Computación Cuántica
Facultad de Ingeniería
Universidad de Antioquia
2025-1

Práctica No. 4
Criptografía: Distribución de Clave Cuántica

Realización: En parejas.

Fecha de Entrega: Miércoles **18 de junio** del 2025.

Introducción:

Recientemente la Distribución de Clave Cuántica (*Quantum Key Distribution* – **QKD**) ha recibido una significativa y merecida atención entre los expertos en seguridad informática, debido a que los métodos de encriptación tradicionales se están viendo amenazados por la aparición de los computadores cuánticos, por lo cual resulta necesario considerar métodos de encriptación más seguros.

Actualmente las redes **QKD** se han comenzado a implementar en diversas áreas metropolitanas del mundo, de modo tal que existen diferentes proyectos de construcción de redes **QKD** a escala intercontinental[1,2].

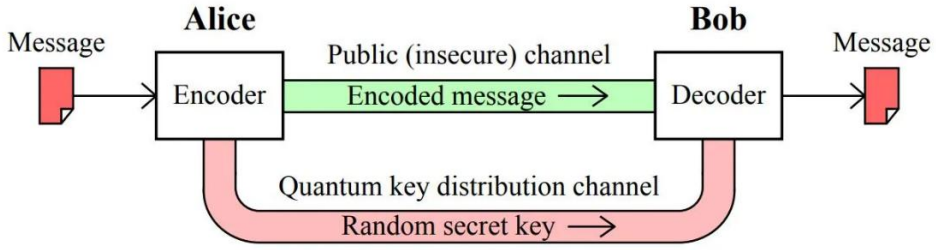

Figura 1: Diagrama de bloques de la criptografía cuántica

[1] **Quantum Messages Cross Germany Using Conventional Fiber Toshiba sent QKD keys more than 250 kilometers:** https://spectrum.ieee.org/quantum-key-distribution-commercial-fiber

[2] **Quantum Key Distribution Demonstrates Unconditional Security in Practical Implementation with SCS Protocol:** https://quantumzeitgeist.com/quantum-key-distribution-demonstrates-unconditional-security-in-practical-implementation-with-scs-protocol

La **QKD** proporciona un medio para compartir de manera segura una clave secreta entre dos partes distantes (canal inferior de la **Figura 1**), tal que dicha clave queda protegida contra intrusos que posean capacidad computacional ilimitada. La eficiencia de la seguridad de este método de comunicación se fundamental en la física experimental, y no en las matemáticas, para cifrar los datos que se desean comunicar.

1. Marco Teórico

1.1. Consulte y explique el protocolo de seguridad **BB84**[3].

1.2. Indique las condiciones en las que se fundamenta este protocolo **BB84**.

2. Análisis de un ejemplo del Tutorial del Qiskit

Analice el programa desarrollado en **Qiskit** del tutorial denominado ***"Quantum Key Distribution"***, disponible en el siguiente enlace:

https://github.com/**Qiskit**/textbook/blob/main/notebooks/ch-algorithms/quantum-key-distribution.ipynb

2.1. Cree una copia local de dicho tutorial disponible en **Github**.

- Antes de ejecutarlo, actualícelo a la versión 2.0 de **Qiskit**.

2.2. Defina en sus propias palabras, en qué consiste el protocolo *Quantum Key Distribution* (**QKD**) presentado en este tutorial.

- De acuerdo con el programa presentado en dicho tutorial, indique: ¿cuántos circuitos cuánticos se requieren en la implementación de este protocolo *Quantum Key Distribution*? **Explique.**
- Igualmente explique qué información ingresa y se retorna en cada función "**def**" implementada en este tutorial.

2.3. En la **Tabla 1** se resume la información registrada por los diferentes entes involucrados en el protocolo **QKD** implementado en el tutorial. Indique en qué consiste la información mostrada en cada una de las filas de esta tabla, así como las tres columnas presentadas.

---

[3] **BB84**: Charles **B**ennett and Gilles **B**rassard (1984).

- De acuerdo con la información registrada en dicha tabla, explique por qué la información enviada a través del **Canal de *Eve***, no es suficiente para que un *"espía"* descifre la clave compartida entre ***Alice*** y ***Bob***.
- Presente **evidencia matemática** de su explicación (cadenas de *bitstring*, estados y circuitos cuánticos, etc).

Alice's Knowledge	Over Eve's Channel	Bob's Knowledge
alice_bits		
alice_bases		
message	message	message
		bob_bases
		bob_results
	alice_bases	alice_bases
bob_bases	bob_bases	
alice_key		bob_key
bob_sample	bob_sample	bob_sample
alice_sample	alice_sample	alice_sample
shared_key		shared_key

Tabla 1: Información registrada por las diferentes partes involucradas en el protocolo **QKD**.

2.4. Realice su propio análisis del riesgo de intrusión por parte de un tercero (***"Eve"***), y demuestre por qué la probabilidad de la intrusión no ser detectada es:

$$\boldsymbol{P(\text{Intrusión no detectada}) = 0,75^{x}}$$

2.5. Modifique la función *"np.random.seed"* de la manera necesaria a fin de generar una clave secreta realmente aleatoria entre ***Alice*** y ***Bob***.

- Muestre la clave secreta obtenida con su código modificado.
- Luego guarde dicha **clave secreta**, y realice el siguiente numeral **3.**

3. Encriptado de un mensaje secreto

Con la **clave secreta** binaria creada por ***Alice*** y ***Bob*** en el tutorial anterior, realice dos programas en Python (**su_nombre.ipynb** y **su_compañero.ipynb**, referente a su nombre y el de su compañero de laboratorio, respectivamente), a fin de que uno de ustedes envíe un mensaje codificado usando la clave secreta binaria desarrollada con el **QKD**, y su compañero de laboratorio decodifique el mensaje recibido.

3.1. Su mensaje debe ser de al menos 15, y máximo 25 caracteres. Ejemplo: ***"Reunión secreta 02:15hrs".***

3.2. Elabore un programa en Python que implemente un método propio para **codificar** y **decodificar** dicho mensaje, a partir de la **clave secreta binaria** compartida por ***Alice*** y ***Bob***. (**Explique** su método utilizado).

3.3. Agregue comentarios y explicaciones propias en **español** a sus códigos en Python y **Qiskit**.

3.4. De acuerdo con el numeral **2.4** de la sección anterior, ¿cuál es la probabilidad de que un tercero (***"Eve"***), lea su mensaje secreto sin ser detectado? **Explique**.

- De acuerdo con su resultado obtenido, ¿se confirma entonces el nivel de seguridad esperado con este protocolo **QKD**, según su consulta realizada en el **ítem 1.1**? **Explique su respuesta**.

4. Informe

4.1. Presente su informe con el análisis pedido en esta guía de laboratorio.

4.2. Adjunte a este informe un archivo **zip** con los códigos escritos y "pantallazos" de cada ejecución realizada.

4.3. Presente conclusiones, y bibliografía adicional utilizada para realizar este informe.