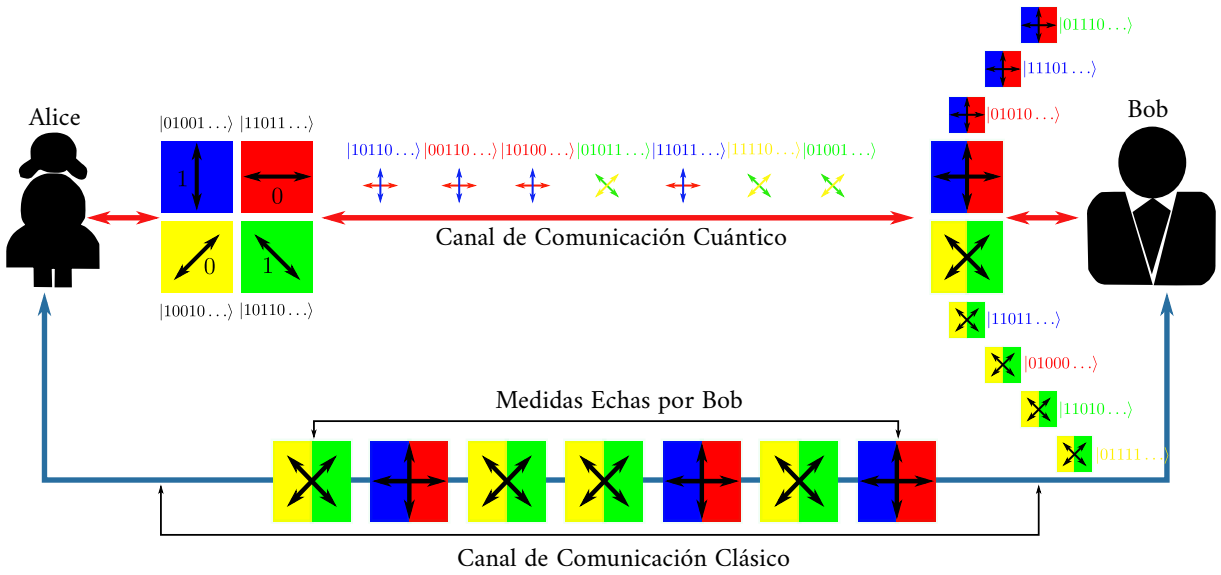

Figura. 4.4 Construcción y transmisión de cadenas binarias entre Alice y Bob: Alice genera sus cadenas Binarias y las envía utilizando una canal cuántico de comunicación, Bob recibe las cadenas y realiza mediciones de la polarización pueda construir una nueva cadena binaria que Alice no conoce.

Alice pasa la cadena lógica que envió a Bob por la secuencia de polarización que Bob midió; con esto Alice compara la cadena inicial de bits con la generada desde el experimento e identifica los bits que son comunes en las dos cadenas, luego Alice le comparte a Bob los filtros de polarización que se usaron correctamente en la secuencia, pero sin decirle su asociación con la codificación 1 y 0, los estados de polarización que se usaron correctamente forman la clave cuántica, como se muestra en la figura 4.5 [171].

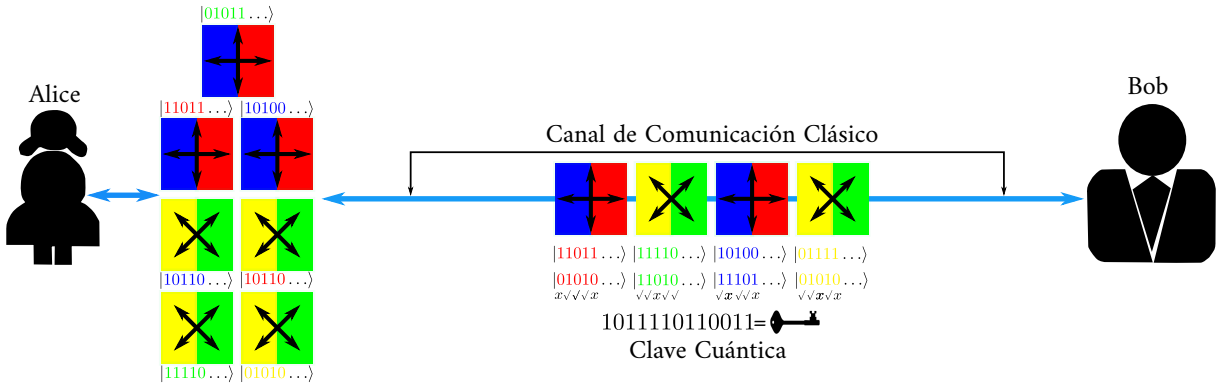

Figura. 4.5 Verificación de los estados generados por Alice y medidos por Bob: En la figura se muestra cómo Alice y Bob compara sus cadenas binarias independientes para construir la clave criptográfica, para este proceso se utiliza un canal clásico de comunicación.

Después que Alice y Bob tiene la clave criptográfica definida, se puede codificar un mensaje y enviarlo a través de un canal clásico inseguro, en la figura 4.6 se muestra el algoritmo cuántico de cifrado de información K05.

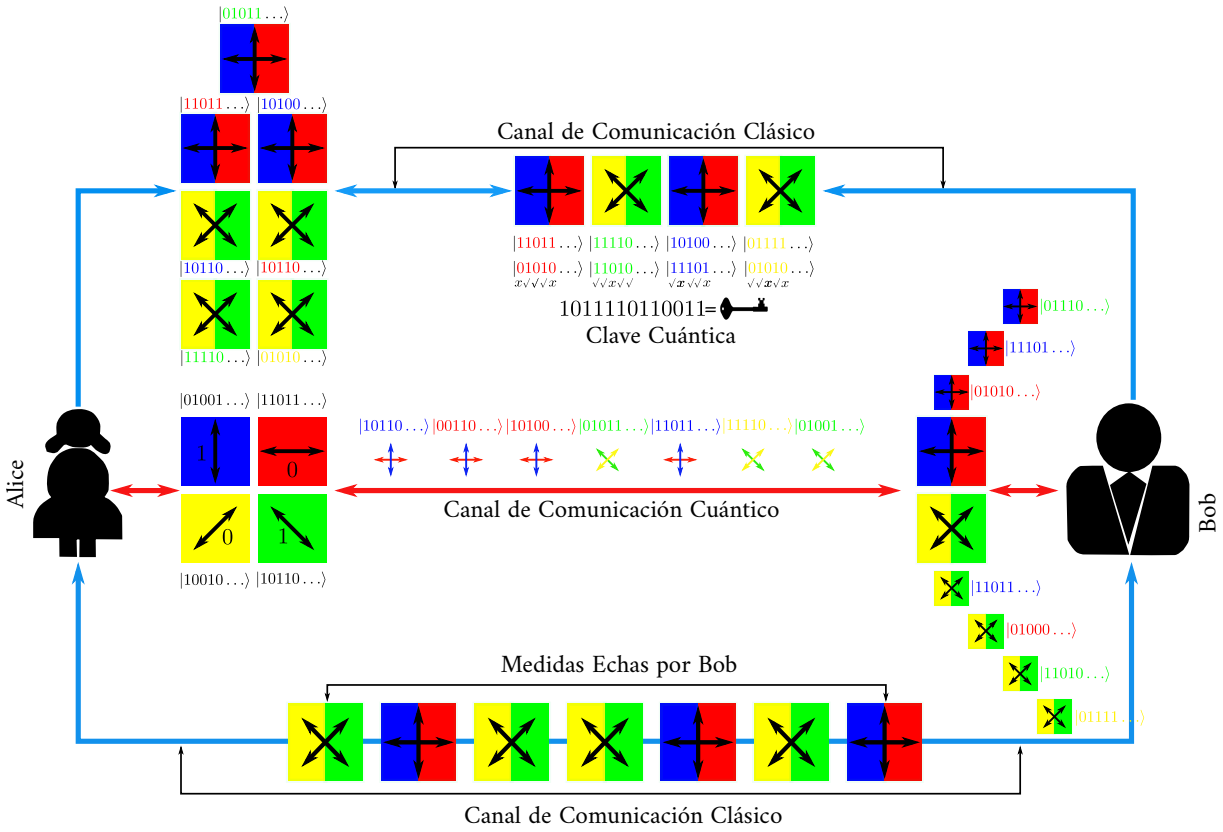

Figura. 4.6 Protocolo K05: En la figura se muestra el proceso completo de de generación y transmisión de claves cuántica utilizando el protocolo K05, se especifica como Alice y Bob construyen las cadenas binaria de fotones y miden la polarización para obtener los bits correctos de la clave cuántica.

4.2.3. Protocolo SARG04

El protocolo SARG04, es un modificación del protocolo BB84 echa en el 2004 por Valerio Scarani, Antonio Acín, Grégoire Ribord y Nicolas Gisin, la diferencia con respecto a BB84 se presenta cuando Alice y Bob realizan la comparación de resultados de las mediciones echas sobre los estados, para este proceso Alice le comunica a Bob utilizando un canal clásico, uno de los estados no ortogonales preparados $F(x,y) = |\psi_{+x}\rangle + |\psi_{-y}\rangle$, Bob interpreta el estado con una lógica opuesta a la que tiene la medida enviada. Por ejemplo, si Alice envía el estado $|\psi_{+1}\rangle$ a Bob por el canal cuántico y por el canal clásico envía $F_{1,1}$, entonces si Bob mide con la base B_+ el estado que obtiene es $|\psi_{+1}\rangle$ que coincide con el Alice envió, con este resultado no se puede concluir nada, si Bob mide el estado enviado con la base B_x obtiene el estado $|\psi_{x1}\rangle$ que no da ninguna información de la base que Alice utilizo, por lo tanto el resultado es inconsistente, pero si Bob mide con la base contraria a Alice y obtiene un estado diferente al que Alice envió por el canal clásico $|\psi_{x0}\rangle$, que sucede un 25 % de las veces, entonces Bob sabe con certeza que la base no es la correcta y por lo tanto la base que no se utiliza en la correcta para medir el estado que Alice envió, es decir que [172].

- Alice genera dos secuencias aleatorias de la misma longitud n, una con las bases y otra de valores codificados, luego Alice envía el estado asociado a Bob utilizando un canal cuántico.

- Bob genera una secuencia aleatoria con las base del mismo longitud n y mide los estados.

- Alice le comunica a Bob un par de estados correspondientes a una base.
- Bob interpreta el para de estados envidos teniendo presente dos condiciones: 1) si la medida hecha por Bob da como resultado el mismo estado que Alice público, el estado debe ser descartado, 2) si la medida hecha por Bob da diferente al estado publicado por Alice, entonces Bob tiene la seguridad de que la base utilizada es incorrecta, por lo tanto la otra base es la correcta para medir el estado enviado por Alice [173, 174].

4.2.4. Protocolo B92

El protocolo B92, modifica el protocolo BB84, se pasó de utilizar cuatro estados de polarización a utilizar dos sistemas de polarización diferentes, uno vertical-horizontal y otro diagonal izquierda-derecha. Para generar la clave criptográfica Alice y Bob preparan y guardan cada uno de forma aleatoria una cadena de bits conformada por unos 1 y ceros 0, luego se establece una comunican entre Alice y Bob utilizando un canal cuántico que puede ser fibra óptica o espacio libre y un canal clásico, llamada telefónica o una red de computadoras. Con el sistema construido Alice, de su cadena de binaria ya generada le envía a Bob una serie de fotones polarizados, donde el estado de polarización vertical representa el cero 0 y estado +45 grados representa el uno 1. Bob tiene para realizar mediciones en los estados dos polarizadores, uno horizontal que representa el uno 1 y otro en −45 grados que representa el cero 0 de la cadena binaria ya construida por Bob; por lo tanto, cada Bob recibe un fotón polarizado realiza una medición con uno de los dos polarizadores y registra el valor medido, si pasó o no como se muestra en la figura 4.7.

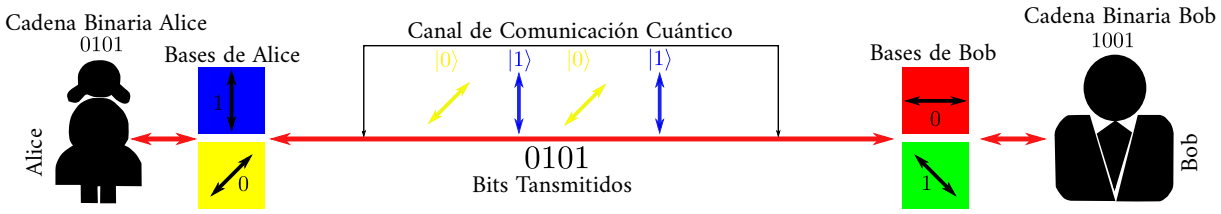

Figura. 4.7 Protocolo B92, inicio de la trasmisión de estados cuántico: Alice envía de forma aleatoria una serie de fotones polarizados en una sus dos bases +45° o 90° vertical, para que Bob realiza las mediciones y se pueda generar la clave cuántica de cifrado.

Si Alice y Bob coinciden con el mismo polarizador vertical-horizontal o diagonal, el fotón recibido por Bob no pasa y se registra un no, ya que los estados polarizados por Alice siempre van a ser perpendiculares a los estados de Bob, mientras que los polarizadores diferentes forman un ángulo de 45 grados, y por la incertidumbre cuántica el 50 % de los fotones se comportan como si estuvieran polarizados paralelamente y pueden atravesar el polarizador con que Bob realizó la medida, el otro 50 % de los fotones no pasan los polarizadores porque tiene comportamiento de polarización perpendicular. Después que se termina la medición de los fotones polarizados Bob le envía a Alice los aciertos y fallos si o no, utilizando un canal clásico de comunicación y sin mostrar la polarización que Bob aplicó en cado un de los bits medidos, luego Alice y Bob guardan la secuencia binaria con los aciertos si que son los bits que van a generar la clave criptográfica. En este protocolo el promedio de bits enviados con éxito es del 25 % con errores del 1,6 % generados por las imperfecciones de los canales óptico y el ruido de los detectores [98]. En figura 4.8se muestra los fotones que pueden atraviesan los polarizadores de Bob, son aquellos en donde los polarizadores de Alice y Bob forman un ángulo de 45 grados, por lo tanto, existen cuatro posibilidades con el 25 % de certeza.

- Alice fotón polarizado en +45 graods y Bob con polarización en −45 grados, en esta posición

el fotón nuca atraviesa los polarizadores de Bob.

- Alice fotón polarizado en vertical y Bob con polarización horizontal, en esta posición el fotón nuca atraviesa los polarizadores de Bob.
- Alice fotón polarizado en +45 grados y Bob con polarización en horizontal, en esta posición el 50 % de los fotones atraviesan los polarizadores de Bob, lo que corresponde a un 12 % de éxito y el otro 50 % no atraviesan los polarizadores que también corresponden con un 12 % de fracaso.
- Alice fotón polarizado en vertical y Bob con polarización en -45 grados, en esta posición el 50 % de los fotones atraviesan los polarizadores de Bob, lo que corresponde a un 12 % de éxito y el otro 50 % no atraviesan los polarizadores que también corresponden con un 12 % de fracaso. Con 3 y 4 se tienen los bits que generan la clave criptográfica después que Bob comparte los resultados de las mediciones con Alice mediante un canal clásico inseguro, en la figura 4.8 se muestra la generación de y transmicion de clave cuántica utilizando el protocolo B92, [175].

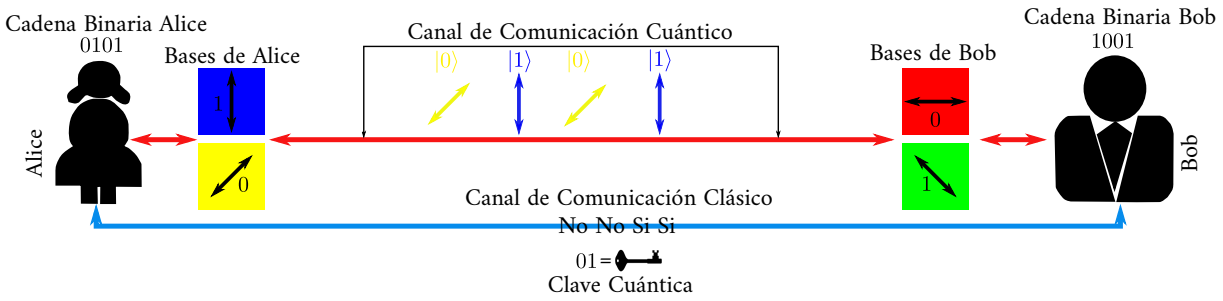

Figura. 4.8 Protocolo B92: En este protocolo Alice prepara y envía una cadena de qubits preparados individualmente al azar, en dodnde el estado $|0\rangle$ representa el valor de bit 0, o $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ que representa el valor de bit 1, luego Bob mide cada qubit entrante aleatoriamente $\{|0\rangle, |1\rangle\}$, con las bases diagonal $\{|+\rangle, |-\rangle\}$, y designa el bit medido como 0(1) si su resultado de medición es $|-\rangle$ $(|1\rangle)$, y para los resultados de medición $|0\rangle$ o $|+\rangle$, lo resultado no serán concluyente y descartan. Una parte de los bits concluyentes se utilizan para verificar si hay espía en el canal de comunicación y el resto son para generar la clave de cifrado.

4.2.5. Protocolo E91

El protocolo E91, es uno protocolos de criptografía cuántica basado en estados entrelazados, fue propuesto por Artur Ekert en 1991, utiliza estados Bell emitidos por una fuente común (SPDC) y distribuidos entre Alice y Bob, que utilizan bases de polarización elegidas al azar, donde Alice interpreta los estados H, D como 0 y los estados V, A como 1, mientras que Bob debe interpretar los estados en forma contrario para obtener la misma clave, por ejemplo, si se tiene el estado $|\psi^-\rangle$ entonces se tendrá la misma clave, porque $|\psi^-\rangle = 1/\sqrt{2}\{|H\rangle_A |V\rangle_B - |V\rangle_A |H\rangle_B\}$, pero con la base diagonal se obtiene que $|H\rangle_A = a_H^\dagger |0\rangle = 1/\sqrt{2}(a_D^\dagger + a_A^\dagger)|0\rangle = 1/\sqrt{2}(|D\rangle_A + |A\rangle_A)$ y $|V\rangle_A = a_H^\dagger |0\rangle = 1/\sqrt{2}(a_D^\dagger - a_A^\dagger)|0\rangle = 1/\sqrt{2}(|D\rangle_A - |A\rangle_A)$. Se realiza el mismo para el fotón B, y por lo tanto, $|\psi^-\rangle$ se puede escribir como $|\psi^-\rangle = 1/2\sqrt{2}\{(|D\rangle_A + |A\rangle_A)(|D\rangle_B - |A\rangle_B) - (|D\rangle_A - |A\rangle_A)(|D\rangle_B + |A\rangle_B)\}$, de donde $|\psi^-\rangle = \{|A\rangle_A |D\rangle_B - |D\rangle_A |A\rangle_B\}$ y como $|\psi^-\rangle$ no varia para ninguna transformación de las bases, entoces para cualquier base X, Alice y Bob detectarán estados polarizados ortogonales. Las mediciones que realiza Alice y Bob se registran en dos secuencias que contienen, una el par entrelazado y la otra la base, cuando este proceso finaliza Alice y Bob se comunican utilizando un canal clásico, comparan los resultados y descartan aquellos pares entrelazados medidos con bases

diferentes y los que tienen igual medida son la clave criptográfica, por ultimo para comprobar si hay un intruso en el canal Alice y Bob prueban las desigualdades de Bell, tal como se muestra en la figura 4.9 [25].

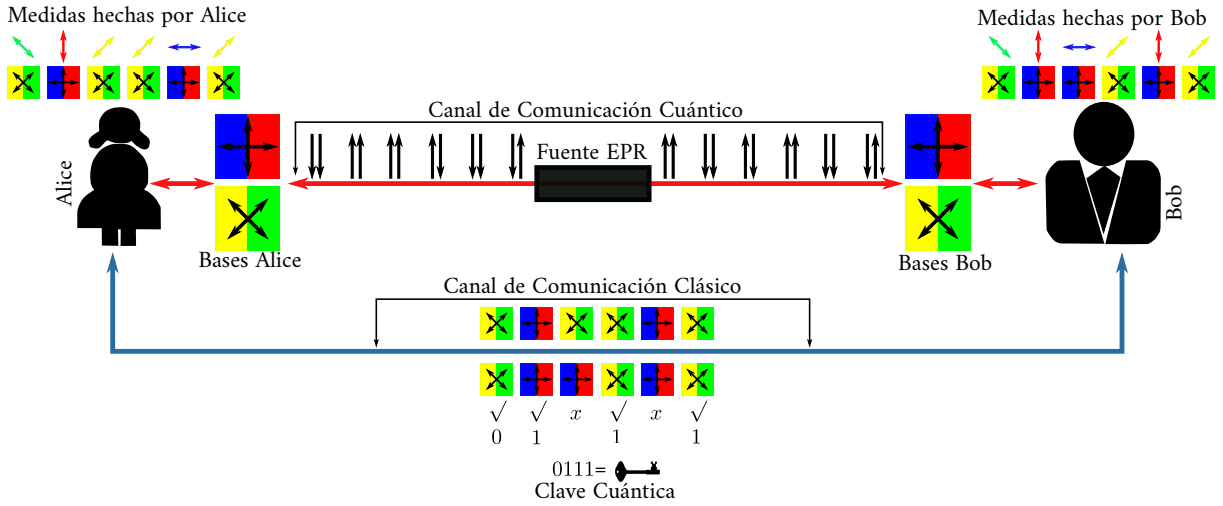

Figura. 4.9 Protocolo E91: Una fuente de pares entrelazados envía fotones a Alice y Bob, que son medidos con una de las bases preparadas, Alice y Bob se comparten los resultados de las mediciones utilizando un canal clásico, los resultados diferentes se descartan, los iguales generan la clave.

4.2.6. Protocolo coherente unidireccional COW

Protocolo COW, (Coherent One-Way protocol) fue desarrollado por Nicolas Gisin en el 2004 [176], los bits lógicos se codifican a tiempo, en una secuencia de pulsos coherentes débiles adaptados por un láser CW con un modulador de intensidad externo. Por lo tanta Alice puede codificar utilizando intervalos de tiempo T que contienen pulsos 0, sin luz o pulsos μ, con un número medio de fotones de $\mu < 1$, en este proceso el bit lógico $0_L(1_L)$ corresponden a una secuencia $0 - \mu(\mu - 0)$. Por mantener la seguridad del protocolo se envían secuencias de señuelo $\mu - \mu$. Bob, registra el tiempo de llegada de los fotones en el detector D_B para la línea de datos y para el monitoreo utiliza el detector D_M. de la medición de los tiempos en el detector D_B se tiene la clave sin procesar de la que Alice y Bob extraen la clave de cifrado. La seguridad se garantiza con la verificación de la detección en D_M. La secuencias señuelo y las secuencias lógicas 1_L0_L, utilizan un interferómetro no balanceado que tiene una diferencia de longitud T, se usa para estimar la información del espía y no Introducir errores en la clave, el funcionamiento de este protocolo se describe mendicante la siguiente secuencia [177].

- Alice prepara de forma aleatoria dos secuencias binarias conformadas por 0 y 1 que le envía a Bob, la primera secuencia tiene una probabilidad de $(1 - f)/2$ y la segunda es una secuencia señuelo con probabilidad f para garantizar la seguridad de la transmisión.

- Bob genera una clave sin procesar realizando medidas de tiempo en el detector D_B, monitoreando el detector D_{M1} para estar seguro que no hay intrusos en el canal.

- Bob muestra clásicamente los bit cuando las medidas del detector D_B coinciden con los tiempos del detector de monitoreo D_{M1}.

- Con el tiempo de detección del detector de monitoreo D_{M2}, Alice verifica las dos secuencias generada por ella y la secuencia en la salida del interferómetro; si hay un espía en el canal la correlación entre los dos pulsos se destruye y el intruso será detectado.

- Por otra parte si se comprueba que el canal no hay intrusos, Alice le comunica a Bob cuales son los bit señuelo debe eliminar de la clave sin procesar.

- Por último Alice y Bob genera un clave compartida teniendo en cuenta la corrección de errores y la privacidad, como se muestra en la figura 4.10 [178].

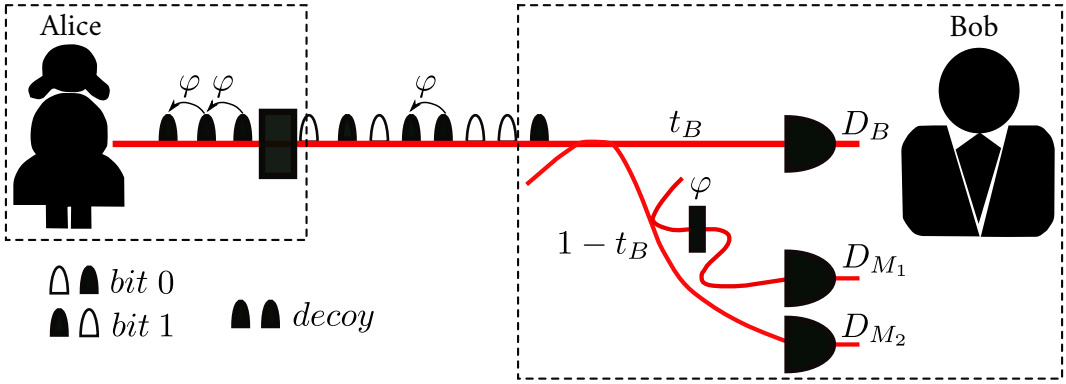

Figura. 4.10 Protocolo COW: En la figura se muestra una medición, donde Bob puede leer la clave sin procesar en el detector D_B, además, tiene un interruptor para enviar algunos pares de pulsos consecutivos a una línea de monitoreo que examina la coherencia entre los pulsos pares e impares enviados por Alice, figura modificada del trabajo [178].

4.2.7. Protocolo Distribución de Clave Cuántica de Cambio de fase Diferencial DPS-QKD

El protocolo DPS-QKD, es uno de los protocolos más adecuados para realizar transmisiones de clave cuánticas a largas distancias, ya que en pulsos consecutivos se conservan la polarización y la fase en canales de fibra óptica, es decir que se preserva la fase relativa y la polarización. Para DPS Alice genera seguridad aprovechando la propiedad de los estados no ortogonales para codificar la información, que dice que un estado no ortogonal no se puede identificar con certeza utilizando una sola medida; se puede demostrar si suponemos que existe un intruso en el canal, Eve quien puede emplear varias técnicas para medir el estado y obtener la información, se convierte directamente en una fuente de ruido que introduce errores al sistema, los cuales pueden ser detectados en la generación de claves [179]. Para entender la estructura de DPS, se mostrara el funcionamiento del protocolo utilizando tres pulsos consecutivos como se muestra en la figura 4.11. Bob puede medir un fotón a la vez de los codificados por Alice, utilizando cuatro instantes de tiempo, donde la probabilidad de detectar el fotón en cada instante de tiempo está dada por.

- En el camino (a-s) $p = 1/6$.

- Camino (b-s) o (c-l) $p = 1/6 + 1/6 = 1/3$.

- Camino (b-l) o (c-s) $p = 1/6 + 1/6 = 1/3$, IV) camino (c-l) $p = 1/6$, la clave solo se puede generar con los fotones detectados por segunda y tercera vez, por lo tanto cuando Alice envía N pulsos, la clave que se forma contiene $2N/3$ fotones.

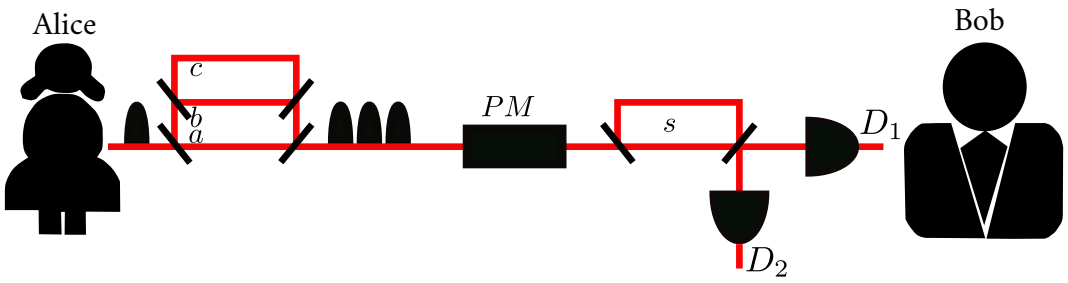

Figura. 4.11 Protocolo distribución de clave cuántica de cambio de fase diferencial DPS-QKD para 3 pulsos consecutivos: En la figura se muestra la configuración de DPS-QKD. Alice modula aleatoriamente y ortogonalmente el estado de polarización de cada pulso de la señal DPS, y Bob recibe la señal transmitida con un interferómetro de retardo de dos bits. Con la configuración de dos pulsos separados al doble del intervalo de tiempo que interfieren entre sí, y los fotones resultantes que entran en un detector de conversión ascendente dos estados de polarización ortogonal. Con estas características se puede decir que la eficiencia de detección se mantiene constante siendo los detectores sensibles a la polarización, figura modificada del trabajo [180].

El protocolo DPS se puede generalizar para n pulsos, y funciona similarmente que para 3 pulsos. Alice codifica cada fotón como una superposición de pulsos que posan por un interferómetro de Mach-zehnder desequilibrado, donde la probabilidad de que los fotones toman los n caminos posibles es de $p = 1/n$, por lo que Bob tiene $n + 1$ intervalos de tiempo para detectar el fotón enviado, con esta configuración si el fotón se mide en el primero o en el último intervalo de tiempo, la detección se presenta como un proceso aleatorio, es decir que el fotón puede estar en el detector 1 o en el detector 2, en el intervalo de tiempo $n - 1$ se generan las claves si el fotón está en la primero o última ranura y puede tomos el n-simo camino más corto o más largo, con probabilidad $p = 1/n + 1/n = 1/2n$, para los $n - 1$ tiempos restantes la probabilidad de las trayectorias es $p = 1/n + 1/n = 1/2n$, lo que significa que los fotones pueden recorrer k-ésimas trayectorias, y por lo tanto cuando Alice transmite N fotones superpuestos con n pulsos, solo $N(n-1)/n$ fotones contribuyen a la generación de la clave como se muestra en la figura 4.12 [181, 182].

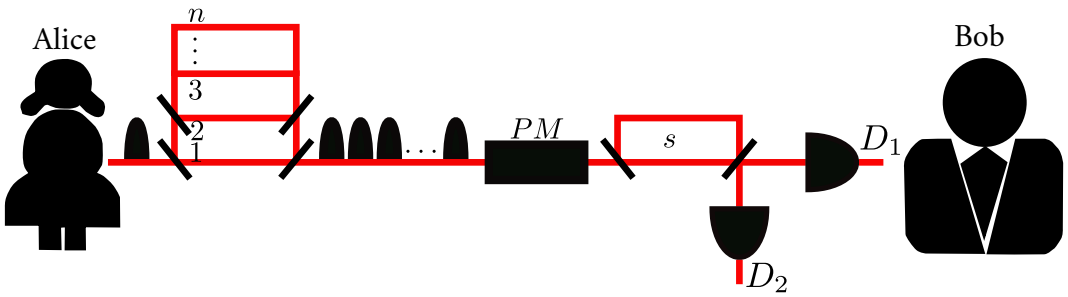

Figura. 4.12 Protocolo DPS-QKD para n pulsos consecutivos: En la figura se muestran pulsos aislados que llegan a Bob, que detecta un posible fotón en tres intervalos de tiempo, figura modificada del trabajo [183].

4.2.8. Protocolo Distribución de Claves Cuánticas de un Paso basada en el Entrelazamiento EPR, EQKD

El Protocolo EQKD se desarrolló utilizando el entrelazamiento y la codificación superdensa, propiedades de la mecánica cuántica que mejoran la seguridad de la información, este protocolo

genera las claves de cifrado mediante una secuencia organizada establecida por el transmisor Alice y el receptor Bob:

- Alice prepara una secuencia de N bits clásicos y la enumera de forma ordenada.
- Alice prepara una secuencia de estados EPR ordenados igual que en la secuencia de los bits clásicos, en este punto se realiza la codificación densa que forma los estados S, que luego se le trasmiten a Bob utilizando un canal cuántico. En este proceso Alice queda con el registro de la ubicación de los pares entrelazados que son los encargados de dar la ubicación de los bits cuánticos en la secuencia S.
- Bob recibe la secuencia de bits por el canal cuántico y por un canal clásico recibe de Alice la información de ubicación de las pares EPR.
- Bob extrae los estados EPR utilizando la información de ubicación que Alice compartió y realiza las mediciones de los estados con las desigualdades de Bell.
- Bob escoge de forma aleatoria un subconjunto de los estados que medio, estos serán fotones señuelos que Bob le envía a Alice con la información de ubicación, luego Alice le comparte a Bob la ubicación real de los fotones señuelo para que este compare los resultados y pueda verificar si han interceptado el canal, por lo tanto si el canal a permanecido sin intrusos se puede utilizar la claves, de lo contrario se descarta y se vuelve a empezar con el proceso de generaciones de claves. cuando se confirma la seguridad del canal, Alice y Bob comparte el resultado de las medidas restantes, hacen correspondencia para obtener la clave, en la figura 4.13 se muestra el funcionamiento del protocolo.

Luego se utilizan los estados de Bell para hacen las mediciones sobre los estados que Alice prepara y transmite.

$$
\begin{aligned}
|00\rangle = |\phi^{+}\rangle &= \frac{(|00\rangle + |11\rangle)}{\sqrt{2}} \\
|00\rangle = |\phi^{-}\rangle &= \frac{(|00\rangle - |11\rangle)}{\sqrt{2}} \\
|10\rangle = |\psi^{+}\rangle &= \frac{(|01\rangle + |10\rangle)}{\sqrt{2}} \\
|11\rangle = |\psi^{-}\rangle &= \frac{(|01\rangle - |10\rangle)}{\sqrt{2}}.
\end{aligned}
$$

Si el canal cuántico es interceptado por un intruso, Eve que mide los bits generados por Alice con el teorema de Bell para luego transmitir a Bob una nueva secuencia de estados. Teniendo en cuenta el principio de incertidumbre y el teorema de no clonación, las operaciones que realiza Eve generan errores en las medidas que Bob realiza, estos errores se incrementan por que Eve no conoce la información de ubicación de los pares EPR y por lo tanto a Eve le toca elegir al azar la ubicación de los EPR para enviársela a Bob, por lo tanto la seguridad del protocolo EQKD depende de la longitud de la secuencia de bits cuánticos, cuando más larga sea la secuencia más seguro será el protocolo, pues el intruso al realizar medidas sin información previa genera una tasa de errores mayor a la permitida por el protocolo.

El protocolo MEQKD es una modificación del algoritmo EQKD, utiliza el entrelazamiento y la codificación superdensa para general la clave de cifrado, en MEQKD se preparan pares EPR para un grupo de bits de la siguiente manera: 1) Alice prepara una secuencia N que se divide en grupos

ordenados de cuatro bits, 2) Alice enumera los bits de cada grupo de la forma $1, 2, 3, 4$ 3) Alice prepara los estados EPR teniendo en cuenta la posición de los bits dentro de los grupos, con esta información forma los estados S, por ejempla $(1, 2), (3, 4)$ o $(1, 3), (2, 4)$ y comunica S a Bob por un canal cuántico, 4) Bob recibe a S y elige aleatoriamente la ubicación de los bits dentro de los grupos $(1, 2), (3, 4)$ o $(1, 2), (3, 4)$ el para extraer los pares EPR y realizar las mediciones con el teorema de l Bell, 5) cuando Bob termina de medir se comunica con Alice por un canal clásico, Alice le comparte a Bob la ubicación de cada par EPR dentro de los grupos, si la ubicación de Alice y Bob no coinciden, la clave que genera ese grupo se descarta, si coinciden entonces Bob modifica la clave de ese grupo con codificación densa y la agrega a la clave completa sin procesar, 6) Bob escoge aleatoriamente una parte de clave, que serán los estados señuelos y le comunica a Alice la información de ubicación de estos estados, luego Alise le comparte Bob la ubicación original de los estados señuelos para que sea comparada, con esta información se tiene que si no hay espías la tasa de error es inferior a 46,875 % y por lo tanto el canal de comunicación es seguro para que se utilice la clave generada, 7) por último Alice y Bob comparten la clave restante, realizan correspondencia y corrección de errores para obtener la clave de cifrado [184].

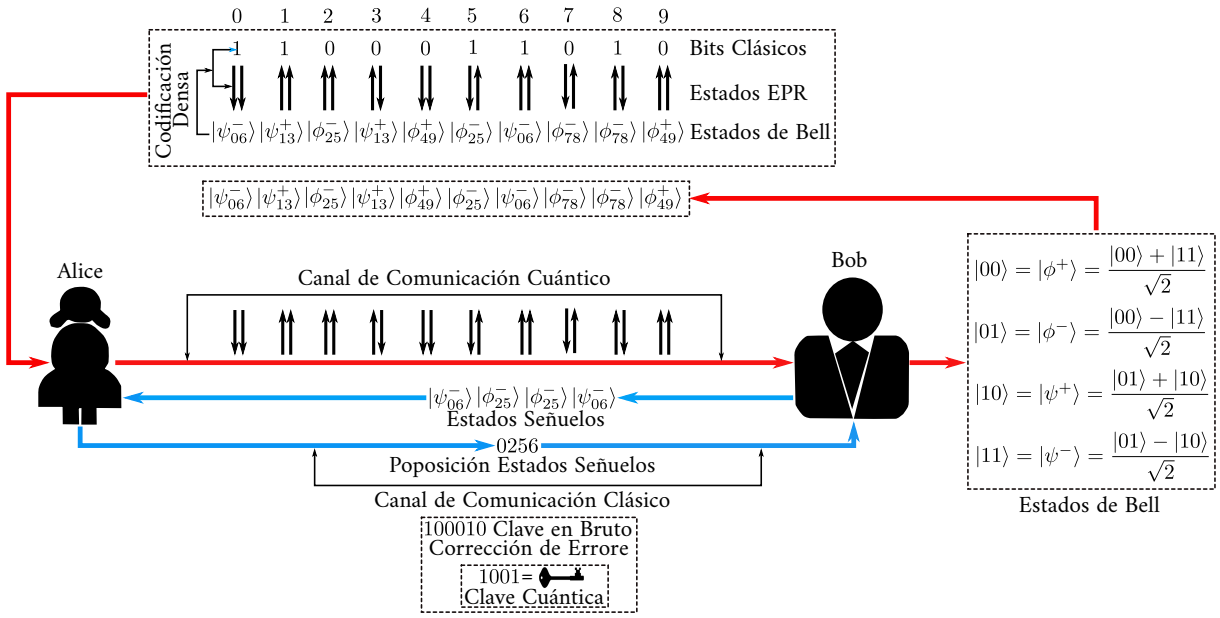

Figura. 4.13 Protocolo distribución de claves cuánticas de un paso basada en el entrelazamiento EPR, EQKD:En la figura se muestra la generación de claves criptografías utilizando una fuente de pares de fotones entrelazados EPR.

CAPÍTULO 5

MÉTODO CRIPTOGRÁFICO UTILIZANDO LA GENERACIÓN Y DISTRIBUCIÓN DE CLAVES CUÁNTICAS

En este capítulo se desarrolla una simulación de la generación y distribución de claves cuánticas utilizando el protocolo BB84, que estará integrado al cifrador de Vernam el cual permitirá codificar información con alta seguridad, aprovechando las propiedades de la mecánica cuántica que proporcionan las herramientas suficiente para abordar el problema de seguridad de la información. Es decir, se utilizará un sistema físico cuántico, como la polarización de fotones individuales, para realizar la distribución de claves cuánticas. Se muestran todos las fases de diseño del método criptográfico y su respectivo algoritmo.

5.1. Entorno de Desarrollo

En este trabajo, se implementó un modelo de criptografía cuántica simulado utilizando el lenguaje de programación Quiskit, que integra la generación y distribución de claves cuánticas del protocolo BB84 que ha sido el más utilizado en criptografía cuántica y el cifrado de Vernam que bajo condiciones de clave aleatoria tiene seguridad perfecta. Estos dos sistemas van a permitir codificar información con alta seguridad. En el desarrollo del modelo estarán interactuando tres agentes: Alice que representa el transmisor, Bob que representa el receptor y Eve que representa un espía en el canal de comunicación. Alice y Bob se van a encargar de generar la clave cuántica y Eve tratará de vulnerar el sistema de clave simétrica. El modelo está dividido en tres fases que permiten un mejor desarrollo de la simulación:

- Fase 1: En este fase se realiza la generación y distribución de la clave de criptografía cuántica entre Alice y Bob utilizando el protocolo BB84.

- Fase 2: En este fase se realiza el cifrado y descifrado de información entre Alice y Bob utilizando el modelo de Vernam y la clave de criptografía cuántica generada en el la fase 1.

- Fase 3: En este fase se pondrá la espía Eve en el canal de comunicación cuántico, para verificar si Alice y Bob pueden detectar el intruso, y decidir si la clave cuántica es segura para seguir con el proceso o abortarlo por fallas de seguridad.

5.1.1. Tecnología Utilizada

La simulación del método criptográfico está desarrollada, como se mencionó al inicio, con la tecnología Qiskit de IBM. Ésta es un entorno de simulación de código abierto construido para la computación cuántica. Posee herramientas para simular circuitos cuánticos a partir de librerías predeterminadas, que permiten probar los prototipos de computadores cuánticos desarrollados con los modelos de circuitos cuánticos universales. Qiskit corre principalmente sobre el lenguaje de programación python, pero ya existen versiones para Swift y JavaScript. Fue fundado por IBM Research para mejorar la colaboración científica del desarrollo de software para su servicio de computación cuántica en la nube. Este entorno de programación es la herramienta principal del trabajo debido a que permite ejecutar algoritmos cuánticos tanto en computadoras cuánticas como clásicas, utilizando librerías como:

- Terra: Es la librería de Qiskit que proporciona herramientas para crear circuitos cuánticos, y permite que los algoritmos se pueden ejecutar en el computador de IBM. Por otra parte Terra permite optimizar las simulaciones de circuitos cuánticos en dispositivo particulares.
- Aqua: Es una librería de algoritmos de dominio cruzado sobre los que se pueden construir aplicaciones específicas. Por ejemplo Qiskit Chemistry se desarrolló para utilizar Aqua en cálculos de química cuántica, y se puede utilizar para optimización, inteligencia artificial y finanzas. Aqua fue diseñado para ser extensible y para que se acople a modelos de optimización y oráculos.
- Aer: Proporciona simuladores de computación cuántica de alto rendimiento con modelos de ruido realistas.
- Ignis: Proporciona herramientas para caracterizar el ruido en dispositivos a corto plazo, y la mitigación y corrección de errores. En el Apéndice A aparece mas información sobre el lenguaje de programación y su forma de instalación en computadores clásicos.

En este sentido el Qiskit le permite a cualquier usuario desarrollar algoritmos, simulaciones y hacer experimentos cuánticos utilizando un lenguaje de alto nivel como python y procesamiento clásico, permite trabajar con qbits individuales y con compuertas cuánticas para poder explorar las ideas de computación cuántica. En el apéndice A de este trabajo se hara una breve descripción de este entorno de desarrollo.

5.2. Método Criptográfico

Para la generación y transmisión de claves cuánticas con polarización de fotones, se utilizo el protocolo BB84, y para realizar el cifrado y descifrado de los mensajes se utiliza el algoritmo de criptografía simétrica de Vernam que está compuesto por una compuerta Or-exclusiva (XOR) para el proceso de codificación de información. Este método criptográfico funciona con la interacción de tres agentes conocidos como, Alice que representa el emisor, Bob que representa el receptor, y Eve que representa un intruso que quiere espiar el sistema. Adicionalmente, se simularán dos canales de comunicación: uno cuántico que permitirá trasmitir los estados polarizados para generar la clave cuántica, y un canal clásico por donde se realizará la reconciliación de la clave y el envió del mensaje cifrado en forma clásica. Con respecto al intruso Eve, esta puede escuchas la conversación que se transmite por el canal cuántico, pero no puede alterarla, mientras que el flujo de información que va por el canal clásico está completamente abierto y puede ser manipulado, alterado o transformado [185].

En la primera parte del método criptográfico se utilizará la generación de claves cuánticas del protocolo BB84, donde Alice envía a Bob un conjunto aleatorio de bits codificados con dos bases de polarización que generan cuatro qubit, que se envían empleando el canal cuántico, los cuatro estados forman dos bases de politización con estados cuánticos, horizontal $|\leftrightarrow\rangle$, vertical $|\updownarrow\rangle$, oblicuo a derecha $|\nearrow\rangle$, oblicuo a izquierda $|\nwarrow\rangle$, y dos bases de polarización B_{HV} y B_x de tal manera que Alice puede enviar a Bob un 0 o un 1 codificado en cualquiera de los estados $|\leftrightarrow\rangle, |\updownarrow\rangle, |\nearrow\rangle, |\nwarrow\rangle$, que se transmite por medio de un canal cuántico que puede ser una fibra óptica o espacio libre. las bases B_{HV}, B_x satisfacen las condiciones del producto escalar entre estados, de igual manera los estados de diferente base no son ortogonales, ya que se cumple que $\langle B_{HV}|B_x\rangle \neq 0$. De esta manera se garantiza que un estado queda completamente determinado al proyectarlo sobre su base, mientras que si se proyecta sabré la otra base el resultado sera aleatorio, como se muestra en la figura 5.1 [186].

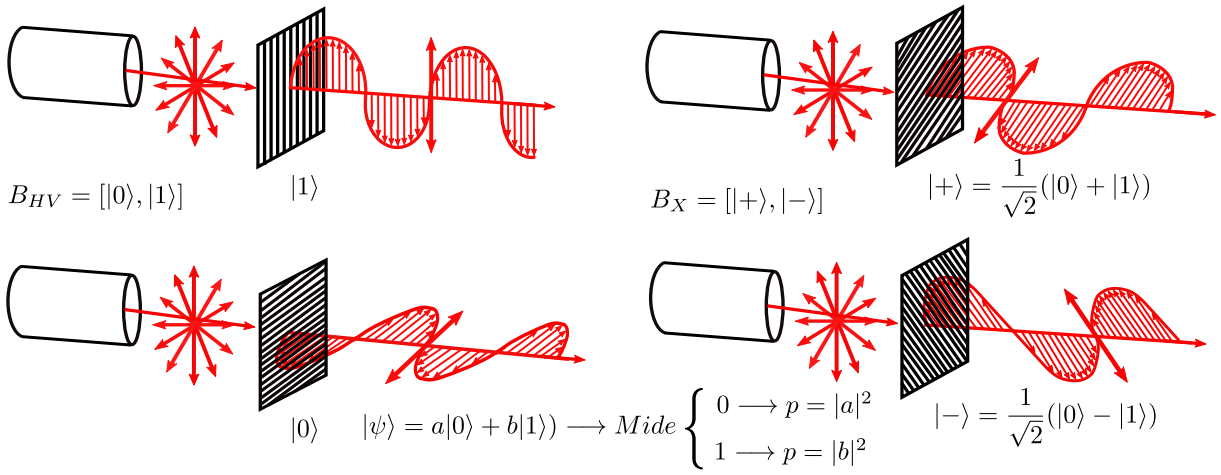

Figura. 5.1 Bases de polarización del protocolo BB84: En la figura se muestra las bases y los estados de polarización que se utilizan en el protocolo BB84 para generar la clave cuántica.

La generación y distribución de claves cuánticas utilizando el protocolo BB84 se realiza con la siguiente secuencia ordenada de pasos:

- Alice genera una cadena aleatoria formada por ceros y unos de tal manera que $a_1.a_2, ..., a_n$ con $a_n \in 0, 1$.
- Para cada bit de la cadena generada por Alicia, esta elige en forma aleatoria una de dos bases, con B_{HV} que puede codificar un 0 en $|0\rangle$ con polarización horizontal, y el 1 se codifica como $|1\rangle$ con polarización vertical, con la base B_X el 0 se codifica como $|+\rangle$ con polarización de 45°, y el 1 se codifica en $|-\rangle$ con polarización de $-45°$, con este proceso se forma una cadena de bases representadas con unos y ceros codificados, $\alpha_1.\alpha_2, ..., \alpha_n$ con $\alpha_n \in 0, 1$. En resumen para el modelo Alice puede utilizar la base $B_x = [|+\rangle, |-\rangle]$ que representan la polarización de $45°, -45°$ para codificar los estados $|+\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle)$, el estado $|-\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle)$, o la base $B_{HV} = [|0\rangle, |1\rangle]$ que representan la polarización horizontal y vertical para codificar los estados $|0\rangle = 1/\sqrt{2}(|+\rangle + |-\rangle)$, el estado $|1\rangle = 1/\sqrt{2}(|+\rangle - |-\rangle)$
- Bob recibe la cadena $\alpha_1, \alpha_2, ..., \alpha_n$ enviada por Alice, y elige de forma aleatoria una de sus dos bases B_{HV}, B_X para medir cada una de los estados enviados, es decir, que si Bob utiliza la B_{HV} entoces puede medir el estado $|0\rangle$ con $p = 1$, y $|1\rangle$ con $p = 1$, pero si tiene el estado $|+\rangle$ puede medir el $|0\rangle$ con $p = 1/2$ y $|1\rangle$ con $p = 1/2$, similarmente si tiene el estado $|-\rangle$ puede medir el $|0\rangle$ con $p = 1/2$ y $|1\rangle$ con $p = 1/2$. Pero si utiliza la base B_X si tiene el estado $|0\rangle$,

puede medir $|+\rangle$ con $p = 1/2$, y el $|-\rangle$ con $p = 1/2$, similarmente el estado $|1\rangle$ puede medir el $|+\rangle$ con $p = 1/2$, y el $|-\rangle$ con $p = 1/2$, también puede medir el estado $|+\rangle$ con $p = 1$ y $|-\rangle$ con $p = 1$. Esta fase inicial del protocolo se muestra en la figura 5.2.

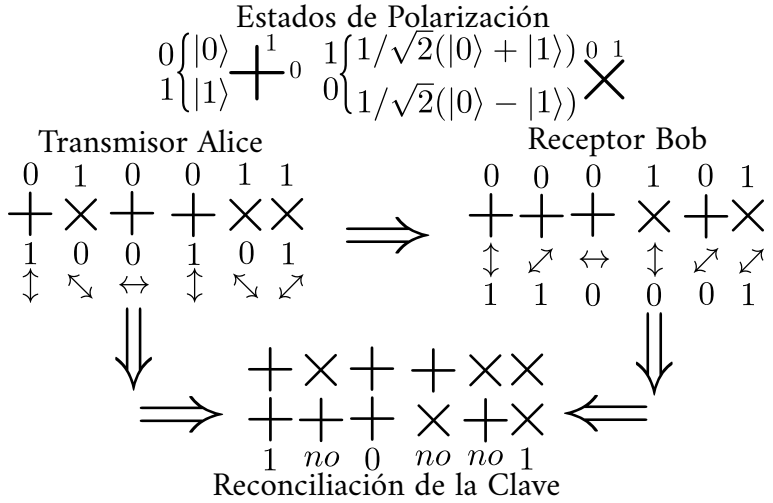

Figura. 5.2 Distribución de clave cuánticas con el protocolo BB84: En la figura se muestra el proceso como Alice envía fotones polarizados a Bob y como este mide los estado para luego hacer reconciliación de la clave cuántica.

5.2.1. Simulación del Método Criptográfico

Para simular la generación y distribución de las claves cuánticas se utiliza el lenguaje de programación python y el entorno qiskit con sus diferentes librerías. La construcción del algoritmo que simula el protocolo BB48 empieza con la exportación de los paquetes necesarios para el funcionamiento del entorno de desarrollo como:

```
#importar paquetes

import math
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np

# Import Qiskit

from qiskit import Aer, execute
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
```

Para que se genere una clave cuántica segura, esta deberá tener una longitud igual o mayor que la del mensaje, ya que si la clave es más corta que el mensaje es posible que un intruso pueda interceptar la clave y descifrar la información. Por lo tanto en la primera parte de la simulación el transmisor calcula la longitud del mensaje a cifrar y lo multiplica por tres, para tener una mayor seguridad de clave. Por ejemplo para el caso del mensaje “La nueva información hace posible las nuevas ideas” con

una longitud de 50 caracteres, entonces la clave tendrá una longitud de 150 bits. Con esta información inicia la simulación de la cogeneración y distribución de claves cuánticas, en donde Alice genera la siguiente secuencia binaria aleatoria de 150 bits, que es clave inicial

- Cadena binaria inicial de Alice

  011100000111011101011011010001000111001010101010001110001001010111111110110100000111011100010101001011110100011001000100111100001001010000011110 1

Esta cadena binaria es la primera secuencia aleatoria que Alice genera para luego codificar bit a bit en los estados de polarización que se le enviaran a Bob como se muestran el la figura 5.3. Este proceso se realiza mediante el siguiente algoritmo:

```
#mensaje secreto

mes = 'La nueva información hace posible las nuevas ideas'
print('Tu mensaje secreto: ',mes)

#tamaño inicial de la clave

n = len(mes)*3

#romper el mensaje en partes más pequeñas si la longitud > 10

nlist = []
for i in range(int(n/10)):
    nlist.append(10)
if n%10 != 0:
    nlist.append(n%10)
print('Longitud de la llave inicial: ',n)
```

```
# Hacer cadenas aleatorias de longitud string-length

def randomStringGen(string_length):

#Variables de salida utilizadas para acceder a los resultados

    output_list = []
    output = ''

#inicia la información del circuito cuántico

    backend = Aer.get_backend('qasm_simulator')
    circuits = ['rs']

#Ejecución del circuito cuántico en arreglos de 10 qubits
#Los resultados se agregarán y se recortarán al tamaño n bit correcto.
```

```
    n = string_length
    temp_n = 10
    temp_output = ''
    for i in range(math.ceil(n/temp_n)):

#Inicializar los registros cuánticos en el circuito.

        q = QuantumRegister(temp_n, name='q')
        c = ClassicalRegister(temp_n, name='c')
        rs = QuantumCircuit(q, c, name='rs')

#crear temp_n número de qubits todos en superposiciones
#La puerta Hadamard es la que hace la superposicione.

        for i in range(temp_n):
            rs.h(q[i])
            rs.measure(q[i],c[i])

#Ejecutar circuito y extraer 0s y 1s de la Clave

        result = execute(rs, backend, shots=1).result()
        counts = result.get_counts(rs)
        result_key = list(result.get_counts(rs).keys())
        temp_output = result_key[0]
        output += temp_output

#Salida de retorno recortada al tamaño n

    return output[:n]
```

```
key = randomStringGen(n)
ln = len(key)
print('Clave inicial: ',key)
print('Longitud de la Clave: ',ln)
```

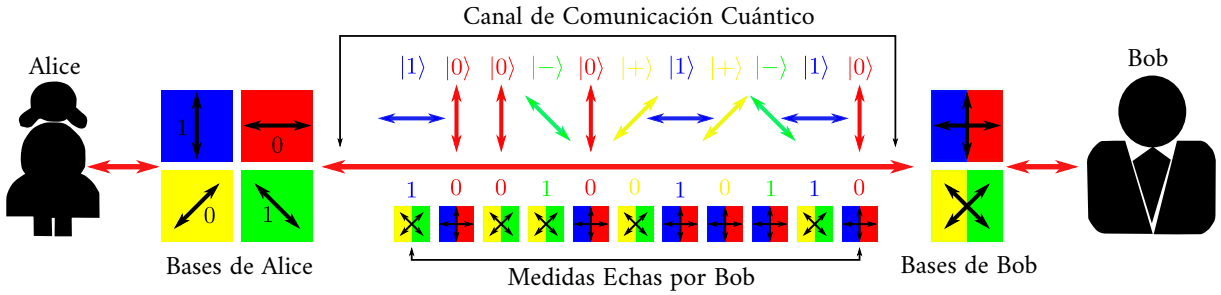

Figura. 5.3 Estructura de la transmisión de los estados cuánticos: Envía fotones polarizados de Alice a Bob utilizando un canal cuántico, fibra óptica.

Como Bob determina el estado que Alice le ha enviado escogiendo aleatoriamente, para cada uno de los estados recibidos, una de las dos posibles bases de $\langle B_{HV}|B_X\rangle \neq 0$, en este proceso de medición y almacenamiento es predecible que Bob escoja la misma base que Alice en el 50 % de los casos. Ya que Alice envía un 0 codificado en el estado horizontal del sistema $|\leftrightarrow\rangle$, Bob pude medir el estado cuántico$|\leftrightarrow\rangle$ en la base B_{HV}, o el estado cuántico $|\nearrow\rangle$en la base B_X, similarmente si Alice envía un 1 codificado en el estado $|\updownarrow\rangle$, Bob puede medir el estado $|\updownarrow\rangle$ o el estado $|\nwarrow\rangle$ dependiendo de la base que utilice B_{HV} o B_X, como se muestra en la tabla 5.1.

Estado	Medida con B_{HV}	Medida con B_X
$\|0\rangle$	$\|0\rangle$ con $p = 1$	$\|+\rangle$ con $p = 1/2$, y $\|-\rangle$ con $p = 1/2$
$\|1\rangle$	$\|1\rangle$ con $p = 1$	$\|+\rangle$ con $p = 1/2$, y $\|-\rangle$ con $p = 1/2$
$\|+\rangle$	$\|0\rangle$ con $p = 1/2$, y $\|1\rangle$ con $p = 1/2$	$\|+\rangle$ con $p = 1$
$\|-\rangle$	$\|0\rangle$ con $p = 1/2$, y $\|1\rangle$ con $p = 1/2$	$\|-\rangle$ con $p = 1$

Tabla. 5.1 Probabilidad de medir un estado cuántico: En la tabla se muestra la probabilidad que tiene Bob para medir los estados codificados por Alice utilizando las bases $B_{HV} = [|0\rangle, |1\rangle]$, ó $B_X = [|+\rangle, |-\rangle]$.

Después que Bob mida y almacene toda la cadena de estados, se comunica con Alice utilizando un canal clásico y le dice en que bases midió cada una de los estados, comparan realizadas y escogen los estados que son iguales para generar la clave de cifrado. Como se especifica en la siguiente secuencia [187]:

- Bob se comunica con Alice y envía la cadena de bases con que midió los estados. Utiliza para este proceso un canal de comunicación público inseguro.
- Por el mismo canal, Alice indica los medidas que son correctas.
- Alice y Bob comparan los resultados y borran de sus cadenas los bits en los que se han usado bases diferentes.
- Alice envía a Bob una lista de posiciones junto a su valor para estimar la tasa de error.
- Si la tasa de error es inferior al 25 %, se da el intercambio de claves, si es superior se aborta la comunicación ya que hay la certeza de que existe un intruso en el canal de comunicación, como se muestra en la figura 5.6.

En el algoritmo, después de que Alice genera su primera cadena aleatoria binaria, escoge una cadena de rotación al azar para enviar los estados a Bob por medio de un canal cuántico que para este caso es simulado, pero en caso de realizase el protocolo en forma experimental el canal cuántico seria un fibra óptica o el espació libre. Similarmente cuando le llegan los estados a Bob, este también escoge al azar una cadena de rotación, que le permite medir cada estado, de este proceso Bob genera dos cadenas, una con la rotación de las bases, y la otra con las medidas de los estados. Posteriormente Alice y Bob se comunican por canal clásico y comparten las cadenas de rotación, de donde Alice le comunica a Bob cuales son les estados correctos para generar la clave criptográfica, tal como se muestra en la siguiente parte del algoritmo.

```
#generar cadenas de rotación al azar para Alice y Bob

Alice_rotate = randomStringGen(n)
Bob_rotate = randomStringGen(n)
print("Cadena de rotación de alicia:",Alice_rotate)
print("Cadena de rotación de Bob:  ",Bob_rotate)

#iniciar cuántico cuántico

backend = Aer.get_backend('qasm_simulator')
shots = 1
circuits = ['send_over']
Bob_result = ''

#Definir las variables temporales utilizadas en la división del
#cuántico cuántico, es decir si la longitud del mensaje > 10

for ind,l in enumerate(nlist):
    if l < 10:
        key_temp = key[10*ind:10*ind+l]
        Ar_temp = Alice_rotate[10*ind:10*ind+l]
        Br_temp = Bob_rotate[10*ind:10*ind+l]
    else:
        key_temp = key[l*ind:l*(ind+1)]
        Ar_temp = Alice_rotate[l*ind:l*(ind+1)]
        Br_temp = Bob_rotate[l*ind:l*(ind+1)]

#inicia la información de tu circuito cuántico

    q = QuantumRegister(l, name='q')
    c = ClassicalRegister(l, name='c')
    send_over = QuantumCircuit(q, c, name='send_over')

#preparar qubits basados en clave; Añadir puertas Hadamard
#para Alice y Bob
#cadenas de rotación

    for i,j,k,n in zip(key_temp,Ar_temp,Br_temp,range(0,len(key_temp))):
        i = int(i)
        j = int(j)
        k = int(k)
        if i > 0:
            send_over.x(q[n])

#cadena de rotación de Alicia

        if j > 0:
```

```
            send_over.h(q[n])

#cadena de rotación de Bob

        if k > 0:
            send_over.h(q[n])
        send_over.measure(q[n],c[n])

#ejecutar el circuito cuántico

    result_so = execute([send_over], backend, shots=shots).result()
    counts_so = result_so.get_counts(send_over)
    result_key_so = list(result_so.get_counts(send_over).keys())
    Bob_result += result_key_so[0][::-1]

print("Los resultados de bob: ", Bob_result)
```

```
#imprimir circuito cuántica para las medidas de Bob inicializado en cero
send_over.draw()
```

```
#Ejecute el circuito en el simulador de qasm

backend_sim = BasicAer.get_backend('qasm_simulator')

#número de repeticiones del circuito 1024 valor predeterminado

job_sim = execute(send_over, backend_sim, shots=1024)

#los resultados de la simulación
result_sim = job_sim.result()
counts = result_sim.get_counts(send_over)
print(counts)
```

De la simulación se obtienen la cadena binaria aleatoria de las bases de rotación que Alice utiliza para codificar y la cadena binaria aleatoria de las bases rotación que Bob le asigno a los estados trasmitidos para medirlos. También se obtiene la cadena binaria con las medidas echas por Bob, utilizando el circuito cuántico de la figura 5.5.

- Cadena binaria aleatoria de la rotación de las bases de Alice

  1100110000010111001010110100111011111110101000000011010110000011101001101110011001011000011101000010000111011011101101111011101101011101011000110000100.

- Cadena binaria aleatoria de la rotación de las bases de Bob

10010100101000010011000100110000100111101111001010100111001111110010011110010110
01111101001011010101100001000101101101001000100101011010101111001010110.

- Cadena binaria aleatoria de las medidas echas por Bob

01110000011101110101101101000100011100101010101000111000100101011111111101101000
00111000000111100010101001011110100011001000100111100001001010000011110 1

En la ejecución del circuito cuántico que Bob utiliza para medir los estados enviados por Alice, aparecen errores que se deben tener en cuenta en la transmisión de información. Para esto se utiliza la herramienta backend del Qiskit Aer, que simula la ejecución ideal de un circuito cuántico y devuelve el vector de estado cuántico de la simulación representado en el histograma de la figura 5.4, que muestra la salida del circuito usando un modelo de ruido generado automáticamente basado en los parámetros de un dispositivo real.

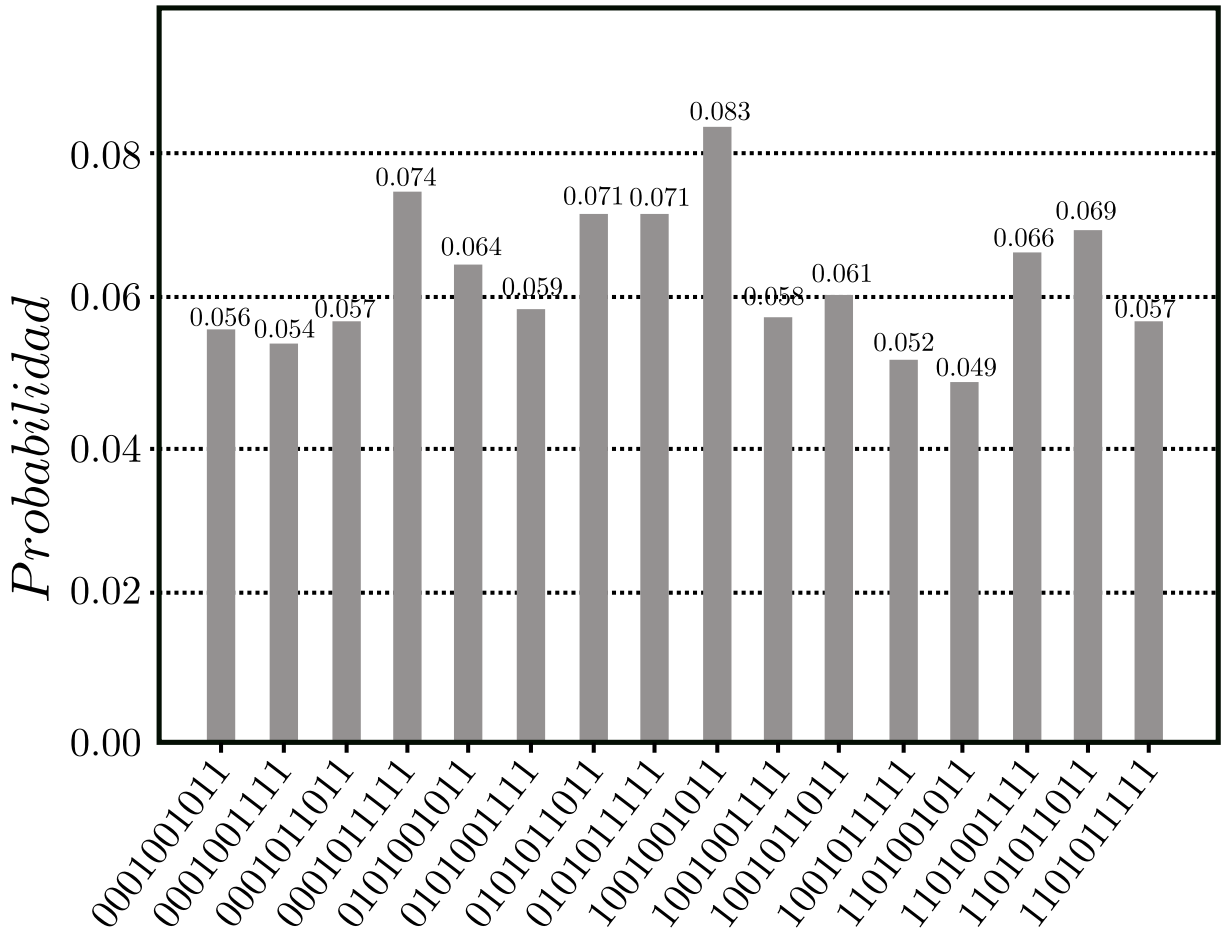

Figura. 5.4 Simulación de la cadena de salida del circuito cuántico de Bob: El histograma muestra la cadena de bits de salida del circuito cuántico que Bob utiliza para medir los estados agrupados por bloque de 10 bots. El número de veces que se ejecuta el circuito es 1024, figura generada con el Qiskit.

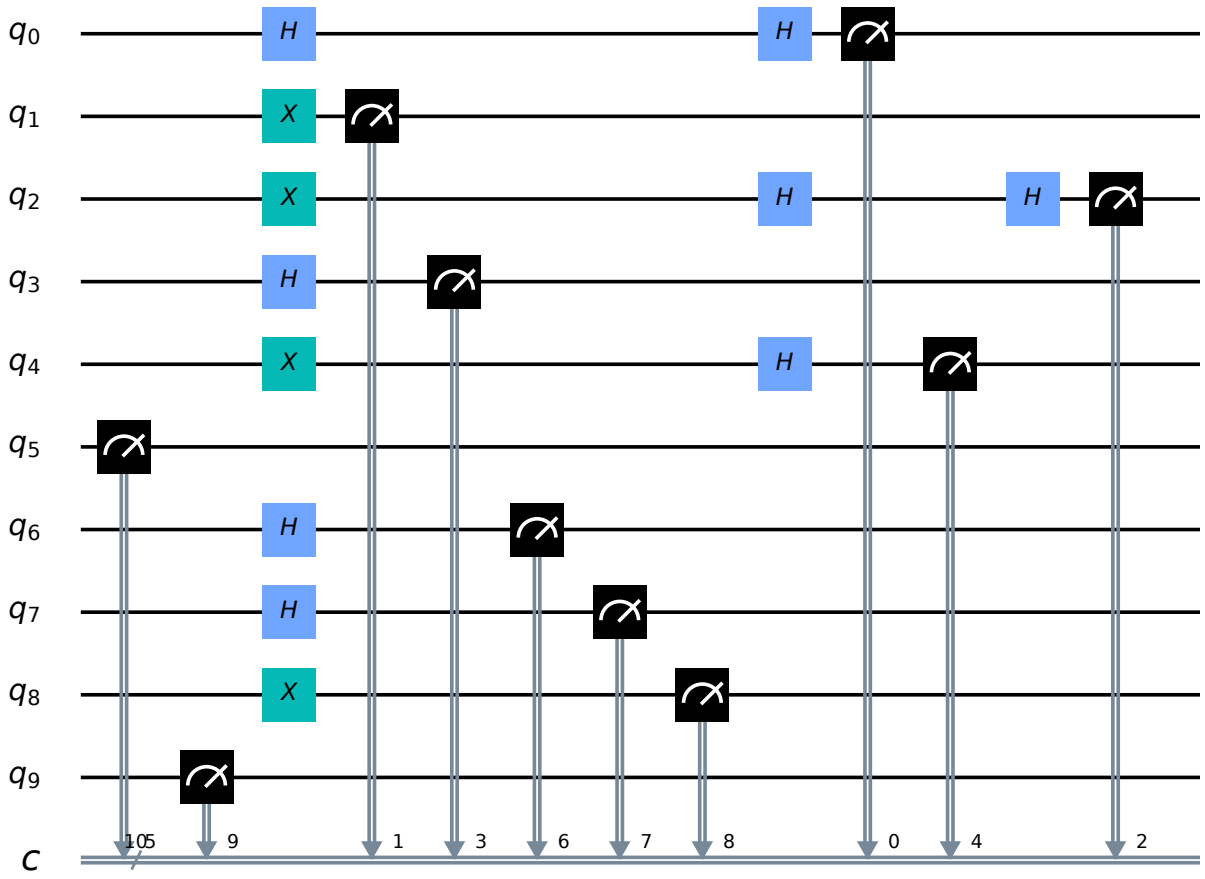

Figura. 5.5 Circuito cuántico para medir los estados entre Alice y Bob: El circuito cuántico inicializado en el estado $|0\rangle$ para que Bob empiece a medir los estados cuánticos trasmitidos por Alice, figura generada con el Qiskit.

```
#selección de los bit de la clave

def makeKey(rotation1,rotation2,results):
    key = ''
    count = 0
    for i,j in zip(rotation1,rotation2):
        if i == j:
            key += results[count]
        count += 1
    return key

Akey = makeKey(Bob_rotate,Alice_rotate,key)
Bkey = makeKey(Bob_rotate,Alice_rotate,Bob_result)

print("Clave de Alice:",Akey)
print("Calve de Bob:",Bkey)
```

Después que Alice y Bob seleccionan las posiciones que coinciden en sus cadenas binarias de rota-

ción, Alice selecciona los bits que forman la clave cuántica de su cadena binaria inicial, similarmente Bob selecciona los bits que forman la clave cuántica de su cadena binaria de medidas. En el protocolo no se comparten la cadena binaria inicial de Alice y la cadena de medias de Bob, son secretas,en la figura 5.6 se muestra el proceso de reconciliación de la clave cuántica.

- Clave de Alice

  0100011110111100110010001100110110111111010000110000101001011110101010111100111111

- Clave de Bob

  0100011110111100110010001100110110111111010000110000101001011110101010111100111111

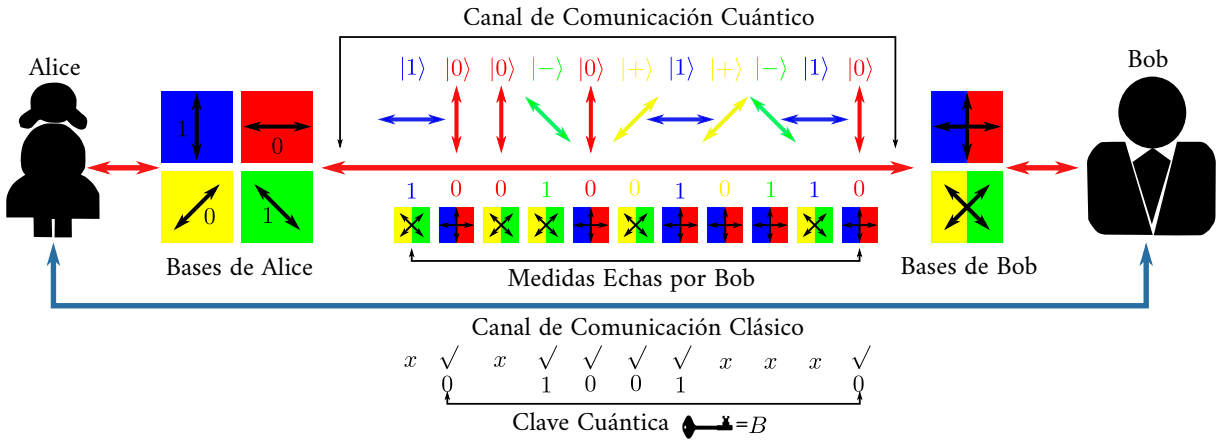

Figura. 5.6 Reconciliación de la clave cuántica del protocolo BB84: En la figura de esquematiza el proceso de transmisión, medida y coparacion de estados cuánticos para generar la clave criptográfica.

Para cifrar y descifrar el mensaje se va ha utilizar el cifrador de Vernam, esta técnica criptográficas opera un texto en una compuerta XOR[1] bit a bit con una clave pseudoaleatoria del mismo tamaño del mensaje. Es decir, a cada carácter del mensaje se le asigna n bits, los cuales se suman en una XOR mód 2 con una clave de igual longitud, en este proceso M_I representan los n bits de cada carácter del mensaje, K_i es la clave, C_i es el mensaje cifrado, por lo tanto $C_i = M_I \oplus K_i$ para $i = 1, 2, ..., n$. El descifrado utiliza la propiedad involutiva de la XOR, $C_i \oplus K_i = (M_i \oplus K_i \oplus K_I)$ y como $K_i \oplus K_i = 0$, para cualquier K_i se obtiene el mensaje descifrado como $C_i \oplus K_i = M_I$ tal como se muestra en la figura 5.7 [39].

Además se puede decir que al utilizar el cifrador de Vernam con claves cuánticas aleatorias generadas con el protocolo BB84 se tiene un mensaje completamente segur, por que se tiene un secreto perfecto, demostrado por Claude Shannon en la década de los 40 usando elementos de la teoría de la información. Esto significa que el texto cifrado no presenta ninguna información del mensaje original, es decir que la probabilidad a priori de un mensaje original M es igual a la probabilidad a posteriori de un mensaje original M dado el mensaje cifrado, de forma mas general se puede obtener

[1]La XOR es una puerta lógica digital que cumple con la siguiente ecuación utilizando algebra booleana $x = A * \overline{B} + \overline{A}B = A \oplus B$ es decir, si A o B son ceros 0 o unos 1 entonces la salida $x = 0$, pero si $A = 0, B = 1$ o $A = 1, B = 0$, La operación XOR es verdadera si las entradas no son iguales, de otro modo el resultado es falso.

con el algoritmo Vernam utilizando la generación y distribución de claves cuánticas del protocolo BB84 [188, 189]. También se a desmostado que este sistema criptográfico tiene un secreto perfecto, si se cumple la siguiente igualdad:

$$p(M) = p_C(M), \tag{5.1}$$

donde M representa el mensaje original, $p(M)$ es la probabilidad a priori de haber recibido un mensaje M con $\sum_M p(M) = 1$, $p_C(M)$ que representa la probabilidad a posteriori de haber recibido un mensaje M en tanto se ha recibido un mensaje cifrado C. Y Utilizando la teoría de información mutua se puede probar que se tiene un secreto perfecto si se cumple que:

$$I(M, C) = 0, \tag{5.2}$$

ya que la información mutua se puede calcular a partir de la entropía, entonces se tiene que:

$$I(M, C) = H(M) - H(M/C) = H(C) - H(C/M), \tag{5.3}$$

de donde se tiene que:

$$H(M) = H(M/C), \tag{5.4}$$

además se puede probar que:

$$I(M, C) \geq H(M) - H(K). \tag{5.5}$$

La anterior desigualdad muestra que cuando la incertidumbre de un conjunto de claves es pequeño, la información mutua es más grande y por tanto habrá una independencia mayor entre el mensaje original y el texto cifrado.y si se tiene un secreto perfecto $I(M, C) = 0$, entonces se tiene $H(K) \geq H(M)$ que se denomina como la Desigualdad pesimista de Shannon, y si cada clave y cada mensaje original tienen la misma probabilidad de ocurrencia, y además el número de claves se mayor o igual al conjunto de mensajes cifrados $|K| \geq |M|$, entonces un sistema con secreto perfecto, satisface que el espacio de las claves aleatorias es igual o mayor que el espacio de los mensajes [190].

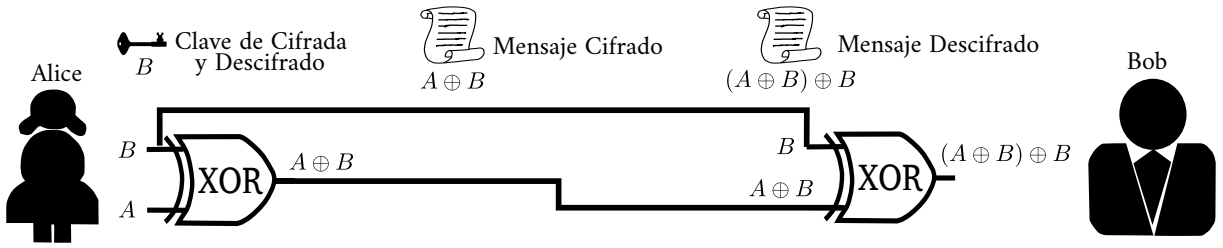

Figura. 5.7 Cifrado de Vernam: En la figura se muestra la operación XOR para cifrar y descifrar.

```
#cifrado y descifrado del mensaje con la clave generada cáusticamente

shortened_Akey = Akey[:len(mes)]
encoded_m=''

#cifrar el mensaje utilizando mi clave de cifrado clave final

for m,k in zip(mes, shortened_Akey):
    caracter = ord(m)
    xor = caracter ^ ord(k)
```

```
    encoded_m += chr(xor)
print('mensaje codificado:  ',encoded_m)

#hacer clave igual longitud tiene mensaje

shortened_Bkey = Bkey[:len(mes)]

decoded_m = ''
for m,k in zip(encoded_m, shortened_Bkey):
    caracter_d = ord(m)
    xor = caracter_d ^ ord(k)
    decoded_m += chr(xor)
print('mensaje decodificado:',decoded_m)
```

Después que Alice y Bob realizan la reconducción de la clave se puede cifrar o descifrar la información, Para este caso se utiliza el proceso del cifrador de Vernam que costa de una compuertas XOR que permite codificar un mensaje computándolo bit a bit con la clave generada con el protocolo cuántico BB84, en la figura 5.8 se esquematiza el proceso de cifrado y descifrado de información utilizando el cifrador de Vernam y la distribución de claves cuánticas del protocolo BB84.

- Mensaje cifrado

  $|P \cdot \_DTGQ \cdot X\_V\wedge C]QRX\tilde{A}\wedge \cdot YQRU \cdot @CXR]U\cdot]QB \cdot\wedge ETFQB \cdot YUTPB$

- Mensaje descifrado

  La nueva información hace posible las nuevas ideas

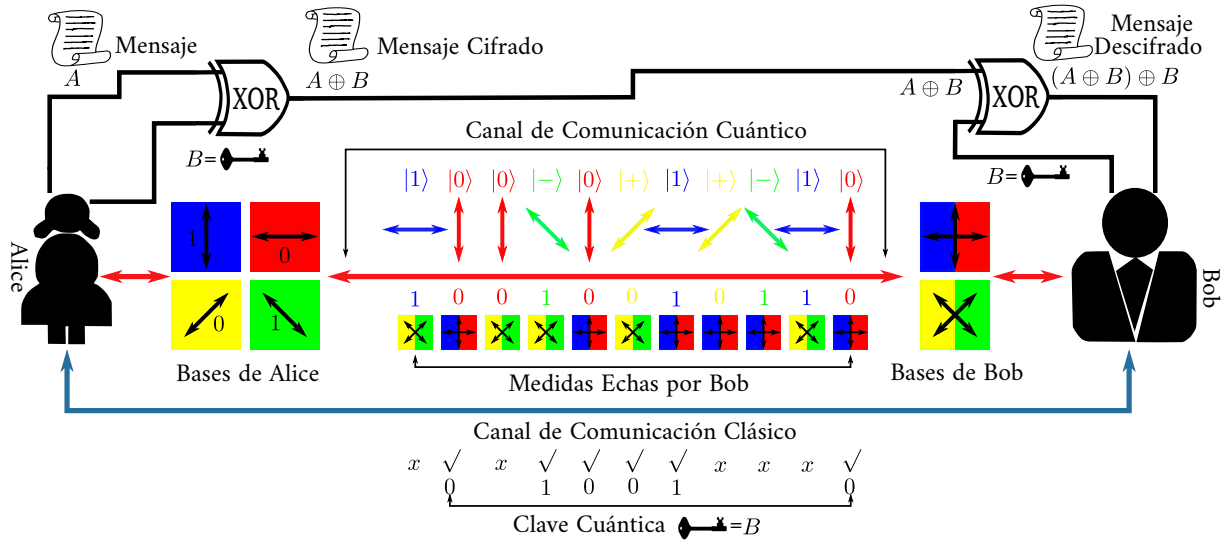

Figura. 5.8 Método criptográfico híbrido: En la figura se esquematiza un método criptográfico, que utiliza la generación y distribución de clave cuántica del protocolo cuántico BB84 y la codificación y decodificación del cifrado de Vernam como libreta de un solo uso.

5.2.2. Intersección y Reenvió de Estados Cuánticos

Una prueba de seguridad, para los protocolos de transmisión de claves cuánticas es cuando Eve intercepta el canal de comunicación cuántico del sistema, captura el trafico de información y lo retransmite para que le víctimas crean que están hablando con la entidad correcta a través de una conexión segura, cuando lo que sucede en realidad es que toda comunicación esta siendo controlada por el intruso. El éxito de la interceptación depende fundamentalmente de la capacidad del intruso para medir y reenviar los estados [191].

De esta manera cuando Alice envía los fotones polarizados a Bob, existe la posibilidad que el canal cuántico sea interceptado por un espía Eve, que tiene el conocimiento de las mimas base B_{HV} y B_X empleadas por Alice y Bob para generar la clave criptográfica. Eve intercepta el fluido de información en el canal cuántica y realiza mediciones no demoledoras con sus bases, entonces, cuando Alice envía un fotón polarizado con una de sus bases, este es medido por Eve utilizando una de sus bases en forma aleatoria, luego por la influencia del teorema de no clonación en mecánica cuántica, Eve al interactuar con los qubits modifica su estado, y es este estado modificado el que se le retransmite. Como Bob no sabe aun que el canal cuántico a sido interceptado por un intruso, entonces el mide los estados que le llegan en forma aleatoriamente con una de sus dos bases B_{HV} o B_X. Cuando finaliza la transmisión de fotones, Bob ya a generado sus dos cadenas binarias una con la rotación de las bases y otra con la medidas echas a los estados, se comunica con Alice utilizando un canal clásico de comunicación, Alice y Bob comparan sus cadenas de rotación y encojen los bits donde coincidieron, pero como Eve interceptó el canal y modificó los estados, lo que originó una tasa de bits error mas grande de la aceptada por el protocolo de distribución de claves cuánticas, esto acontecimiento alerta a Alice y Bob de la presencia de un intruso en el canal de comunicación cuántico, por lo tanto Alice y Bob abortan la comunicación y no se genera la clave criptográfica, proceso que se muestra en la siguiente figura 5.9.

Cuando el intruso Eve tiene la capacidad de interceptar cada qubit en forma individual, medirlo en forma aleatoria con una de sus dos bases para retransmitirlo a Bob, se tiene un ataque de intersección y reenvió [192] de información por el canal cuántico de comunicación. En este ataque el intruso Eve almacenada una cadena de medidas s con $s_n \in 0, 1$, y longitud n, donde cada bit de la cadena tiene una probabilidad $I = \sum_{i=1}^{n} p((A_n = s_n)/(\alpha_n = \beta_n))$, donde α_n es la cadena de codificación de Alice, β_n es la cadena de medición de Bob, que representa la información de Eva sobre Alicia, pera para el intruso tener la información I tubo que medirse los estados transmitidos par Alice, lo que introduce un error E en la cadena compartida por Alice y Bob, de donde es clara que para un $E = \sum_{i=1}^{n} p((a_n \neq b_n)/(\alpha_n = \beta_n))$ existe un I máxima que se puede alcanzar, a_n es la cadena binaria de Alice y b_n es la cadena binaria de Bob.

Luego como Eva tiene la posibilidad de manipular la información en el canal cuántico, entonces elle hacer un ataque complementario, manipular la cadena de rotación de las bases con las que Alice piensa que Bob midió, pero este proceso no le aporta ninguna información adicional. También Eve puede manipular y elegir las posiciones que Bob descarta, peo esto tampoco aporta información adicional, por que Eve tendría la misma información sobre todos los bits, entonces Eve no tiene ninguna ganancia de información manipulando las pociones de los estados en el sistema Bob.

Por otra parte, cuando Eve realiza un ataque no simétrico, esta puede tener toda la información de Alice y la mitad de los bits de Bob, pero esto no es suficiente información para que el ataque sea eficiente, lo que se necesita es que Eve sea capas de tener la información correcta de mas de la mitad de los bits de Bob, y para esto Eve podría pensar en hacer un ataque de suplantación de identidad de la siguiente manera.

Como Alice genera cadena aleatoria binaria ai la cual codifica bit a bit en la cadena de bases aleatoria α_i. Entonces Eva tiene la capacidad de interceptar todos los qubits y medirlas con una cadena de bases aleatoria e_i, en este proceso el intruso guarda el resultado de la medición y reenvía a Bob el qubit resultante. Bob comunica por el canal clásico la cadena de bases que midió. Eva intercepta esta la cadena de Bob y envía a Alicia su cadena de bases.

En esta parte del ataque Eva ya tiene una cadena de bits exactamente igual a la de Alicia y además conoce la mitad de las posiciones de la cadena de Bob, exactamente aquellas pociones de las bases que coinciden, y si Eve tuviera cuatro posiciones para comparar cada bit de Alice conseguirá tener toda las pociones de Bob y el protocolo seria vulnerado, pero como Eve solo tiene dos posiciones disponibles, entonces el intruso tiene que emparejar el bit i de Alice de manera correcta con el bit j de Bob hasta a aproximarse a $2i + j$, lo que conduce nuevamente a una ataque simétrico que no es eficiente en la trasmisión de claves cuánticas [193].

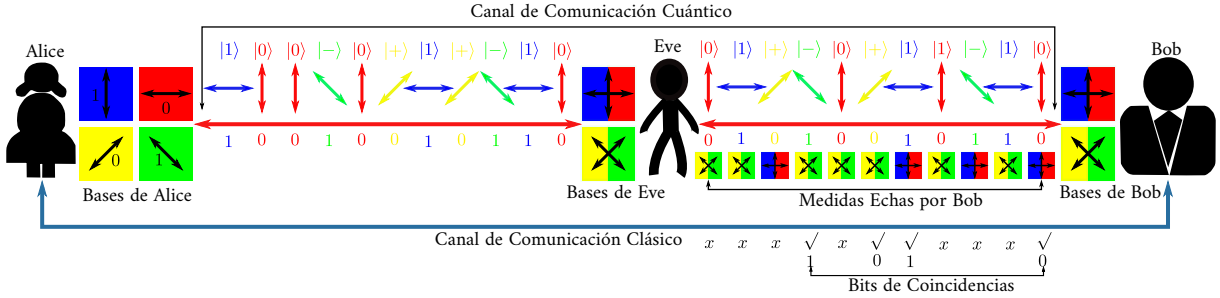

Figura. 5.9 Intruso en el canal cuántico: Eve no puede copiar completamente los estados enviados por Alice a Bob.

A continuación Eve que representa un atacante en el protocolo BB84, intercepta el canal cuántico de comunicación entre Alice y Bob, y trata de generar un ataque de intersección y reenvió de estados cuánticos. Es decir que Eva puede elige si intercepta o no cada qubit, si lo intercepta entonces lo mide utilizando una base seleccionada de forma aleatoria, guarda el resultado de la medida y reenvía a Bob el qubit resultante. Después que Eva obtiene su secuencia binaria, cada bit tendrá una única probabilidad de ser igual al bit de Alice.

```
#inicia la simulación cuántica

backend = Aer.get_backend('qasm_simulator')
shots = 1
circuits = ['Eve']
Eve_result = ''
for ind,l in enumerate(nlist):

#Definir las variables temporales para el programa cuántico

    si la longitud del mensaje > 10
    if l < 10:
        key_temp = key[10*ind:10*ind+l]
```

```
        Ar_temp = Alice_rotate[10*ind:10*ind+l]
    else:
        key_temp = key[l*ind:l*(ind+1)]
        Ar_temp = Alice_rotate[l*ind:l*(ind+1)]

#inicia el resto de la información de tu circuito cuántico

    q = QuantumRegister(l, name='q')
    c = ClassicalRegister(l, name='c')
    Eve = QuantumCircuit(q, c, name='Eve')

#rotación de Eve

    for i,j,n in zip(key_temp,Ar_temp,range(0,len(key_temp))):
        i = int(i)
        j = int(j)
        if i > 0:
            Eve.x(q[n])
        if j > 0:
            Eve.h(q[n])
        Eve.measure(q[n],c[n])

    result_eve = execute(Eve, backend, shots=shots).result()

    counts_eve = result_eve.get_counts()

    result_key_eve = list(result_eve.get_counts().keys())

    Eve_result += result_key_eve[0][::-1]

print("resultados de Eve: ", Eve_result)
```

El intruso Eve puede medir y reenviar los estados que Alice le trasmite a Bob sin que ellos se den cuanta de la intrusión. En este proceso Eve mide y almacena una cadena binaria con los estados que serán para retransmitirlas a Bob, pero como los estados cuánticos no se puede copiara ni clonar, entonces Eve genera un error en la cadena de medidas de Bob, permite suponer una instrucción en el sistema. En la simulación se muestra la cadena binaria generada por Eve, y en la figura 5.10 se muestran los resultados de la simulación del circuito cuántico usando un modelo de ruido.

- Cadena binaria de medidas hechas por Eve al medir los estados cuánticos

  1100001111100011001011111001111110011001111000100111010100000001100011101001111011
  1100011011011001000010011010001000101110011001001110010011110010110

```
#Ejecute el circuito en el simulador de qasm para las medidad de Eve.
```

```
backend_sim = BasicAer.get_backend('qasm_simulator')

# el número de repeticiones del circuito es de 1024

job_sim = execute(Eve, backend_sim, shots=1024)

# Grab the results from the job.

result_sim = job_sim.result()

counts_Eve = result_sim.get_counts(Eve)

print(counts_Eve)

plot_histogram(counts_Eve)
```

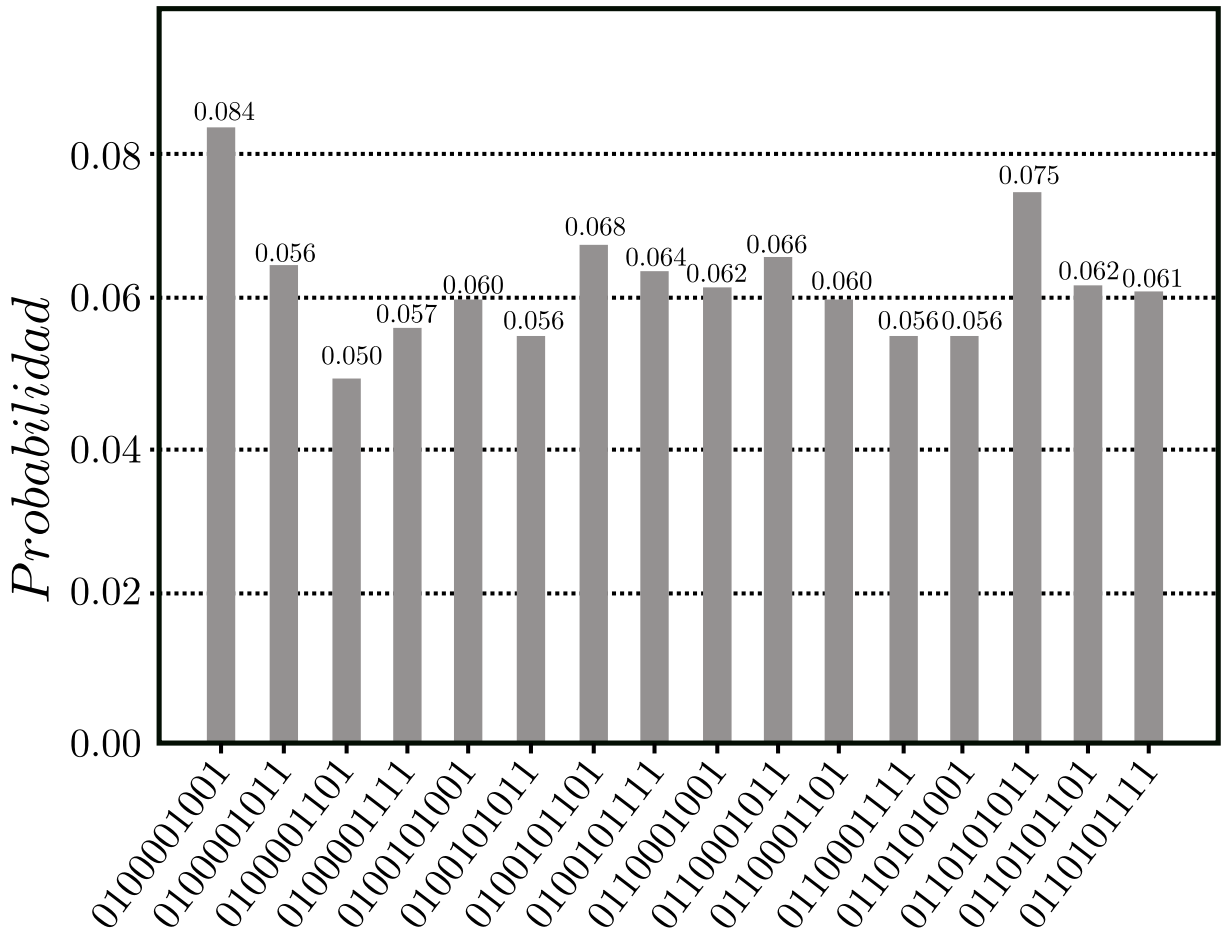

Figura. 5.10 Simulación de la cadena de salida del circuito cuántico de Eve: El histograma muestra la cadena de bits de salida del circuito cuántico de la figura 5.11 que Eve utiliza para medir los estados agrupados por bloque de 10 bots. El número de veces que se ejecuta el circuito es 1024, figura generada con el Qiskit.

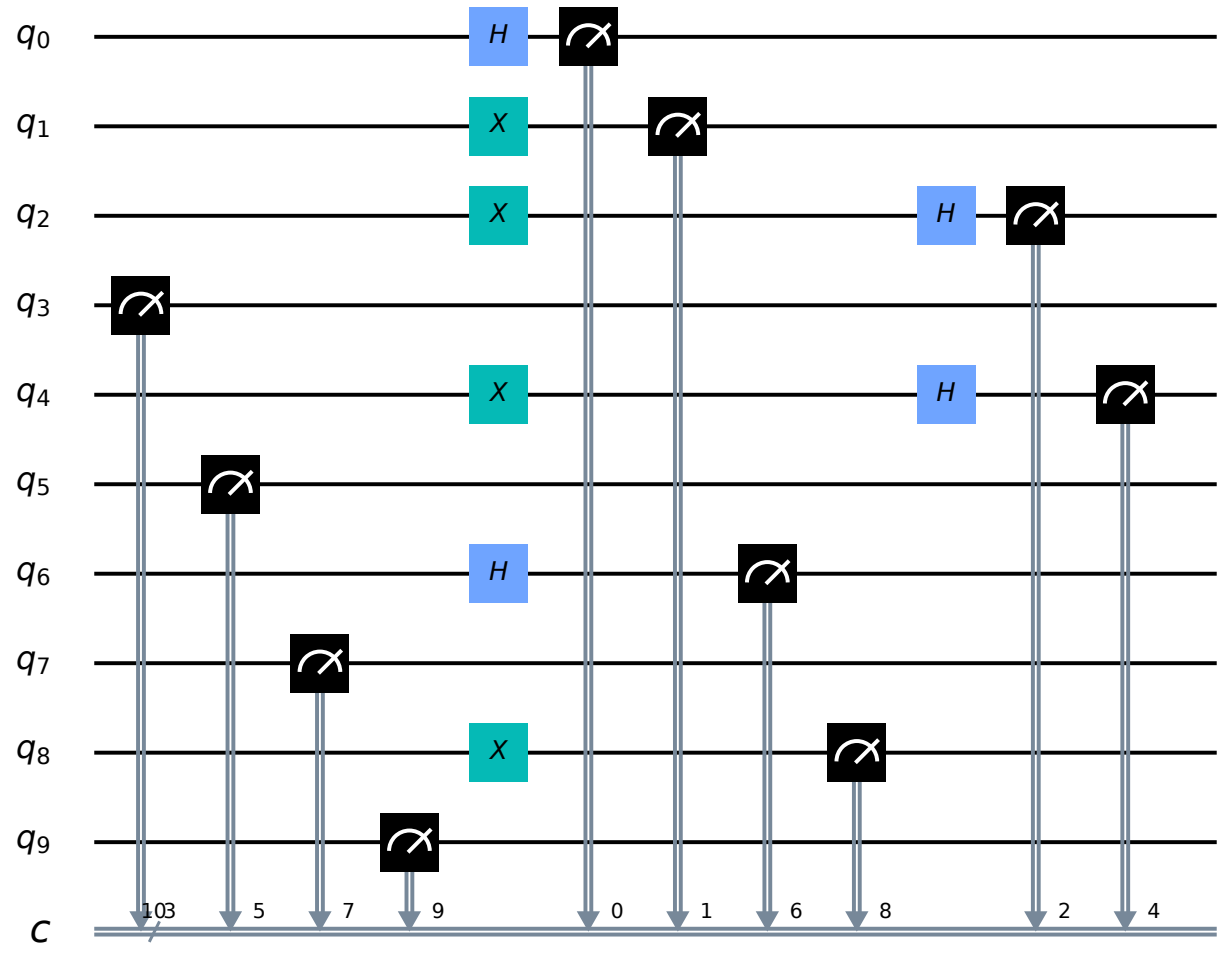

Figura. 5.11 Circuito cuántico para medir los estados entre Alice y Eve: El circuito cuántico inicializado en el estado $|0\rangle$ para que Eve empiece a medir los estados cuánticos trasmitidos por Alice, figura generada con el Qiskit.

Después que el intruso Eve mide y reenvía lo estados cuánticos a Bob, mide lo estados cuánticos utilizando el circuito cuántico de la figura 5.13. En este proceso Bob obtiene una cadena de binaria de rotación para las base y para las medidas, que en la figura 5.12 se muestra la simulación de la salida del circuito con un modelo de ruido y una repetición de 1024.

```
#inicia tu programa cuántico

backend = Aer.get_backend('qasm_simulator')
shots = 1
circuits = ['Eve2']

Bob_badresult = ''
for ind,l in enumerate(nlist):

#Definir las variables temporales utilizadas
```

```
#división del cuántico si la longitud del mensaje > 10

    if l < 10:

        key_temp = key[10*ind:10*ind+l]
        Eve_temp = Eve_result[10*ind:10*ind+l]
        Br_temp = Bob_rotate[10*ind:10*ind+l]

    else:

        key_temp = key[l*ind:l*(ind+1)]
        Eve_temp = Eve_result[l*ind:l*(ind+1)]
        Br_temp = Bob_rotate[l*ind:l*(ind+1)]

#inicia el resto de la información de tu circuito cuántico

    q = QuantumRegister(l, name='q')
    c = ClassicalRegister(l, name='c')
    Eve2 = QuantumCircuit(q , c, name='Eve2')

#preparar qubits

    for i,j,n in zip(Eve_temp,Br_temp,range(0,len(key_temp))):
        i = int(i)
        j = int(j)
        if i > 0:
            Eve2.x(q[n])
        if j > 0:
            Eve2.h(q[n])
        Eve2.measure(q[n],c[n])

    result_eve = execute(Eve2, backend, shots=shots).result()

    counts_eve = result_eve.get_counts()

    result_key_eve = list(result_eve.get_counts().keys())

    Bob_badresult += result_key_eve[0][::-1]

print("Los resultados anteriores de Bob (w/o Eve):",Bob_result)

print("Los resultados de bob de eva:\t\t",Bob_badresult)
```

```
#Ejecute el circuito en el simulador de qasm

backend_sim = BasicAer.get_backend('qasm_simulator')
```

```
# Hemos establecido el número de repeticiones del circuito.
# para ser 1024, que es el valor predeterminado.
job_sim = execute(Eve2, backend_sim, shots=1024)
#resultados de la simulación
result_sim = job_sim.result()
counts_Eve_Bob = result_sim.get_counts(Eve2)
print(counts_Eve_Bob)
plot_histogram(counts_Eve_Bob)
```

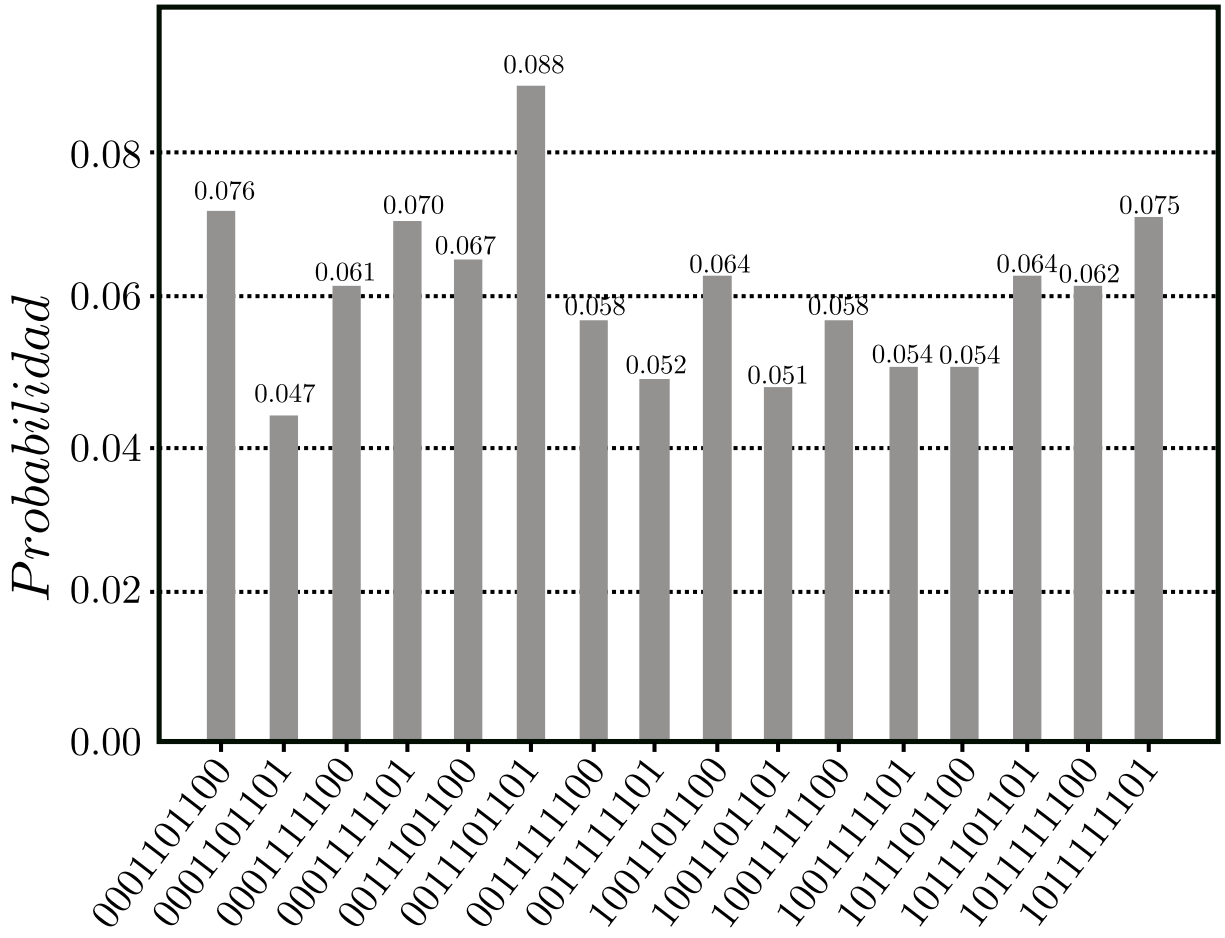

Figura. 5.12 Simulación de la cadena de salida del circuito cuántico de Bob con el intruso Eve: El histograma muestra la cadena de bits de salida del circuito cuántico de la figura 5.13, Bob mide los estados reenviados pro el intruso utiliza, las medidas estados agrupados por bloque de 10 bits y el número de veces que se ejecuta el circuito es 1024, figura generada con el Qiskit.

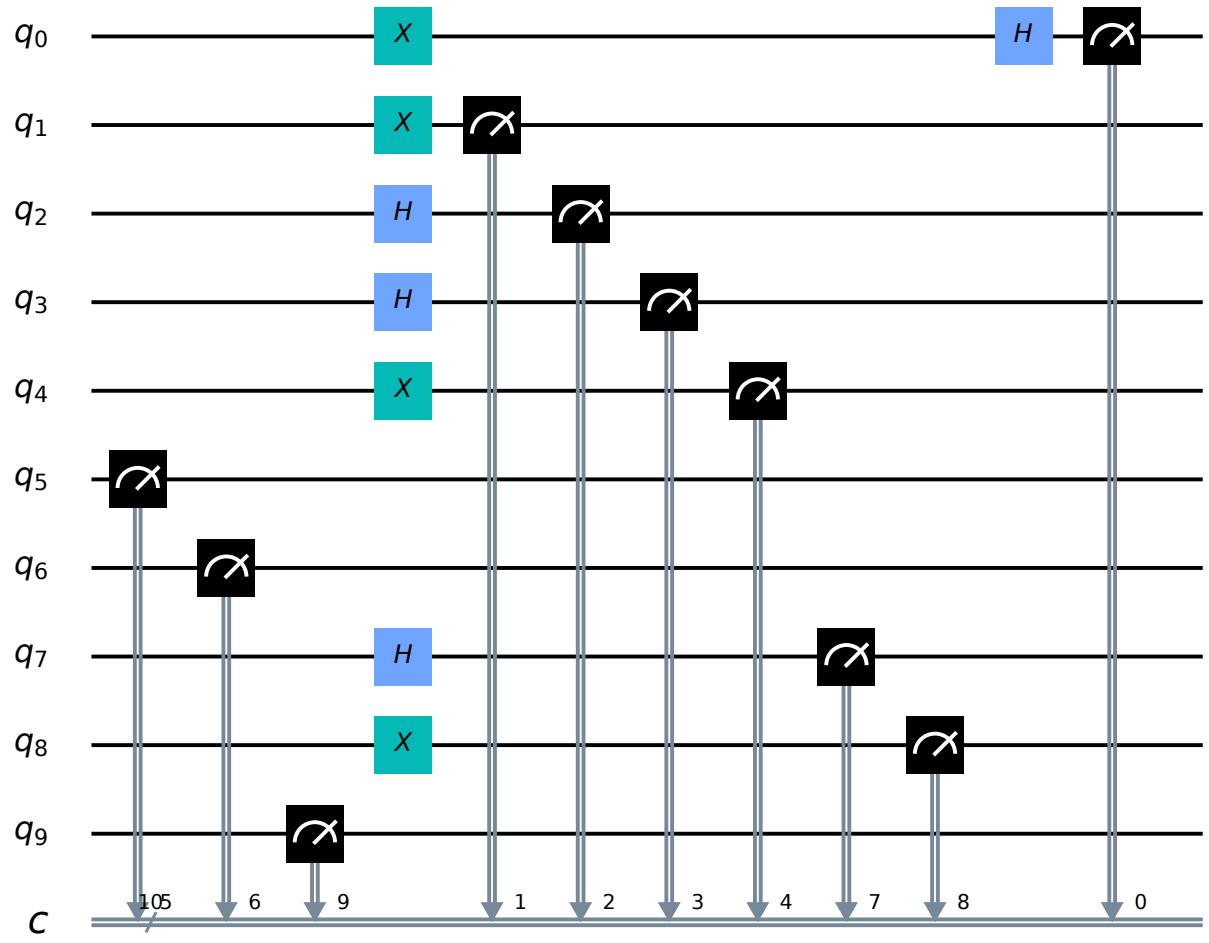

Figura. 5.13 Circuito cuántico para medir los estados entre Eve y Bob: El circuito cuántico inicializado en el estado $|0\rangle$ para que Bob empiece a medir los estados cuánticos retransmitidos por Eve, figura generada con el Qiskit.

Para comprobar que un intruso en el canal de comunicación altera los resultados de las cadenas de rotación de Bob, se muestran la medidas de los estados cuánticos echas por Bob sin intruso y con intruso en la transmisión.

- Resultados de la medidas de Bob sin intruso

  10110000001000101011011011111111111111010000011111001110101101001111000111110111011100100111001001000100010010000011110011011011100011000100011110011 0

- Resultados de la medidas de Bob con intruso:

  00110100100001110000101101111110000111011100110011001110010100011011100101101100001001000110010010101110110110011101111111010001101111001010100011001 0

Para realizar la reconciliación de la clave, como Alice y Bob no saben que el canal esta interceptado, entonces Bob y Alice se comparten las cadenas binarias de rotación que les permitirán saber la posición de los bit correctos de la clave. Pero como los estados fueron modificados por el intruso los posiciones correctas se modificaron y ya la clave de Alice y Bob no coinciden.

```
#hacer llaves para Alice y Bob

Akey = makeKey(Bob_rotate, Alice_rotate, key)
Bkey = makeKey(Bob_rotate, Alice_rotate, Bob_badresult)
print("clave de Alice:   ", Akey)
print("clave de Bob:      ", Bkey)

check_key = randomStringGen(len(Akey))

print('comprobar:', check_key)
```

- Clave de Alice con intruso

  1000100101111111111001011101011011011100000100001011010100100000110110

- Clave de Bob con intruso

  1100000100101111001101011010001011001100011110111111010101100100100010

Como las clave de Alice y Bob no coinciden, ellos sospechan que el canal de comunicación fue interceptado por un intruso, ya que aparecían más bit errores en la reconciliación de clave, estos Alice y Bob deciden compartir un subconjunto de las cadenas con las mediciones de los estados, para verificar la cantidad de errores y poder saber si el canal cuántico de comunicación está interceptado y poder abortar la comunicación.

```
#encontrar valores en la cadena de rotación utilizados para la clave

Alice_keyrotate = makeKey(Bob_rotate, Alice_rotate, Alice_rotate)
Bob_keyrotate = makeKey(Bob_rotate, Alice_rotate, Bob_rotate)

# Detectar la interferencia de Eva
#extraer un subconjunto de la clave de Alicia

sub_Akey = ''
sub_Arotate = ''
count = 0
for i,j in zip(Alice_rotate, Akey):
    if int(check_key[count]) == 1:
        sub_Akey += Akey[count]
        sub_Arotate += Alice_keyrotate[count]
    count += 1

#extraer un subconjunto de la clave de Bob

sub_Bkey = ''
sub_Brotate = ''
count = 0
```

```
for i,j in zip(Bob_rotate,Bkey):
    if int(check_key[count]) == 1:
        sub_Bkey += Bkey[count]
        sub_Brotate += Bob_keyrotate[count]
    count += 1

print("subconjunto de la llave de Alicia:",sub_Akey)
print("subconjunto de la llave de Bob:  ",sub_Bkey)

#compara la clave de Alicia y Bob

secure = True
for i,j in zip(sub_Akey,sub_Bkey):
    if i == j:
        secure = True
    else:
        secure = False
        break;
if not secure:
    print('Eve detecto!')
else:
    print('Eve escapó a la detección!')

#subcleve_Alice y subclave_Bob son de conocimiento público

if secure:
    new_Akey = ''
    new_Bkey = ''
    for index,i in enumerate(check_key):
        if int(i) == 0:
            new_Akey += Akey[index]
            new_Bkey += Bkey[index]
    print('new A and B keys: ',new_Akey,new_Bkey)
    if(len(mes)>len(new_Akey)):

        print('Tu nueva llave no es lo suficientemente larga.')
```

- Subconjunto de la llave de Alicia

  00001111110011011111001000011101010 01

- Subconjunto de la llave de Bob

  10001111011010001011111101111101110 00

- ąEve está presente en el canal de comunicación!

Con los anteriores resultados Alice y Bob saben con certeza que en el canal de comunicación cuántico existe un intruso que está tratando de obtener la clave criptográfica, entonces se aborta la

comunicación y no se genera la clave. Eve se detecta con una probabilidad $p = 1 - (3/4)^N, N = bit$ [194].

```
# gráfica de detección de un intruso

x = np.arange(0., 30.0)
y = 1-(3/4)**x

plt.plot(y)
plt.title('Detección de intruso en el cana de comunicación')
plt.xlabel('Longuitud en bits de la clave cuántica')
plt.ylabel('Probabilidad de detección de Eva')
plt.show()
```

En la figura 5.14 se muestra la probabilidad con respecta al numero de bit de la clave de que un ataque tenga éxito, se observa claramente que a medida que longitud de la clave crece la probabilidad de no detección de clave tiende a 1, lo que significa que entre mayor sea la longitud de la clave, menor será la posibilidad de que un intruso pueda obtener la clave cuántica.

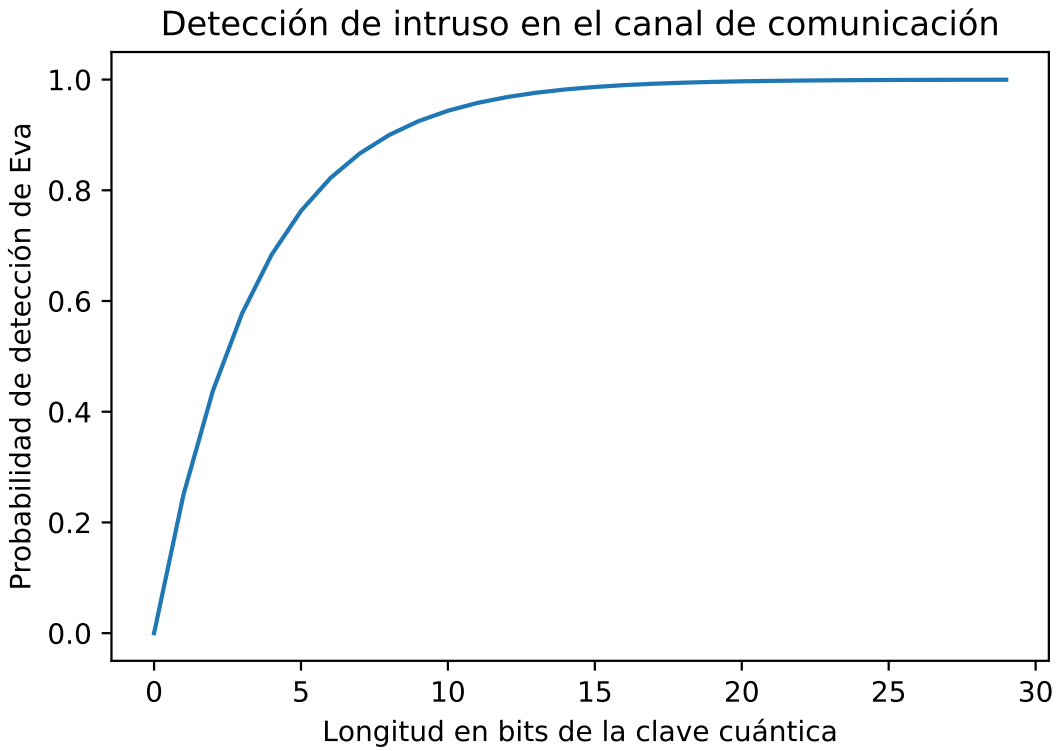

Figura. 5.14 Probabilidad de detectar un intruso en el canal cuántico de comunicación: en la figura se muestra que a medida que la longitud de la clave cuántica crece en número de bits, es mas probable detectar a Eve en al cana de comunicación, figura generada con el Qiskit.

5.3. Análisis de Resultados

La simulación realizada en este trabajo de investigación es una herramienta de gran utilidad para entender el comportamiento de la generación y distribución de claves cuánticas utilizando la polarización de fotones en el protocolo BB84. Además le permite a los investigadores en criptografía cuántica conocer características previas de los modelos a implementar como: la cantidad de qubits necesarios para detectar un intruso en el proceso de distribución de la clave, y realizar experimentación en los laboratorios con mas información recolectada.

En la fase 1 de la simulación se generó la clave de criptografía cuántica, y en el proceso lógico se observa que Alice prepara una secuencia n de bits que depende de la longitud del mensaje a cifrar, y se codifica en una polarización lineal de $0\,\%, 90\,\%$, o en una polarización diagonal $45\,\%, -45\,\%$ que se representan con los estados $|0\rangle, |1\rangle$. Esta secuencia es enviada a Bob por un canal cuántico de fibra óptica simulado, de donde se tiene que:

- Para el ejemplo de la simulación Alice envió 150 qubits a Bob que tiene una probabilidad de selección de las bases de $0,5$. Es decir, tiene la posibilidad de medir los estados entrantes con una de las dos bases posibles con igual probabilidad.
- Después que Bob mide los estados enviados, anuncia sobre un canal público clásico, en este caso simulado, los qubits que él ha medido acertadamente. Posteriormente Alice y Bob revelan las secuencias de las bases que usaron cada uno. Cuando las bases sean iguales, lo que ocurre aproximadamente en un $50\,\%$ de las veces, Alice y Bob toman ese qubit como parte de la clave personal, y al no tener ruido del canal, las dos claves resultantes deberían ser idénticas si no hay espía. Como se muestra en la secuencia de clave de la simulación que está formada por 75 bits.

En la fase 2 de la simulación se realiza el proceso de cifrado y descifrado de la información. Se toma la clave cuántica generada en la fase 1 y el mensaje a cifrar y se operan en una compuerta XOR que permite codificar. Luego para decodificar se opera también con una compuerta XOR el mensaje codificado con la misma clave para obtener el mensaje en claro.

Para la fase 3 de la simulación, se introdujo un espía Eve que interceptó el canal cuántico, para que mida y retransmita los estados enviados por Alice a Bob. En la intersección del canal Eve tiene una selección de base de 0,5, por tener también dos bases de medida. Por lo tanto Eve mide al azar lo estados enviados por Alice, proceso que transformo lo estados originales y envía a Bob una nueva secuencia de qubits correspondiente a sus medidas. Luego como Eve tiene también dos base para medir, entonces ella solo puede escoger la base correcta el $50\,\%$ de las veces, lo que genera que al menos el 0,25 de sus qubits sean diferentes a los que son enviados por Alice.

De las cadenas binarias que resultan en la simulación se puede calcular la probabilidad de acierto que tiene Eve cuando trata de interceptar y reenviar los bits de información. Denotando por x es el bit de Alice, x'' el bit de Bob, con una base común B para Alice y Bob, y x' es el bit medido por Eve con una base B', entonces la probabilidad de coincidencia de las bit entre Alice y Eve se puede calcular como:

$$\begin{aligned} P\big(x = x'\big) &= P\big(x = x'/B = B'\big) + P\big(x = x'/B \neq B'\big)P\big(B \neq B'\big); \\ &= 1 * \left(\frac{1}{2}\right) + \left(\frac{1}{2}\right) * \left(\frac{1}{2}\right); \\ &= 0{,}75. \end{aligned}$$

Además para ver como afecta la medición de Eve la generación de clave, se puede calcular la coincidencia de los bits entre Alice y Bob mediante la siguiente expresión:

$$\begin{aligned} P\big(x = x''\big) &= P\big(x = x''/B = B'\big)P\big(B = B'\big) + P\big(x = x''/B \neq B'\big)P\big(B \neq B'\big); \\ &= 1 * \left(\frac{1}{2}\right) + \left(\frac{1}{2}\right) * \left(\frac{1}{2}\right); \\ &= 0{,}75. \end{aligned}$$

Por lo tanto se observa que cuando existe un intruso que intercepta y reenvía los bits, éste tiene un 75 % de acierto, lo que significa que el intruso introduce como mínimo un error en la generación y distribución de la clave cuántica del 25 % en el receptor.

CAPÍTULO 6

CONCLUSIONES

En esta investigación se presentan los análisis y resultados de un método de criptografía cuántica, que reúne un conjunto de principios, propiedades y definiciones de sistemas cuánticos, que están siendo estudiados e integrados a los modelos de seguridad de la información, para mejorar la integridad y autenticidad de los mensajes trasmitidos en las redes de comunicación. Este método se simula usando la base teoría de la generación y distribución de claves cuánticas del protocolo BB84, y el cifrdor de Vernam que permiten cifrar información de modo seguro, y ademas se empieza a superar los problemas de la criptografía moderna puede ser vulnerada por los algoritmos cuánticos de Peter Shor y Lov K. Grover implementados en computadores cuánticos, ya que permiten reducir los tiempos de procesamiento de forma exponencial a polinomial, y permite realizar ataques en tiempo de ejecución para obtener las claves de cifrado construidas con los algoritmos clásicos. Lo resultados se muestran a partir de simulaciones computacionales desarrollas en Qiskit, por lo tanto las principales conclusiones que deja este trabajo de investigación son:

- El estudio de los sistemas cuántica, nos permitió entender que existen principios y propiedades físicas, como la superposición, la no clonación, y el entrelazamiento que permiten construir sistemas criptográficos cuánticos con una alta probabilidad de seguridad en el proceso de cifrado de la información. Ademas la criptografía cuántica es la aplicación de la teoría de información cuántica que más se a desarrollado en la ultima década, se esta pasando de las pruebas de laboratorio a la integración con los dispositivos informáticos.
- La integración del protocolo cuántico BB84 con el cifrado de Vernam, genera un secreto perfecto, ya que se pueden tener claves completamente aleatorias para tener una mejor seguridad cuando se va ha codificar información de modo simétrico.
- La criptografía cuántica es la solución a los problemas de seguridad de la información, esta garantiza la integridad de la información en los sistemas de comunicación, ademas no se necesitan entidades certificadoras para generar confianza, porque con la distribución de claves cuánticas, el transmisor y el receptor siempre saben si se intercepto la comunicación.
- Otra de las conclusiones de este trabajo de investigación, es que se puede decir con certeza que la generación y distribución de claves cuánticas es una poderosa herramienta dentro de la transmisión de información de manera segura, y que va a ser implementada en los sistemas de comunicación futuros. Esto, porque permite conciliar una clave segura que será utilizada en la transmisión de un mensajes privados independiente de cualquier valor de entrada, hecho que es imposible en la criptografía con base en estados clásicos. En la criptografía cuántica se le otorga la habilidad al sistema de detectar automáticamente cualquier interceptación en la transmisión, aumentando la fiabilidad, y la posibilidad de no tener que autenticar el canal.

- Se ha mostrado que un sistema de generación y distribución de claves cuánticas puede predecir la existencia de un intruso en la comunicación, antes de codificar la información. Pues el intruso al tratar de medir un estado compartido entre Alice y Bob, genera errores detectables en el sistema que conlleva a que se aborte la transmisión de clave.

- El lenguaje de programación Qiskit que se utilizó en este trabajo de investigación, es una de las herramientas computaciones más completas para simular algoritmos cuánticos. Con este lenguaje de programación se pueden manipular programas cuánticos, siguiendo el modelo de circuito para la computación cuántica universal, y puede usarse para cualquier hardware cuántico que este construido con este modelo. Además, es sencillo de utilizar por que opera en cualquier entorno Python, cuadernos Jupyter, Swift, o JavaScript. Cuando se trabaja computación cuántica con el Qiskit y Python, los procesos se entienden fácilmente por ser un lenguaje de programación muy intuitivo.

- Se ha analizado la capacidad que tiene un intruso de obtener la clave cuántica en el protocolo BB84, mediante un ataque de interceptación y reenvió de información en el canal cuántico. Se llegó a la conclusión de que si existe un intruso, le es imposible obtener la clave cuántica en este protocolo, siempre y cuando ésta tenga una longitud mayor o igual que el mensaje que se quiere codificar. Además, el transmisor y el receptor se pueden dar cuenta con certeza si existe un intruso cuando se está en el proceso de generación y transmisión de la clave cuántica, característica que hace superior a la criptografía cuántica. Sin embargo, un intruso si puede generar una denegación de servicios en el sistema, como se mencionó en el trabajo.

- El aporte de seguridad de la mecánica cuántica a la criptografía se refleja cuando un intruso al intentar extraer información del sistema revela su presencia en la comunicación, pues al intentar medir un estados cuántico éste se modifica irreversiblemente, ya que los estados cuánticos no pueden ser copiados o modificados sin alterarlos.

6.1. Trabajos Futuros

Sin duda uno de los campos de investigación de la teoría de información y computación cuánticas está concentrado en los procesos de generación y distribución de claves cuánticas, puesto que estos modelos permiten preservar la seguridad de la información. Por esto, del presente trabajo se pueden desprender las siguientes líneas a trabajar en el futuro:

- Realizar la implementación del método criptográfico simulado en los computadores cuánticos de IBM. Para esto se debe desarrollar el método con compuertas cuánticas y utilizando las herramientas que ofrece el lenguaje de programación Qiskit para realizar operaciones en circuitos cuánticos universales.

- Realizar la implementación experimental del método criptográfico desarrollado en un laboratorio de óptica cuántica. Para esto, se debe utilizar como canal de comunicación cuántico una fibra óptica que permitirá transmitir los estados cuánticos codificados, y como canal clásico de comunicación se utilizar un cable coaxial para realizar la comparación y reconciliación de la mediciones hechas por el receptor y el transmisor. Sin embargo, antes se debe estudiar y entender la evolución dinámica de las correlaciones y decoherencias en los canales cuánticos de comunicación, que permitirán obtener las condiciones necesarias para trasferir la información en forma óptima.

- Otra posible aplicación de este trabajo de investigación es realizar la implantación experimental de este método criptográfico utilizando codificación por frecuencia y usando el espacio libre como canal de comunicación cuántico. Para la radiación de lo estados cuánticos codificados en frecuencia se puede utilizar en el transmisor una antena que permita radiar los qubits en el espacio libre. Similarmente en el receptor se tendrá dispuesta una antena con las mismas especificaciones que las del transmisor. Estas antenas formarán un radio enlace que en primera primera instancia estará separado una distancia de 100 metros, y como el canal clásico de comunicación será el mismo espacio libre pero configurado en otra frecuencia, estará certificado para prevenir una posible interceptación de las claves cuánticas.

APÉNDICE A

QISKIT

Qiskit es un entornó de simulación de código abierto construido para la computación cuántica. Posee herramientas para crear y manipular programas cuánticos, y sus algoritmos se pueden probar en los prototipos de computadores cuánticos, que siguen el modelo de circuito cuántica universal. Qiskit fue fundada por IBM Research para permitir el desarrollo de software para su servicio de computación cuántica en la nube. La versión principal de Qiskit usa el lenguaje de programación Python, pero ya existen versiones para Swift y JavaScript.

Lo primero que tiene que hacer para instala Qiskit es verificar los requisitos mínimos de la maquina.

requisitos

- Instalar Python 3,5 o posterior.
- Instalar Anaconda, para pode usar la multiplataforma de distribución de Python para computación científica. Jupyter Notebook.

Qiskit es compatible con los siguientes sistemas de 64 bits:

- Ubuntu 16,04 o posterior
- macOS 10,12,6 o posterior
- Windows 7 o posterior

Cuando se usa Windows requiere componentes de tiempo de ejecución de $VC++$. Entonces se tiene que instalar

- Microsoft Visual $C++$ Redistributable para Visual Studio 2017
- Microsoft Visual $C++$ Redistributable para Visual Studio 2015

Instalación

Se puede instalar: una es en entornos virtuales Python para separar Qiskit de otras aplicaciones y mejorar su experiencia. Se utiliza el comando conda, incluido con Anaconda y se utilizan los comandos por terminal.

- $condacreate - nname_of_my_envpython = 3$
- $sourceactivatename_of_my_env$

Para el sistema operativo Windows, usa el siguiente comando.

- $activatename_of_my_env$

Posteriormente también se pueden instalar los paquetes de Qiskit, que incluye Terra, Aer, Ignis y Aqua. Con pip.

- $pipinstallqiskit$

Por último si se requieren se pueden instalar dependencias opcionales con el siguiente comando.

- $pipinstallqiskit - terra[visualization]$

zdespués que se tengan todos los paquetes de Qiskit, se puede trabajar en el entorno Jupyter Notebook, de se puede desarrollar mas fáciles simulaciones cuánticas. Para más información de instalación de ejemplos pude visitar la pagina en la red, https://qiskit.org/documentation/install.html.

BIBLIOGRAFÍA

[1] Robert Wille, Rod Van Meter, and Yehuda Naveh. IBM qiskit tool chain: working with and developing for real quantum computers. In *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1234–1240. IEEE, 2019.

[2] Mohammed Farik and Shawkat Ali. The need for quantum-resistant cryptography in classical computers. In *2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, pages 98–105. IEEE, 2016.

[3] Alberto Porzio. Quantum cryptography: approaching communication security from a quantum perspective. In *2014 Fotonica AEIT Italian Conference on Photonics Technologies*, pages 1–4. IEEE, 2014.

[4] Lidong Chen. Cryptography standards in quantum time: new wine in old wineskin. *IEEE Security & Privacy*, 15(4):51, 2017.

[5] Claude E Shannon and Warren Weaver. The mathematical theory of communication (urbana, il, 1949.

[6] Luis de la Peña. *Introducción a la mecánica cuántica*. Fondo de Cultura Económica, 2014.

[7] O Organista, V Gómez, D Jaimes, and J Rodríguez. Una idea profunda en la comprensión del mundo físico: el principio de superposición de estados. *Latin American Journal of Physics Education*, 1:83–88, 2007.

[8] Seth Lloyd. Ultimate physical limits to computation. *Nature*, 406(6799):1047, 2000.

[9] Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete. *Physical Review*, 47(10):777, 1935.

[10] Erwin Schrödinger. Discussion of probability relations between separated systems. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 555–563. Cambridge University Press, 1935.

[11] E Schrödinger. Mathematical proceedings of the cambridge philosophical society. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 555–563, 1935.

[12] John S. Bell. *Speakable and unspeakable in quantum mechanics: collected papers on quantum philosophy*. Cambridge University Press, 2004.

[13] Alain Aspect, Philippe Grangier, and Gérard Roger. Experimental realization of Einstein-Podolsky-Rosen-Bohm Gedanken experiment: a new violation of IBMell's inequalities. *Physical Review Letters*, 49(2):91, 1982.

[14] John S Bell. On the Einstein Podolsky Rosen paradox. *Physics*, 1(3):195, 1964.

[15] Olmo Nieto-Silleras, Cédric Bamps, Jonathan Silman, and Stefano Pironio. Device-independent randomness generation from several Bell estimators. *New Journal of Physics*, 20(2):023049, 2018.

[16] Punit Chaudhury, Susmita Dhang, Monpreet Roy, Saurav Deb, Jyotirmoy Saha, Aditya Mallik, Sauvik Bal, Saraswata Roy, Mrinal Kanti Sarkar, Sanjay Kumar, et al. Acafp: Asymmetric key based cryptographic algorithm using four prime numbers to secure message communication. a review on RSA algorithm. In *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*, pages 332–337. IEEE, 2017.

[17] Guang-liang Guo, Quan Qian, and Rui Zhang. Different implementations of aes cryptographic algorithm. In *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, pages 1848–1853. IEEE, 2015.

[18] Alan G Konheim. Hashing functions: examples and evaluation. 2010.

[19] Lov K Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2):325, 1997.

[20] Lily Chen, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. *Report on quantum cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016.

[21] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE, 1994.

[22] Matthew Edward Briggs. *An introduction to the general number field sieve*. PhD thesis, Virginia Tech, 1998.

[23] David Biron, Ofer Biham, Eli Biham, Markus Grassl, and Daniel A Lidar. Generalized Grover search algorithm for arbitrary initial amplitude distribution. In *NASA International Conference on Quantum Computing and Quantum Communications*, pages 140–147. Springer, 1998.

[24] Charles H Bennett, Gilles Brassard, and Artur K Ekert. Quantum cryptography. *Scientific American*, 267(4):50–57, 1992.

[25] Hitesh Singh, DL Gupta, and AK Singh. Quantum key distribution protocols: a review. *Journal of Computer Engineering*, 16(2):1–9, 2014.

[26] Kausik Saha, Sirshendu Sekhar Ghosh, and Dilip Kumar Shaw. Quantum key distribution scheme: an improvement based on BB84 protocol. *International Journal of Advanced Research in Computer Science*, 9(2):287, 2018.

[27] Akshata Shenoy-Hejamadi, Anirban Pathak, and Srikanth Radhakrishna. Quantum cryptography: key distribution and beyond. *Quanta*, 6(1):1–47, 2017.

[28] Sheng-Kai Liao, Hai-Lin Yong, Chang Liu, Guo-Liang Shentu, Dong-Dong Li, Jin Lin, Hui Dai, Shuang-Qiang Zhao, Bo Li, Jian-Yu Guan, et al. Long-distance free-space quantum key distribution in daylight towards inter-satellite communication. *Nature Photonics*, 11(8):509, 2017.

[29] Margaret B Cozzens, Steven J Miller, and Steven J Miller. *The mathematics of encryption: an elementary introduction*. American Mathematical Society Providence, 2013.

[30] M Rathidevi, R Yaminipriya, and SV Sudha. Trends of cryptography stepping from ancient to modern. In *2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT)*, pages 1–9. IEEE, 2017.

[31] José Manuel Sánchez Muñoz. Criptología nazi. los códigos secretos de hitler. *Pensamiento Matemático*, 3(1):59–120, 2013.

[32] Kashish Goyal and Supriya Kinger. Modified caesar cipher for better security enhancement. *International Journal of Computer Applications*, 73(3):0975–8887, 2013.

[33] Santiago Fernández. La criptografía clásica. *Sigma: Revista de Matemáticas*, 2004.

[34] Osama S Faragallah, Fathi E Abd El-Samie, Hossam Eldin H Ahmed, Ibrahim F Elashry, Mai H Shahieen, El-Sayed M El-Rabaie, and Saleh A Alshebeili. *Image encryption: a communication perspective*. CRC Press, 2013.

[35] Konstantin Lisickiy, Viktor Dolgov, and Iryna Lisickaya. Cipher with improved dynamic indicators of the condition of a random substitution. In *2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, pages 396–399. IEEE, 2017.

[36] Christian Hill. *Learning scientific programming with Python*. Cambridge University Press, 2016.

[37] Aditi Bhateja and Shailender Kumar. Genetic algorithm with elitism for cryptanalysis of vigenere cipher. In *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pages 373–377. IEEE, 2014.

[38] Gilbert S Vernam. Cipher printing telegraph systems: For secret wire and radio telegraphic communications. *Journal of the AIEE*, 45(2):109–115, 1926.

[39] Alfonso Muños and Sergienko Aguire, Jorge. *Cifrado de las cominicaciones dijitales de la cifra clásica al algoritmo RSA*. OxWURD, 2013.

[40] Simon Singh. The code book: the science of secrecy from ancient egypt to quantum cryptography. *Swiat Ksiazki*, pages 19–21, 2003.

[41] Marian Rejewski. How polish mathematicians deciphered the enigma. *Annals of the History of Computing*, 3(3):213–234, 1981.

[42] Marian Rejewski. Mathematical solution of the enigma cipher. *Cryptologia*, 6(1):1–18, 1982.

[43] Deniz Engin and Berna Ors. Implementation of enigma machine using verilog on an FPGA. In *2015 9th International Conference on Electrical and Electronics Engineering (ELECO)*, pages 945–948. IEEE, 2015.

[44] Andrew Hodges. *Alan Turing: the Enigma: the Enigma*. Random House, 2012.

[45] Peter Donovan. The flaw in the jn-25 series of ciphers, ii. *Cryptologia*, 36(1):55–61, 2012.

[46] Simon Singh. *The code book*, volume 7. Doubleday New York, 1999.

[47] Claude E Shannon. Communication theory of secrecy systems. *Bell system technical journal*, 28(4):656–715, 1949.

[48] Claude E Shannon and Warren Weaver. *The mathematical theory of communication*. University of Illinois Press, 1998.

[49] Gurpreet Singh. A study of encryption algorithms (RSA, DES, 3DES, AES) for information security. *International Journal of Computer Applications*, 67(19), 2013.

[50] Sourabh Chandra, Smita Paira, Sk Safikul Alam, and Goutam Sanyal. A comparative survey of symmetric and asymmetric key cryptography. In *2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE)*, pages 83–93. IEEE, 2014.

[51] Bidisha Mandal, Sourabh Chandra, Sk Safikul Alam, and Subhendu Sekhar Patra. A comparative and analytical study on symmetric key cryptography. In *2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE)*, pages 131–136. IEEE, 2014.

[52] Tingyuan Nie and Teng Zhang. A study of DES and blowfish encryption algorithm. In *Tencon 2009-2009 IEEE Region 10 Conference*, pages 1–4. IEEE, 2009.

[53] Phil Karn, Perry Metzger, and William Simpson. The ESP triple DES transform. Technical report, 1995.

[54] John Kelsey, Bruce Schneier, and David Wagner. Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and triple-DES. In *Annual International Cryptology Conference*, pages 237–251. Springer, 1996.

[55] Monk-Ping Leong, Ocean YH Cheung, Kuen Hung Tsoi, and Philip Heng Wai Leong. A bit-serial implementation of the international data encryption algorithm IDEA. In *Proceedings 2000 IEEE Symposium on Field-Programmable Custom Computing Machines (Cat. No. PR00871)*, pages 122–131. IEEE, 2000.

[56] Hemant K Singh, Amitay Isaacs, Tapabrata Ray, and Warren Smith. Infeasibility driven evolutionary algorithm IDEA for engineering design optimization. In *Australasian Joint Conference on Artificial Intelligence*, pages 104–115. Springer, 2008.

[57] Stefan Wolter, Holger Matz, Andreas Schubert, and Rainer Laur. On the vlsi implementation of the international data encryption algorithm IDEA. In *Proceedings of ISCAS'95-International Symposium on Circuits and Systems*, volume 1, pages 397–400. IEEE, 1995.

[58] C Burnwick and Don Coppersmith. The mars encryption algorithm. *IBM, http://csrc. nist. gov/encryption/aes/round2/AESAIgs/MARS*, 1999.

[59] Nawal F El Fishawy and Osama M Abu Zaid. Quality of encryption measurement of bitmap images with RC6, MRC6, and Rijndael block cipher algorithms. *IJ Network Security*, 5(3):241–251, 2007.

[60] Ross Anderson1 Eli Biham2 Lars Knudsen. Serpent: a proposal for the advanced encryption standard. In *First Advanced Encryption Standard (AES) Conference, Ventura, CA*, 1998.

[61] Harsh Kumar Verma and Ravindra Kumar Singh. Performance analysis of RC6, Twofish and Rijndael block cipher algorithms. *International Journal of Computer Applications*, 42(16):1–7, 2012.

[62] NIST FIPS Pub. Advanced encryption standard AES. *Federal Information Processing Standards Publication*, 197(441):0311, 2001.

[63] Mahmoud Alfadel, El-Sayed M El-Alfy, and Khaleque Md Aashiq Kamal. Evaluating time and throughput at different modes of operation in AES algorithm. In *2017 8th International Conference on Information Technology (ICIT)*, pages 795–801. IEEE, 2017.

[64] AES Primitives. Advanced encryption standard AES (FIPS-197. 2003.

[65] Priyadarshini Patil, Prashant Narayankar, DG Narayan, and S Md Meena. A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish. *Procedia Computer Science*, 78:617–624, 2016.

[66] Oscar Casas García. Implementación de los cifradores de bloque Rijndael, Serpent, MARS, Ttwofish y RC6 para su uso en sistemas embebidos. 2010.

[67] Yousuf Alsalami, Chan Yeob Yeun, T Martin, and Majid Khonji. Linear and differential cryptanalysis of small-sized random (n, m)-s-boxes. In *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 447–454. IEEE, 2016.

[68] Howard Heys. Integral cryptanalysis of the BSPN block cipher. In *2014 27th Biennial Symposium on Communications (QBSC)*, pages 153–158. IEEE, 2014.

[69] Tonmoy Dhar, Swarup Bhunia, and Amit Ranjan Trivedi. A solitary protection measure against scan chain, fault injection, and power analysis attacks on AES. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 575–578. IEEE, 2017.

[70] Davide Bellizia, Simone Bongiovanni, Pietro Monsurrò, Giuseppe Scotti, and Alessandro Trifiletti. Univariate power analysis attacks exploiting static dissipation of nanometer CMOS, VLSI circuits for cryptographic applications. *IEEE Transactions on Emerging Topics in Computing*, 5(3):329–339, 2016.

[71] Davood Shanbehzadeh and Mohammad Reza Bagheri. Amplified template attack of cryptographic algorithms. In *2017 7th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 142–147. IEEE, 2017.

[72] Massimo Alioto, Massimo Poli, and Santina Rocchi. Differential power analysis attacks to precharged buses: a general analysis for symmetric-key cryptographic algorithms. *IEEE Transactions on Dependable and Secure Computing*, 7(3):226–239, 2009.

[73] Michael Tunstall. Improved partial sums-based square attack on AES. In *SECRYPT*, pages 25–34, 2012.

[74] Kai Schramm, Gregor Leander, Patrick Felke, and Christof Paar. A collision-attack on AES. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 163–175. Springer, 2004.

[75] Yogesh Kumar, Rajiv Munjal, and Harsh Sharma. Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures. *International Journal of Computer Science and Management Studies*, 11(03), 2011.

[76] Shilpi Gupta and Jaya Sharma. A hybrid encryption algorithm based on RSA and diffie-hellman. In *2012 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–4. IEEE, 2012.

[77] Richard A Mollin. RSA *and public-key cryptography*. Chapman and Hall/CRC, 2002.

[78] Susanna S Epp. *Discrete mathematics with applications*. Cengage Learning, 2010.

[79] Severino Collier Coutinho. *The mathematics of ciphers: number theory and* RSA *cryptography*. AK Peters/CRC Press, 1999.

[80] Oded Goldreich. *Foundations of cryptography: volume 2, basic applications*. Cambridge University Press, 2009.

[81] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC Press, 2014.

[82] Joseph H Silverman. *Advanced topics in the arithmetic of elliptic curves*, volume 151. Springer Science & Business Media, 2013.

[83] Louis Goubin. A refined power-analysis attack on elliptic curve cryptosystems. In *International Workshop on Public Key Cryptography*, pages 199–211. Springer, 2003.

[84] Andrew Sutherland. 18.783 elliptic curves lecture, 2015.

[85] Nigel P Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of cryptology*, 12(3):193–196, 1999.

[86] Yogesh Malhotra. Bitcoin protocol: model of cryptographic proof based blobal crypto-currency & electronic payments system. 2013.

[87] Soram Ranbir Singh, Ajoy Kumar Khan, and Takhellambam Sonamani Singh. A critical review on elliptic curve cryptography. In *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pages 13–18. IEEE, 2016.

[88] Nigel P Smart. The exact security of ECIES in the generic group model. In *IMA International Conference on Cryptography and Coding*, pages 73–84. Springer, 2001.

[89] Don B Johnson and Alfred J Menezes. Elliptic curve DSA, ECDSA: an enhanced DSA. In *Proceedings of the 7th conference on USENIX Security Symposium*, volume 7, pages 13–23, 1998.

[90] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.

[91] Eike Kiltz and Krzysztof Pietrzak. Leakage resilient elgamal encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 595–612. Springer, 2010.

[92] Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K Lenstra, Emmanuel Thomé, Joppe W Bos, Pierrick Gaudry, Alexander Kruppa, Peter L Montgomery, Dag Arne Osvik, et al. Factorization of a 768-bit RSA modulus. In *Annual Cryptology Conference*, pages 333–350. Springer, 2010.

[93] Jesús David Granados Ávila et al. *Factorización prima de números naturales para estudiantes del tercer ciclo*. PhD thesis, Universidad Nacional de Colombia, 2011.

[94] R Sherman Lehman. Factoring large integers. *Mathematics of Computation*, 28(126):637–646, 1974.

[95] Muños Alfonso and Ramió Jorge. *Cifrado de las cominicaciones dijitales de la cifra clásica al algoritmo* RSA. OxWURD Computing, 2013.

[96] Bruce Schneier. Cryptographic design vulnerabilities. *Computer*, (9):29–33, 1998.

[97] Tamara Radivilova and Hassan Ali Hassan. Test for penetration in wi-fi network: attacks on WPA2-PSK and WPA2-enterprise. In *2017 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)*, pages 1–4. IEEE, 2017.

[98] Timothy A Hall. The fips 186-4 digital signature algorithm validation system DSA2VS. 2010.

[99] Alejandro Cabrera Aldaya, Cesar Pereida García, Luis Manuel Alvarez Tapia, and Billy Bob Brumley. Cache-timing attacks on RSA key generation. *IACR Cryptology Eprint Archive*, 2018:367, 2018.

[100] Javier Herranz, A Ruiz, and Germán Sáez. Esquemas de firma digital con verificación distribuida. 2008.

[101] Jonathan Emmett, Philip Allan Eisen, James Muir, and Daniel Murdock. Method and system for protecting execution of cryptographic hash functions, September 13 2016. US Patent 9,443,091.

[102] Daniel V Bailey, William M Duane, and Aaron Katz. Protected resource access control utilizing credentials based on message authentication codes and hash chain values, March 17 2015. US Patent 8,984,602.

[103] Justin Fisher and Maxwell Henry Sanchez. Authentication and verification of digital data utilizing blockchain technology, September 29 2016. US Patent App. 15/083,238.

[104] Satoshi Nakamoto et al. Bitcoin: a peer-to-peer electronic cash system. 2008.

[105] Danny Bradbury. The problem with bitcoin. *Computer Fraud & Security*, 2013(11):5–8, 2013.

[106] Bart Preneel. *Analysis and design of cryptographic hash functions*. PhD thesis, Citeseer, 1993.

[107] J Lawrence Carter and Mark N Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[108] Ronald Rivest. The MD5 message-digest algorithm. 1992.

[109] William Stallings, Simon Singh, and Jonathan Dowling. Cryptography and network security, 7/e. 2003.

[110] Rolando P Reyes Ch et al. How easy is to break password protection: a preliminary empirical study. In *2016 IEEE Ecuador Technical Chapters Meeting (ETCM)*, pages 1–6. IEEE, 2016.

[111] Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. Analysis of SHA-512/224 and SHA-512/256. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 612–630. Springer, 2015.

[112] Morris J Dworkin. Sha-3 standard: permutation-based hash and extendable-output functions. Technical report, 2015.

[113] Lenos Ioannou, Harris E Michail, and Artemios G Voyiatzis. High performance pipelined FPGA implementation of the SHA-3 hash algorithm. In *2015 4th Mediterranean Conference on Embedded Computing (MECO)*, pages 68–71. IEEE, 2015.

[114] Victor Shoup. *Advances in cryptology-Crypto 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621. Springer Science & Business Media, 2005.

[115] Mridul Nandi and Douglas R Stinson. Multicollision attacks on generalized hash functions. *IACR Cryptology ePrint Archive*, 2004:330, 2004.

[116] Lars R Knudsen, Xuejia Lai, and Bart Preneel. Attacks on fast double block length hash functions. *Journal of Cryptology*, 11(1):59–72, 1998.

[117] Yahui Wang, Huanguo Zhang, and Houzhen Wang. Quantum polynomial-time fixed-point attack for RSA. *China Communications*, 15(2):25–32, 2018.

[118] Xiaoyang Dong, Bingyou Dong, and Xiaoyun Wang. Quantum attacks on some feistel block ciphers. *IACR Cryptology ePrint Archive*, 2018:504, 2018.

[119] Daniel P Martin, Ashley Montanaro, Elisabeth Oswald, and Dan Shepherd. Quantum key search with side channel advice. In *International Conference on Selected Areas in Cryptography*, pages 407–422. Springer, 2017.

[120] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge University Press, 10th anniversary edition, 2010.

[121] John Preskill. Lecture notes for physics 229: quantum information and computation. *California Institute of Technology*, 16, 1998.

[122] Jun John Sakurai, Jim Napolitano, et al. *Modern quantum mechanics*, volume 185. Pearson Harlow, 2014.

[123] Gordon Baym. *Lectures on quantum mechanics*. CRC Press, 2018.

[124] Masanao Ozawa. Physical content of Heisenberg's uncertainty relation: limitation and reformulation. *Physics Letters A*, 318(1-2):21–29, 2003.

[125] Masanao Ozawa. Operations, disturbance, and simultaneous measurability. *Physical Review A*, 63(3):032109, 2001.

[126] Masanao Ozawa. Uncertainty relations for joint measurements of noncommuting observables. *Physics Letters A*, 320(5-6):367–374, 2004.

[127] A Ruschhaupt, Xi Chen, D Alonso, and JG Muga. Optimally robust shortcuts to population inversion in two-level quantum systems. *New Journal of Physics*, 14(9):093040, 2012.

[128] Myranda Uselton. Investigating quantum computation. *Scientia et Humanitas*, 9:57–69, 2019.

[129] Rémy Mosseri and Rossen Dandoloff. Geometry of entangled states, bloch spheres and hopf fibrations. *Journal of Physics A: Mathematical and General*, 34(47):10243, 2001.

[130] A Cuevas, G Carvacho, G Saavedra, J Cariñe, WAT Nogueira, M Figueroa, Adan Cabello, P Mataloni, G Lima, and GB Xavier. Long-distance distribution of genuine energy-time entanglement. *Nature Communications*, 4:2871, 2013.

[131] Ariana Scarlet Munoz Espinoza. *Swapping de correlaciones cuánticas.* PhD thesis, Universidad de Concepción, 2014.

[132] Julio I de Vicente Majúa. *Medidas de información, incertidumbre y entrelazamiento en Mecánica cuántica.* PhD thesis, Universidad Carlos III de Madrid, 2008.

[133] Yong Jiao, Eyuri Wakakuwa, and Tomohiro Ogawa. Asymptotic convertibility of entanglement: a general approach to entanglement concentration and dilution. *arXiv Preprint arXiv:1701.09050*, 2017.

[134] Charles H Bennett, Herbert J Bernstein, Sandu Popescu, and Benjamin Schumacher. Concentrating partial entanglement by local operations. *Physical Review A*, 53(4):2046, 1996.

[135] Matías Elías Soto Moscoso et al. *Correlaciones cuánticas en teleportación y teoría cuántica de juegos.* PhD thesis, Universidad de Concepción. Facultad de Ciencias Físicas y Matemáticas , 2017.

[136] William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802, 1982.

[137] Serge Huard. Polarization of light. *Polarization of Light, by Serge Huard, pp. 348. ISBN 0-471-96536-7. Wiley-VCH, January 1997.*, page 348, 1997.

[138] Jenny Marcela Rodríguez. Polarización de la luz: conceptos básicos y aplicaciones en astrofísica. *Revista Brasileira de Ensino de Física*, 40(4):e4310, 2018.

[139] William K Wootters. Entanglement of formation and concurrence. *Quantum Information & Computation*, 1(1):27–44, 2001.

[140] Adam Miranowicz and Andrzej Grudka. Ordering two-qubit states with concurrence and negativity. *Physical Review A*, 70(3):032326, 2004.

[141] Alexander Streltsov, Uttam Singh, Himadri Shekhar Dhar, Manabendra Nath Bera, and Gerardo Adesso. Measuring quantum coherence with entanglement. *Physical Review Letters*, 115(2):020403, 2015.

[142] Jin Wang, Herman Batelaan, Jeremy Podany, and Anthony F Starace. Entanglement evolution in the presence of decoherence. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 39(21):4343, 2006.

[143] S Bose, I Fuentes-Guridi, PL Knight, and V Vedral. Subsystem purity as an enforcer of entanglement. *Physical Review Letters*, 87(5):050401, 2001.

[144] Virginia Feldman. Dinámica de correlaciones cuánticas para estados x de dos qubits. 2016.

[145] Wojciech H Zurek. *Complexity, entropy and the physics of information*. CRC Press, 2018.

[146] Harold Ollivier and Wojciech H Zurek. Quantum discord: a measure of the quantumness of correlations. *Physical Review Letters*, 88(1):017901, 2001.

[147] Alexander Streltsov. *Quantum correlations beyond entanglement*. Springer, 2015.

[148] V Vedral. Classical correlations and entanglement in quantum measurements. *Physical Review Letters*, 90(5):050401, 2003.

[149] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge University Press, 10th anniversary edition, 2010.

[150] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum algorithm for the hamiltonian NAND tree. *arXiv preprint quant-ph/0702144*, 2007.

[151] Andrew M Childs and Wim Van Dam. Quantum algorithms for algebraic problems. *Reviews of Modern Physics*, 82(1):1, 2010.

[152] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6):467–488, 1982.

[153] David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

[154] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[155] Raymond N Greenwell. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *The College Mathematics Journal*, 31(1):70, 2000.

[156] Hernando Efraín Caicedo-Ortiz. Algoritmo de factorización para un computador cuántico. *Latin-American Journal of Physics Education*, 4(2):13, 2010.

[157] Hamed Ahmadi and Chen-Fu Chiang. Quantum phase estimation with arbitrary constant-precision phase shift operators. *arXiv Preprint arXiv:1012.4727*, 2010.

[158] Christian Javier Herrera Manosalvas. Simulador de un computador cuántico utilizando el algoritmo de shor para factorizar números enteros. B.S. thesis, PUCE, 2016.

[159] Je a Chiaverini, J Britton, D Leibfried, E Knill, Murray D Barrett, RB Blakestad, Wayne M Itano, John D Jost, C Langer, R Ozeri, et al. Implementation of the semiclassical quantum Fourier transform in a scalable system. *science*, 308(5724):997–1000, 2005.

[160] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge University Press, 10th anniversary edition, 2010.

[161] Gui-Lu Long. Grover algorithm with zero theoretical failure rate. *Physical Review A*, 64(2):022307, 2001.

[162] Lov K Grover. A fast quantum mechanical algorithm for database search. *arXiv Preprint quant-ph/9605043*, 1996.

[163] Thomas G Wong. Grover search with lackadaisical quantum walks. *Journal of Physics A: Mathematical and Theoretical*, 48(43):435304, 2015.

[164] Alisa Bokulich and Gregg Jaeger. *Philosophy of quantum information and entanglement*. Cambridge University Press, 2010.

[165] Norbert M Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A Landsman, Kenneth Wright, and Christopher Monroe. Experimental comparison of two quantum computing architectures. *Proceedings of the National Academy of Sciences*, 114(13):3305–3310, 2017.

[166] Masahito Hayashi. *Quantum information theory*. Springer, 2017.

[167] A García López and J García López. Criptografía cuántica. *Departamento Matemática Aplicada. Escuela Universitaria de Informática. Universidad Politécnica de Madrid. España*, 2005.

[168] Luis Caceres Alvarez and Patricio Collao Caiconte. Comparison and analysis of BB84 and E91 quantum cryptography protocols security strengths. *International Journal of Modern Communication Technologies and Research*, 4(9), 2016.

[169] AS Trushechkin, PA Tregubov, EO Kiktenko, Yu V Kurochkin, and AK Fedorov. Quantum-key-distribution protocol with pseudorandom bases. *Physical Review A*, 97(1):012311, 2018.

[170] Stamatios V Kartalopoulos. K08: a generalized BB84/B92 protocol in quantum cryptography. *Security and Communication Networks*, 2(6):686–693, 2009.

[171] Stamatios V Kartalopoulos. Chaotic quantum cryptography. In *2008 The Fourth International Conference on Information Assurance and Security*, pages 338–342. IEEE, 2008.

[172] Minal Lopes and Nisha Sarwade. On the performance of quantum cryptographic protocols SARG04 and KMB09. In *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, pages 1–6. IEEE, 2015.

[173] Lizal Iswady Ahmad Ghazali, Ahmad Fauzi Abas, Wan Azizun Wan Adnan, Makhfudzah Mokhtar, Mohd Adzir Mahdi, and M Iqbal Saripan. Security proof of improved-SARG04 protocol using the same four qubit states. In *International Conference On Photonics 2010*, pages 1–4. IEEE, 2010.

[174] Sellami Ali and Omer Mahmoud. Implementation of SARG04 decoy state quantum key distribution. In *2011 6th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, pages 86–90. IEEE, 2011.

[175] Mohamed Elboukhari, Mostafa Azizi, and Abdelmalek Azizi. Quantum key distribution protocols: a survey. *International Journal of Universal Computer Science*, 1(2), 2010.

[176] Ahmed I Khaleel. Coherent one-way protocol: design and simulation. In *2012 International Conference on Future Communication Networks*, pages 170–174. IEEE, 2012.

[177] Damien Stucki, Claudio Barreiro, Sylvain Fasel, Jean-Daniel Gautier, Olivier Gay, Nicolas Gisin, Rob Thew, Yann Thoma, Patrick Trinkler, Fabien Vannel, et al. Continuous high speed coherent one-way quantum key distribution. *Optics Express*, 17(16):13326–13334, 2009.

[178] Nicolas Gisin, Grégoire Ribordy, Hugo Zbinden, Damien Stucki, Nicolas Brunner, and Valerio Scarani. Towards practical and fast quantum cryptography. *arXiv Preprint Quant-ph/0411022*, 2004.

[179] Kyo Inoue, Edo Waks, and Yoshihisa Yamamoto. Differential phase shift quantum key distribution. *Physical Review Letters*, 89(3):037902, 2002.

[180] Kyo Inoue, Hiroki Takesue, and Toshimori Honjo. Dps quantum key distribution and related technologies. In *Quantum Communications Realized II*, volume 7236, page 72360I. International Society for Optics and Photonics, 2009.

[181] Shashank Kumar Ranu, Gautam Kumar Shaw, Anil Prabhakar, and Prabha Mandayam. Security with 3-pulse differential phase shift quantum key distribution. In *2017 IEEE Workshop on Recent Advances in Photonics (WRAP)*, pages 1–7. IEEE, 2017.

[182] Edo Waks, Hiroki Takesue, and Yoshihisa Yamamoto. Security of differential-phase-shift quantum key distribution against individual attacks. *Physical Review A*, 73(1):012344, 2006.

[183] Kyo Inoue. Differential phase-shift quantum key distribution systems. *IEEE Journal of Selected Topics in Quantum Electronics*, 21(3):109–115, 2014.

[184] Jian Li, Na Li, Lei-Lei Li, and Tao Wang. One step quantum key distribution based on EPR entanglement. *Scientific Reports*, 6:28767, 2016.

[185] Song-Kong Chong and Tzonelih Hwang. Quantum key agreement protocol based on BB84. *Optics Communications*, 283(6):1192–1195, 2010.

[186] Antonio Ruiz Alba Gaya, David Calvo Díaz-Aldagalán, Víctor García Muñoz, Alfonso Martínez García, Amaya Ocampo, Waldimar Alexander, ROZO CHICUE, JUAN GUILLERMO, José Mora Almerich, and José Capmany Francoy. Practical quantum key distribution based on the BB84 protocol. In *Waves*, volume 1, pages 4–14. Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM), 2011.

[187] Alexander V Sergienko. *Quantum communications and cryptography*. CRC press, 2018.

[188] Charles H Bennett and Gilles Brassard. Quantum cryptography: public key distribution and coin tossing. *Theor. Comput. Sci.*, 560(12):7–11, 2014.

[189] S Pirandola, UL Andersen, L Banchi, M Berta, D Bunandar, R Colbeck, D Englund, T Gehring, C Lupo, C Ottaviani, et al. Advances in quantum cryptography. *arXiv Preprint arXiv:1906.01645*, 2019.

[190] Jan CA Van Der Lubbe, Jan CA VanDerLubbe, and Jan CA Lubbe. *Basic methods of cryptography*. Cambridge University Press, 1998.

[191] Robbi Rahim. Man-in-the-middle-attack prevention using interlock protocol method. *ARPN J. Eng. Appl. Sci*, 12(22):6483–6487, 2017.

[192] Dang Nguyen Duc and Kwangjo Kim. Securing against GRS man-in-the-middle attack. In *Proc. Of SCIS 2007*, pages 23–26. Institute of Electronics, Information and Communication Engineers, 2007.

[193] Eli Biham, Michel Boyer, P Oscar Boykin, Tal Mor, and Vwani Roychowdhury. A proof of the security of quantum key distribution. *Journal of Cryptology*, 19(4):381–439, 2006.

[194] Yehuda Naveh, Elham Kashefi, James R Wootton, and Koen Bertels. Theoretical and practical aspects of verification of quantum computers. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 721–730. IEEE, 2018.