Índice

Computación Cuántica Aplicada En Criptografía

Christian Soto Anaya

26 de diciembre de 2018

1. Introducción

La computación cuántica se empezó a desarrollar en la década de los ochenta por las propuestas de Deutsch y Feymann, que sugirieron que la evolución de los sistemas cuánticos se podría utilizar como herramienta de cálculo. Con la aparición de algoritmos que hacen uso de las leyes de la mecánica cuántica, permiten reducir la complejidad de algunos problemas en la computación clásica, convirtiéndolos de ser intratables a problemas que se puedan llegar a solucionar. [13]

De todos los algoritmos existentes en esta nueva teoría, sobresale el algoritmo de Shor, el cual permite descomponer en factores primos un número N, por lo cual su futura implementación en un computador cuántico traería como consecuencia que sistemas criptográficos basados en procesos de factorización como el sistema de clave pública RSA fueran fácilmente quebrantados. Mientras los procesos asociados a los algoritmos de clave pública se realizan en tiempos de la forma $O(e^{c(\ln N)^{1/3}(\ln)^{2/3}})$ [4], para el algoritmo cuántico de Shor, el tiempo necesario para realizar esta misma tarea es polinómico y de la forma $O((logN)^3)$ [1].

El objetivo de este proyecto es dar a entender cómo funciona el algoritmo RSA, que métodos existen actualmente para vulnerarlo y como la computación cuántica propone un nuevo esquema que permite vulnerarlo de manera más eficiente que los métodos actuales, además de un ejemplo construido en Java que simula el algoritmo de Shor para la factorización de números enteros.

El documento está conformado primero por el estado del arte que contiene los temas de la criptografía clásica, como se desarrolló, como se usa en la actualidad y que avances ha habido en la criptografía cuántica. Posteriormente en el marco teórico se explica cómo funciona la computación cuántica partiendo desde lo qubits y que características son importantes que hay que tener en cuenta para la comprensión del documento. A continuación se expondrán 2 temas, el primero sobre cómo se puede vulnerar el algoritmo RSA y el segundo sobre un

esquema de cómo se realizaría la autenticación cuántica por medio de la propiedad del entrelazamiento cuántico. Y por ultimo las conclusiones sobre cada tema expuesto y la bibliografía que fue utilizada a lo largo del proyecto.

2. Estado del arte

2.0.1. Criptografía clásica

Desde hace tiempo atrás la necesidad de transmitir un mensaje preservando la confidencialidad hizo que los medios para ocultar información se desarrollaran, surgiendo así, a lo largo de la historia, distintos métodos para asegurar la confidencialidad como la estenografía, escitalo espartano, el cifrado cesar, cifrado por sustitución y mucho más.

Con la aparición de las computadoras e internet y para asegurar que la información transmitida por los nuevos medios de transmisión sea confidenciales, aparecen nuevos mecanismos de encriptamiento como los sistemas de cifrado simétrico o de clave privada, que utilizan una sola clave para cifrar y descifrar la información, el principal problema de este mecanismo reside en el intercambio de la clave entre el emisor y el receptor ya que ambos deben usar la misma clave, algunos ejemplos de cifrados simétricos son DES, 3DES, IDEA y AES entre otros.

A partir de los mecanismos de cifrado simétrico y de su principal problema de intercambio de llaves, nacen los mecanismos de cifrado asimétrico o de clave pública como lo es RSA, el cual usa dos claves diferentes, una para cifrar y otra para descifrar la información, una es la clave pública a la que cualquier persona tiene acceso, y la otra es la clave privada que debe ser guardada en secreto para el intercambio de mensajes, tal que, cuando se valla a enviar un mensaje, el emisor usa la clave pública del destinatario para cifrar el mensaje, una vez se halla cifrado el mensaje, solo el destinatario de la clave pública con su clave privada puede descifrar el mensaje. Por esta razón se puede dar a conocer la clave pública, para que todo aquel que se quiera comunicar con el destinatario lo pueda hacer y se soluciona el problema de la distribución de la clave del sistema de cifrado simétrico. Una de las características fuertes de este sistema es que calcular la clave privada a partir de la clave pública aparenta ser difícil, esto solo significa que al día de hoy no se conoce ningún algoritmo que realice el trabajo de forma eficiente, sin embargo todavía existe la posibilidad de algún avance en la complejidad computacional como lo propone la computación cuántica que permita realizar estos cálculos y ponga fuera de servicio los sistemas de cifrado asimétrico. Finalmente este sistema tiende a ser considerablemente más lento que los sistemas de cifrado simétricos

Por lo anterior mencionado se decidió utilizar un enfoque de ambos sistemas y así aprovechar lo mejor de cada uno, en donde se utiliza la criptografía asimétrica para transmitir la clave secreta de la criptografía simétrica, y así se asegura un

intercambio de llaves seguro y el uso de un sistema de cifrado rápido, sin embargo así como apareció RSA como algoritmo de criptografía asimétrica, se propuso en 1977 un reto donde se le pagarían 100 dólares a quien lograra descifrar un mensaje numérico de 129 cifras decimales, dicho reto fue nombrado como RSA-129 y se estimó que faltarían millones de años para lograr dicho reto, de esta manera surgen los retos de factorización de RSA, el cual consiste en una lista números presentados en 1991 de más de 100 dígitos de los cuales la mayoría de estos siguen aun sin factorizar. [6]

2.0.2. Computación Cuántica

La criptografía cuántica es la criptografía que utiliza principios de la mecánica cuántica para garantizar la absoluta confidencialidad de la información transmitida [12]. Las actuales técnicas de la criptografía cuántica permiten a dos personas crear, de forma segura, por medio de una propiedad única de la física cuántica una manera para cifrar y descifrar mensajes, la cual permite que, si un tercero intenta hacer eavesdropping (espiar) durante la transmisión de la clave secreta, el proceso se altera advirtiéndole al emisor y al receptor antes de que se transmita información privada. Esto es una consecuencia del principio de incertidumbre de Heisenberg, que nos dice que el proceso de medir en un sistema cuántico perturba dicho sistema.

El desarrollo de la computación cuántica empieza en los años 90, con la aparición de los primeros algoritmos cuánticos, como el algoritmos de Peter Shor para la factorización de enteros y el algoritmo de Lov Grover para la búsqueda de datos, posteriormente se iniciaron los primeros experimentos prácticos con la primera comunicación segura usando criptografía cuántica a una distancia de 23 Km

En 1998 nace la primera máquina de 2 qubits presentada en la Universidad de Berkeley y un año más tarde, IBM diseña la máquina de 3 qubits, capaz de ejecutar el algoritmo de búsqueda de Grover. En el año 2000 nuevamente IBM diseña el primer computador cuántico de 5 qubits. Al mismo tiempo, científicos en los Álamos anunciaron el desarrollo de un computador cuántico de 7 qubits en el cual se logra ejecutar por primera vez el algoritmo de Shor, en el experimento se calcularon los factores primeros de 15 obteniendo como resultado 3 y 5. Finalmente a mediados de mayo del 2017 IBM presenta un nuevo procesador cuántico de 17 qubits y posteriormente presenta un ordenador cuántico de 50 qubits [3]

2.0.3. Criptografía Cuántica

Desde el año 2002 la criptografía cuántica ya ofrece productos comerciales, las compañías Id Quantique, de Ginebra, y MagiQ Tecnologies, de Nueva York, han ofrecido al público productos que envían una clave de criptografía cuántica [11]. La compañía MagiQ Technologies, creada en 2002 ofrece MagiQ QPN,

un sistema de distribución de clave cuántica que promete la generación en tiempo real de las claves criptográficas, una distribución de clave segura entre partes y cifrado y descifrado de datos. La compañía Id Quantique ofrece productos como cerberis que combina el cifrado de alta velocidad de AES con la seguridad de una distribución de clave cuántica. [10]

Según la Facultad de Informática de la UPM Quince grupos de investigación europeos han puesto en marcha el proyecto QUITEMAD (QUantum Information Technologies MAdrid), que tiene cinco objetivos científicos concretos: criptografía cuántica, computación cuántica, control cuántico y tomografía, correlaciones cuánticas y simulación cuántica, Estas cinco líneas de investigación tienen aplicaciones científicas y tecnológicas relevantes, que van desde la implementación de criptografía cuántica para el sector industrial, hasta el desarrollo y funcionamiento de nuevas técnicas de computación e información cuánticas, incluyendo su realización experimental en colaboración con laboratorios nacionales e internacionales. [7]

3. Marco teórico

Los computadores cuánticos son capaces de realizar grandes tareas de computo reemplazando los bits tradicionales (0 y 1) por bits cuánticos llamados qubits. Los qubits existen en estado de superposición, esto quiere decir puede ser 1 y 0 al mismo tiempo, en lugar de estar restringido a un único estado binario. Esto le permite a los computadores cuánticos ser exponencialmente más poderosos que los computadores clásicos y resolver desafíos que están más allá del alcance de los computadores de hoy en día. También existen algunas características de la mecánica cuántica que ayudan a garantizar la seguridad en la transmisión de información cuántica como los son:

- El teorema de no clonación asegura que un estado cuántico $|\Psi)$ no puede ser copiado
- Cualquier intento por medir y obtener información de un sistema cuántico conllevara a perturbar dicho sistema ocasionando que se modifique.
- No se puede llevar un sistema a su estado original después de una medición, el sistema colapsara a uno de sus estados propios tras realizar una medición sobre este, lo cual es irreversible

Los estados como $|\Psi)$ son descritos como una combinación lineal de estados $|X_0)$, $|X_1)$, ..., $|X_{n-1})$, multiplicados por amplitudes complejas c_0, c_1, ..., c_{n-1} de la siguiente forma:

$$|\Psi) = c_0|X_0) + c_1|X_1) + ... + c_{n-1}|X_{n-1}) \tag{1}$$

Donde los coeficientes complejos representan la probabilidad de que el sistema llegue a ser encontrado en un estado respectivo.

Otra de las propiedades importantes que tiene la mecánica cuántica es la del entrelazamiento cuántico, la cual explica como un grupo de partículas entrelazadas están unidas en su existencia de manera que aunque estén separadas a miles de años luz, el cambio de estado de una de ellas afecta al resto de forma inmediata. [5]
El entrelazamiento cuántico para 2 partículas con sus respectivos estados cuánticos $|\Phi)$ y $|\Psi)$ se define de la siguiente forma

$$|\Psi) = c_{x0}|X_0) + c_{x1}|X_1)$$

$$|\Phi) = c_{y0}|Y_0) + c_{y1}|Y_1)$$

El resultado de entrelazar los dos estados anteriores es

$$|\varphi) = c_{x0}c_{y0}|X_0)\otimes|Y_0)+c_{x0}c_{y1}|X_0)\otimes|Y_1)+c_{x1}c_{y0}|X_1)\otimes|Y_0)+c_{x1}c_{y1}|X_1)\otimes|Y_1) \tag{2}$$

Para tener una noción del funcionamiento de esta propiedad se asignaran valores a las amplitudes complejas para beneficio del ejemplo.

$$|\varphi) = 1|X_0)\otimes|Y_0) + 0|X_0)\otimes|Y_1) + 0|X_1)\otimes|Y_0) + 1|X_1)\otimes|Y_1) \tag{3}$$

Con la ecuación 3 se quiere decir que los estados $|X_0)\otimes|Y_1)$ y $|X_1)\otimes|Y_0)$ no pueden suceder, ya que tienen una amplitud compleja de 0, por lo tanto, si dejamos únicamente los estados cuyas amplitudes son de 1, el estado entrelazado se puede reescribir de la siguiente forma

$$|\varphi) = |X_0)\otimes|Y_0) + |X_1)\otimes|Y_1) \tag{4}$$

¿Que se puede decir de esta ultima expresión? Si medimos la primera partícula, esta tendrá la probabilidad del 50 % de estar en la posición X_0 o en X_1. Si la encontráramos en la posiciónX_0, y como el estado $|X_0)\otimes|Y_1)$ tiene 0 como amplitud, sabemos que no hay probabilidades de encontrar la segunda partícula en la posición Y_1, por lo tanto concluimos que la segunda partícula también se encontrara en la posición Y_0. Igualmente, si la primera partícula es encontrada en la posición X_1, la segunda partícula seria encontrada en la posición Y_1, y ocurriría exactamente lo mismo con la primera partícula si medimos primero la segunda, los estados individuales de cada partícula están inmediatamente relacionados con la otra partícula, dicho de otra forma, sus estados están entrelazados.
John S. Bell propuso estados cuánticos específicos que representan los ejemplos mas simples de entrelazamiento cuántico, también conocidos como estados de Bell.

$$|\Psi^+\rangle = \frac{|0_A1_B\rangle + |1_A0_B\rangle}{\sqrt{2}} \, |\Phi^+\rangle = \frac{|0_A0_B\rangle + |1_A1_B\rangle}{\sqrt{2}}$$

$$|\Psi^-\rangle = \frac{|0_A1_B\rangle - |1_A0_B\rangle}{\sqrt{2}} \, |\Phi^-\rangle = \frac{|0_A0_B\rangle - |1_A1_B\rangle}{\sqrt{2}}$$

Estos estados son importantes en el área de la mecanica cuantica ya que por medio de una medicion de estados de Bell ayuda a determinar que tan correlacionados estan los estados entrelazados [2], una medicion de estados de Bell sobre 2 qubits determina en cual de los 4 estados se encuentran, si antes de la medición los qubits no se encuentran entrelazados, estos quedarán proyectados en un estado de Bell. [9, 8]

4. Cifrado RSA y Autenticación cuántica

4.1. Rompiendo RSA

4.1.1. Funcionamiento de RSA

RSA (Rivest, Shamir y Adleman) es un sistema criptográfico asimétrico o de clave pública desarrollado en 1977 por Ron Rivest, Adi Shamir y Leonard Adleman, del Instituto Tecnológico de Massachusetts (MIT). Es el primer y más utilizado algoritmo de este tipo y su seguridad radica en el problema de la factorización de números enteros. Los mensajes enviados se representan mediante números, y el funcionamiento se basa en el producto de dos números primos grandes elegidos al azar y mantenidos en secreto. Actualmente estos primos son del orden de 10^{200}

El algoritmo funciona de la siguiente forma:

- Se toman dos números primos al azar lo suficientemente grandes, es decir mayores a 10^{200} y los llamaremos P y Q. Para ejemplificarlo tomaremos 2 números primos pequeños.
  P= 13 y Q= 11

- A partir de los 2 números primos escogidos, Definiremos N como el módulo de las claves y Z como la función φ de Euler de N, φ(N) es decir, es la cantidad de números menores o iguales a N y coprimos con N

$$N = P * Q \tag{5}$$

$$Z = (P-1) * (Q-1) \tag{6}$$

  Para el ejemplo:
  N = 13*11 = 143
  Z = (13-1)*(11-1) = 12*10= 120

- Se elige un número E menor que Z y primo relativo(coprimo) con Z lo que significa que E y Z tienen factor común solo a 1.
  E = 7, 1<E<Z

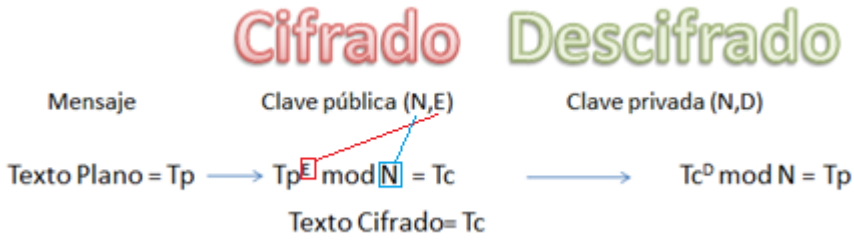
- Se debe encontrar un numero D como el inverso de E módulo Z, tal que

$$(E * D) Mod(Z) = 1 \tag{7}$$

  Para el ejemplo D = 103 es el numero que cumple con la condición
  (7*103) módulo(120) = 1

- (N,E) es la clave pública y (N,D) es la clave privada
  Clave pública = (143,7)
  Clave privada= (143,103).

Una vez teniendo las claves públicas y privadas, los procesos para cifrado y descifrado son:



Continuando con el ejemplo, supongamos que queremos cifrar la letra "a", para esto es necesario transformar la letra en números para poder aplicar los métodos, para el ejemplo usaremos ASCII a = 97

Para romper el algoritmo RSA analizaremos la manera en la que funciona, dado un texto cifrado Tc, sabemos que Tc fue cifrado por medio de la clave pública (N,E), es decir, un texto plano Tp fue elevado a la E módulo N

$$Tc = Tp^E Mod(N)$$

y por lo tanto la manera de descifrarlo es por medio de su llave privada (N,D) de la cual el único componente que desconocemos es el exponente privado D.
Para hallar el valor de D es necesario utilizar la parte 7 del algoritmo, pero esto conlleva a que necesitamos conocer φ(N) es decir Z, y la única forma de hacerlos es mediante la fórmula 6 lo que implica que necesitamos conocer P y Q, dicho de otra manera debemos factoriazar N

La seguridad del algoritmo RSA se basa en la dificultad computacional que conlleva encontrar los dos factores primos de un número compuesto muy grande, resultado del producto de éstos, donde sus primos también son números grandes. Esto es lo que matemáticamente se conoce como el problema de la factorización entera.
Se trata de un problema que en un sentido el cálculo es muy fácil y rápido (por ejemplo multiplicar dos números primos) pero que en sentido contrario (por

ejemplo, encontrar esos dos factores conocido el producto) se vuelve computacionalmente intratable a medida que la entrada es cada vez mayor. Es decir, requiere de unos recursos informáticos excesivos y, por tanto, de un tiempo de cálculo exorbitante.

El algoritmo RSA está diseñado para escoger primos P y Q suficientemente grandes del orden de 10^{200} para que sea inviable hallarlos mediante ordenadores convencionales. Los métodos más eficientes de factorización de números generales que se conocen son la criba general del cuerpo de números (por sus siglas en inglés General Number Field Sieve (GNFS)) y las curvas elípticas. El número más grande factorizado hasta la fecha es RSA-768 el cual es un número de 232 cifras decimales(768 bits) hallado en diciembre 12 del 2009 en el transcurso de 2 años mediante la GNFS el cual tiene como complejidad, dado un número de tamaño m, es $O(e^{(c+O(1))\times m^{1/3}\times(\log m)^{2/3}})$, siendo c una constante que depende de la variante del algoritmo. Este número está muy por debajo del rango manejado por el algoritmo RSA.

4.1.2. Algoritmo Cuántico

Las aplicaciones de la computación cuántica representan una amenaza potencial a las criptografías de clave pública como lo es en este caso RSA, más precisamente el algoritmo de Shor, propuesto por Peter Shor, es un algoritmo cuántico para descomponer en factores un número N en tiempo $O((logN)^3)$. Este algoritmo cimienta su eficiencia en determinar el periodo de una función. Aunque su estudio presenta un grado de complejidad relativamente alto, es muy interesante analizar el nuevo enfoque que la mecánica cuántica ofrece para solucionar el problema de factorización. La descripción paso a paso del algoritmo de Shor es el siguiente

Se debe escoger un número "a" menor que N y asegurarse de que no tenga factores triviales en común con N, esto se puede comprobar realizando el algoritmo de Euclides para calcular el MCD(a,N), si el MCD es diferente de 1 significa que se ha encontrado un factor de N y hemos terminado, de lo contrario "a" es coprimo de N y podemos usarlo, después es necesario hallar las potencias de "a" modulo N que son:

$$a^0 Mod(N),\ \ a^1 Mod(N),\ \ a^2 Mod(N), \ldots$$

en otras palabras necesitamos encontrar los valores de la función

$$F_{a,N}(x) = a^x Mod(N)$$

Para ejemplificar tomaremos N = 15 y a = 2, después de algunos cálculos obtenemos la siguiente información

x	0	1	2	3	4	5	6	7	8	9	10	11	12	···
$f_{2,15}(x)$	1	2	4	8	1	2	4	8	1	2	4	8	1	···

En realidad no se necesitan los valores de la función, si no que necesitamos encontrar el periodo más pequeño de la función tal que:

$$F_{a,N}(r) = a^r Mod(N) = 1$$

Esto es un teorema de la teoría de números que para cualquier coprimo $a <= N$ la función $F_{a,N}$ va a tener como resultado 1 para algún $r < N$. Después de que el resultado sea 1, la secuencia de números sencillamente se repetirá,. Si $F_{a,N}(r) = 1$, entonces:

$$F_{a,N}(r+1) = F_{a,N}(1)$$

En general

$$F_{a,N}(r+s) = F_{a,N}(s)$$

Para el ejemplo tomado, el periodo de la función $F_{2,15}$ es 4, ahora solo falta hallar los factores a partir del periodo, para esto se necesitara un periodo que sea numero par, de lo contrario desechamos dicha "a" y repetimos el proceso con otra hasta hallar un periodo par tal que:

$$a^r \equiv 1 \ \ Mod(N)$$

ahora, al restar 1 a ambos lados para mantener la equivalencia

$$a^r - 1 \equiv 0 \ \ Mod(N)$$

lo que significa que $a^r - 1$ es divisor de N o equivalentemente

$$N|(a^r - 1)$$

Recordando que $1 = 1^2$ y $x^2 - y^2 = (x+y)(x-y)$, tenemos:

$$N|(\sqrt{a^r} - 1)(\sqrt{a^r} + 1)$$

o

$$N|(a^{r/2} - 1)(a^{r/2} + 1) \tag{8}$$

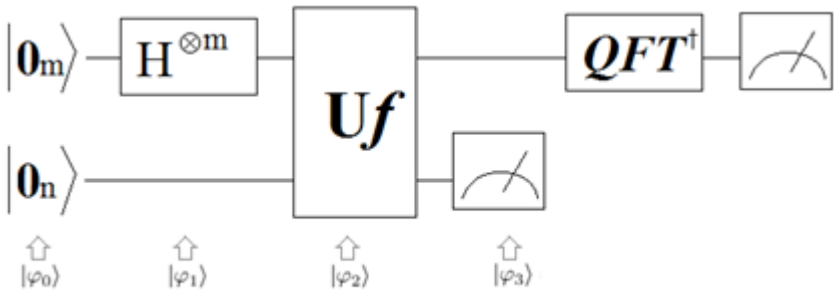
Esto significa que cualquier factor de N también es factor de $(a^{r/2} - 1)$ o de $(a^{r/2} + 1)$ o ambos, de igual forma, los factores de N pueden ser encontrados mirando por medio del algoritmo de Euclides el MCD($(a^{r/2} - 1)$, N) y MCD($(a^{r/2} + 1)$,N)
Debemos asegurarnos que $a^{r/2} \neq -1 Mod(N)$ porque si es así, el lado derecho de la ecuación 8 seria 0, en ese caso no obtendríamos ninguna información sobre

N y deberá ser desechado ese valor "a" en particular y empezar de nuevo.

Para el ejemplo, como el periodo de la función $F_{2,15}$ es 4, tenemos que $2^4 \equiv 1\, Mod(15)$, de la ecuación 8 obtendremos:

$$15|(2^2-1)(2^2+1)$$

Y por lo tanto tenemos que el MCD(3,15) = 3 y el MCD(5,15) = 5, los cuales son los factores respectivos de N.
Se acaba de presentar una explicación del algoritmo de shor de una manera clásica y trivial, puesto que el numero utilizado es un número pequeño y calcular su periodo es fácil, pero ¿Cómo sería con un N grande que tenga cientos de dígitos?, esto quedaría fuera de alcance de cualquier computador convencional, para esto se necesitaría un computador cuántico que con la capacidad de estar en superposición se podrá calcular $F_{a,N}(x)$ para todos los valores de x necesarios al mismo tiempo, para ilustrarlo usaremos el siguiente diagrama cuántico



Algoritmo de Shor

La respuesta del circuito cuántico que implementa la función $F_{a,N}$ siempre será menor que N y por lo tanto se necesitara $n = \log_2 N$ bits de salida, necesitaremos evaluar los primeros N^2 valores de x y por lo tanto se necesitaran como mínimo $m = Log_2 N^2 = 2Log_2 N = 2n$ bits de entrada
Al tiempo φ_0 tendremos:

$$|\varphi_0\rangle = |\mathbf{0}m, \mathbf{0}n\rangle$$

Donde **0**m y **0**n son cadenas de qubits de longitud m y n respectivamente, luego se pone la entrada en superposición de todas las posibles entradas

$$|\varphi_1\rangle = \frac{\sum_{x\in\{0,1\}^m} |\mathbf{x}, \mathbf{0}n\rangle}{\sqrt{2^m}}$$

Al evaluar la función $F_{a,N}$ en todas las posibilidades obtenemos

$$|\varphi_2\rangle = \frac{\sum_{x\in\{0,1\}^m} |\mathbf{x}, F_{a,N}(\mathbf{x})\rangle}{\sqrt{2^m}} = \frac{\sum_{x\in\{0,1\}^m} |\mathbf{x}, a^x Mod(N)\rangle}{\sqrt{2^m}}$$

Es justo aquí donde la grandiosa habilidad de la computación cuántica es usada, se han evaluado todos los valores necesarios de x al mismo tiempo, lo cual solo es posible por el paralelismo cuántico.
Tomando un ejemplo con N=15 tendríamos n=4 y m=8. Para a=13 el estado φ_2 seria

$$\frac{|0,1\rangle + |1,13\rangle + |2,4\rangle + |3,7\rangle + |4,1\rangle + ... + |254,4\rangle + |255,7\rangle}{\sqrt{256}}$$

Continuando con el algoritmo, se medirá el quibit de abajo en φ_2 el cual esta en súper posición de muchos estados, digamos que después de medirlo encontramos

$$a^{\overline{x}} Mod(N)$$

para algún $\overline{X}$, sin embargo por la periodicidad de $F_{a,N}$ y para cualquier s ∈ Z tenemos:

$$a^{\overline{x}} \equiv a^{\overline{x}+sr} Mod(N)$$

$\frac{2^m}{r}$ superposiciones de X tendrán $\overline{X} Mod(N)$ como respuesta, por lo tanto al tiempo de φ_3 tendremos

$$|\varphi_3\rangle = \frac{\sum_{a^x \equiv a^x Mod(N)} |\mathbf{x}, a^{\overline{x}} Mod(N)\rangle}{\frac{2^m}{r}}$$

que se puede escribir como

$$|\varphi_3\rangle = \frac{\sum_{j=0}^{2^m/r-1} |t_0 + jr, a^{\overline{x}} Mod(N)\rangle}{\frac{2^m}{r}}$$

Donde t_0 es la primera vez que $a^{t_0} \equiv a^{\overline{x}} Mod(N)$, es decir, es la primera vez que se repite un valor en cada periodo, continuando con el ejemplo, digamos que el valor de medir el qubit de abajo es 7, en ese caso el estado φ_3 seria

$$\frac{|3,7\rangle + |7,7\rangle + |11,7\rangle + |15,7\rangle + ... + |251,7\rangle + |255,7\rangle}{\frac{256}{4}}$$

El paso final de la parte cuántica del algoritmo es tomar dicha superposición y retornar el periodo, pero al hacer la medición, esta nos devolverá uno de todos los valores posibles, destruyendo todos los demás, por lo tanto es necesario transformar la superposición a otro estado que devuelva la respuesta correcta con alta probabilidad, y para esto es utilizado la transformada de Fourier cuántica. Y una vez teniendo el periodo de la función, solo es hallar los factores de N como se explicó con anterioridad.

4.1.3. Experimento

Hablando correctamente, la implementación de un algoritmo cuántico solo puede ser llevada acabo en un computador cuántico, por esta razón cuando se

habla sobre implementación en este texto en realidad se refiere a una simulación del algoritmo implementado en un computador clásico.
Todos los algoritmos cuánticos tienen una representación en términos de matrices, esta representación es posible ya que los qubits y todas las operaciones que se pueden realizar sobre estos tienen una representación matricial, como por ejemplo, un qubit se representa por medio de un vector(arreglo) de 2 posiciones, donde cada posición tiene como elemento un numero complejo c que representa cada uno de los estados del qubit

$$\begin{matrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \end{bmatrix}$$

Representacion de un qubit

Siendo así, si la posición 0 del vector tiene un 1 como elemento, significa que el qubit se encuentra en estado 0, igualmente si la posición 1 del vector es la que tiene un 1 como elemento, significa que el qubit se encuentra en estado 1.

$$|0\rangle = \begin{matrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{matrix}\mathbf{0}\\\mathbf{1}\end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Qubit en estado 0 Qubit en estado 1

La representación de un qubit en superposición no significa que un vector tenga en sus 2 posiciones un 1, ya que la suma de las normas al cuadrado de cada elemento debe ser igual a 1, $|c_0|^2 + |c_1|^2 = 1$.
Para poner un qubit en estado de superposición este debe ser multiplicado por la matriz de Hadamard

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

Matriz de hadamard

Al realizar dicha multiplicacion se obtiene un vector cuyas posiciones tendran como elemento $\frac{1}{\sqrt{2}}$ lo cual cumple con la condicion de $|\frac{1}{\sqrt{2}}|^2 + |\frac{1}{\sqrt{2}}|^2 = 1$.

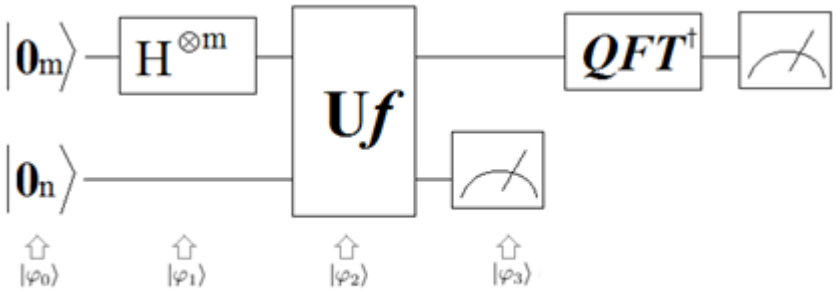
$$\begin{bmatrix}1\\0\end{bmatrix} * \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix} = \begin{bmatrix}1/\sqrt{2}\\1/\sqrt{2}\end{bmatrix}$$

Qubit | Hadamard | Qubit en superposición

Ahora que se definieron las representaciones de los elementos que se utilizan, se explicara como se llevo acabo la implementación del algoritmo de Shor sin embargo, también se implementó el algoritmo de Deutsch para lograr un mayor entendimiento de cómo se debía realizar una implementación adecuada, por esta razón el algoritmo de Deutsch será explicado de igual forma.

4.1.4. Algoritmo de Shor

El algoritmo de Shor es un algoritmo que recibe como parámetro un número N y retorna un factor p de N. Para la implementación se decidió utilizar como parámetro $N = 15$ y así construir en Java los pasos del algoritmo de Shor y calcular sus factores.
Retomando el circuito cuántico de Shor, se mostrara paso a paso su implementación



Algoritmo de Shor

Como se logra ver, para el tiempo φ_0 se necesitaran 2 cadenas de qubits $|\mathbf{0}m\rangle\, y\, |\mathbf{0}n\rangle$ donde $n = 4$ y $m = 8$ ya que $n = Log_2 15$ y $m = 2n$.
Para representar estas cadenas de quibts en java es necesario realizar el producto tensor entre la cantidad de qubits necesarios para alcanzar la longitud n y m

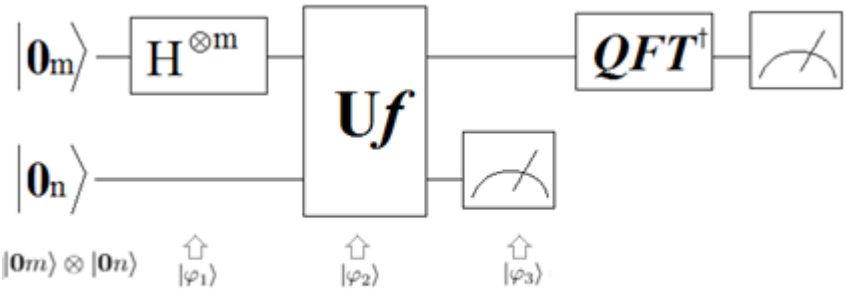
$$|0_m\rangle = |0_8\rangle = \begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}$$

$$|0_n\rangle = |0_4\rangle = \begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix}$$

Cadenas de Qubits

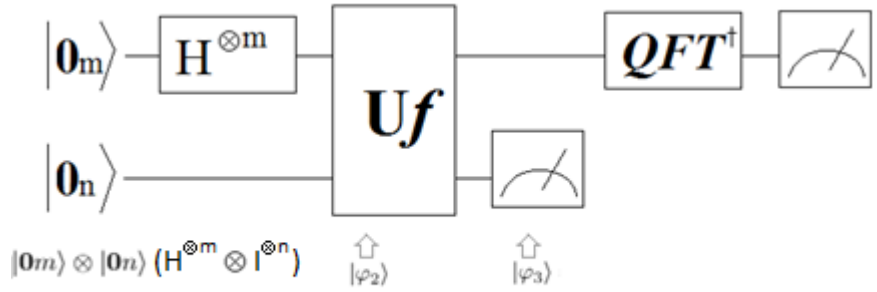
Una vez teniendo las cadenas de qubits $|\mathbf{0}m\rangle\, y\, |\mathbf{0}n\rangle$ definidas, ahora solo es realizar el producto tensor entre estas 2 cadenas $|\mathbf{0}m\rangle \otimes |\mathbf{0}n\rangle$ lo que generaría como resultado para φ_0 una cadena de 12 qubits, es decir un vector de 4096 posiciones.

Continuando con el circuito cuántico, para el tiempo φ_1 las cadenas de qubits se ponen en superposición por medio de la matriz de hadamard



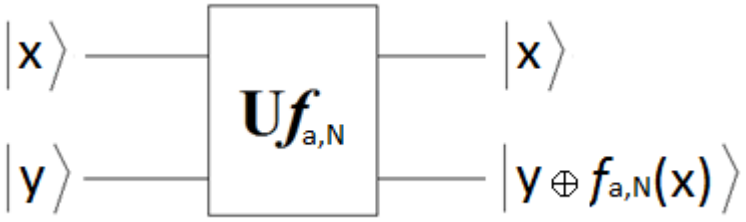
Algoritmo de Shor

Como la matriz de hadamard es una matriz de 2x2, ésta solo puede ser utilizada sobre un solo qubit ya que es un vector de 2x1, sin embargo como se necesita utilizar sobre una cadena de m qubits, también sera necesario hacer una 'cadena' de m matrices de hadamard realizando el producto tensor entre 8 matrices de hadamard, ahora, como la cadena de n qubits no se le aplica ninguna operación, esta es multiplicada por la matriz de identidad, por lo tanto, también se deberá realizar una 'cadena' de n matrices de identidad.



Algoritmo de Shor

Para el tiempo φ_2 se utiliza la función Uf, que para el caso del algoritmo de Shor Uf representa la función $F_{a,N}(x) = a^x Mod(N)$ y la matriz que corresponde a esta función, se construyo con ayuda del siguiente esquema: [13]

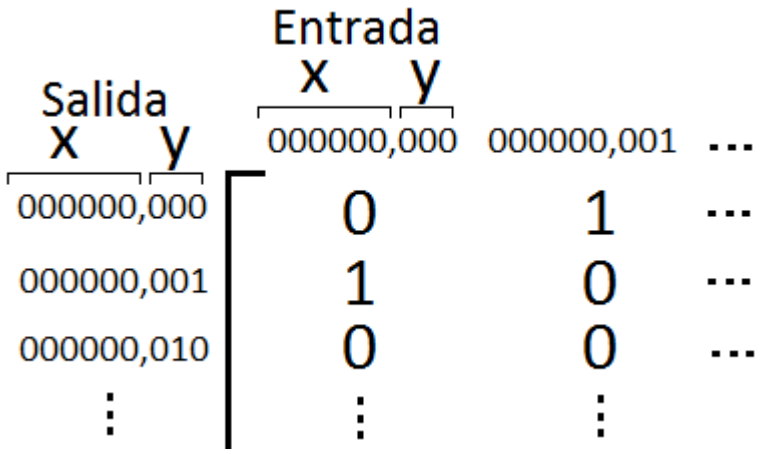


Este esquema nos dice que la función $Uf_{a,N}$ tiene como entrada 2 cadenas de qubits, una cadena de qubits $|\mathbf{X}\rangle = |X_0, X_1, ..., X_{m-1}\rangle$ y $|\mathbf{y}\rangle = |y_0, y_1, ..., y_{n-1}\rangle$ y que tiene como salida $|X\rangle$, el cual no sera cambiado por Uf y $|y \oplus f_{a.N}(\mathbf{X})\rangle$ siendo $\oplus$ el operado Xor, y aunque $\mathbf{X}$ sea una cadena, $f(\mathbf{X})$ evalúa un elemento a la vez.

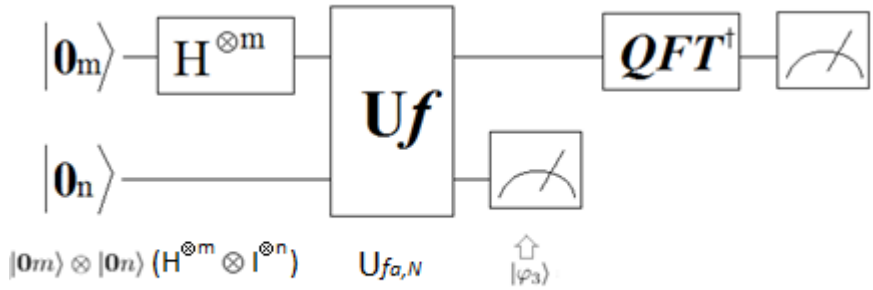
Para poder evaluar la combinación de los estados posibles de la cadena $|X\rangle$ con todos los estados posibles de la cadena $|y\rangle$ se hizo una matriz en la cual las columnas representan las cadenas $|x\rangle$,$|y\rangle$ y las filas las respuestas de $|x\rangle$,$|y \oplus f_{a.N}(\mathbf{X})\rangle$. Por ejemplo, cuando $|X\rangle = 000000$ y $|y\rangle = 001$ entonces se tiene como salida

$$|x\rangle, |y \oplus f_{a.N}(X)\rangle$$
$$|000000\rangle, |001 \oplus f_{a.N}(000000)\rangle$$
$$|000000\rangle, |001 \oplus a^{000000} Mod(N)\rangle$$
$$|000000\rangle, |001 \oplus a^0 Mod(N)\rangle$$
$$|000000\rangle, |001 \oplus 1\rangle$$
$$|000000\rangle, |001 \oplus 001\rangle$$
$$|000000\rangle, |000\rangle$$

la matriz quedaría de la siguiente forma



Continuando con el algoritmo



Algoritmo de Shor

Para el tiempo φ_3 después de aplicar la matriz de Ufa, N se debe aplicar la transformada de Fourier cuántica(QFT) para retonar el periodo a partir de la superposición de los estados. la representación de la QFT es la siguiente

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

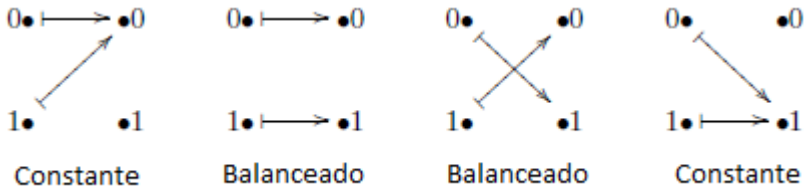
Donde

$$\omega = e^{\frac{2\pi i}{N}}$$

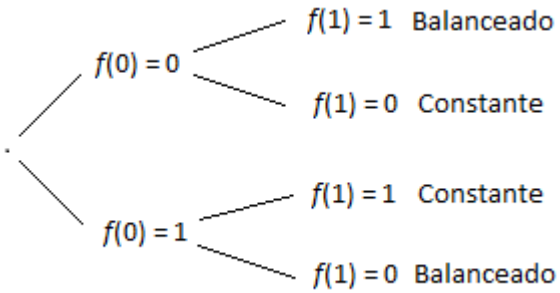
siendo i la parte imaginaria de la ecuacion y N la cantidad de qubits, como la QFT se le aplica a la cadena de qubits m, entonces, N=6.
Por ultimo solo es realizar la medicion de la cadenas de qubits, obtener el periodo r y calcular los factores de N por medio del MCD.

4.1.5. Algoritmo de Deutsch

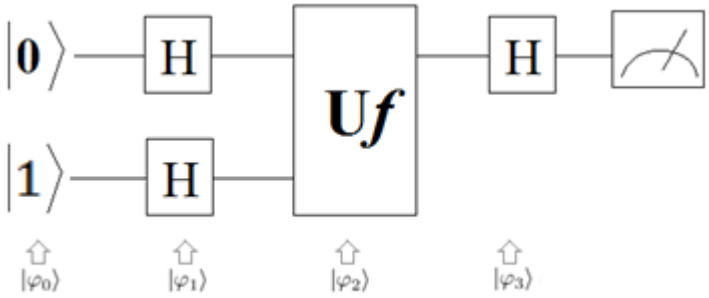
El algoritmo de Deutsch es el algoritmo cuántico más simple que resuelve el siguiente problema: dada una función $f : \{0,1\} \to \{0,1\}$ como una caja negra, en la que se puede evaluar la entrada, pero no es posible mirar dentro de la caja y ver como esta definida la función, el algoritmo determina si la función es balanceada o constante.
Una función $f : \{0,1\} \to \{0,1\}$ es Balanceada si $f(0) \neq f(1)$, es decir si es uno a uno, por otra parte, es una función constante si $f(0) = f(1)$. En total se pueden definir 4 funciones de las cuales 2 son balanceadas y 2 son constantes



En un computador clásico, primero tendríamos que evaluar f con la primera entrada y luego evaluar f en la segunda entrada, y finalmente comparar las salidas, el siguiente árbol de decisiones ilustra como un computador clásico realiza el proceso



La clave está en que con un computador clásico se debe evaluar la función f 2 veces, por otro lado, como un computador cuántico puede estar en superposición de estados, la idea es evaluar las 2 entradas al mismo tiempo. Para esto se seguirá paso a paso el circuito cuántico de Deutsch



Algoritmo de Deutsch

Para realizar la implementación de este circuito cuántico, primero, en el tiempo φ_0 necesitaremos 2 quibts, uno preparado en estado 1 y el otro en estado 0, como resultado en el tiempo φ_0 tendremos el producto tensor de estos 2 qubits.

$$\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$$

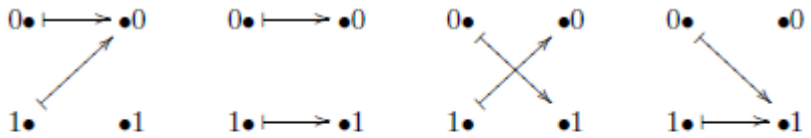
$$|0\rangle \otimes |1\rangle = |0,1\rangle$$

Al tiempo φ_1 los qubits son son puestos en superposición por medio de las matrices de hadamard, como hay 2 matrices de hadamard en el circuito, una junto al qubit de arriba y la otra abajo se tendrá como resultado el producto tensor de 2 matrices de hadamard

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\otimes\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

$$H \otimes H = (H \otimes H)$$

Al tiempo φ_2 se opera la función Uf la cual representa la función principal de un algoritmo cuántico. Para este caso tenemos 4 posibles funciones con sus respectivas matrices



$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \quad \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix} \quad \begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \quad \begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$$

Durante la explicacion, se utilizara la tercera matriz.
Para el tiempo φ_3, como solo se usa una matriz de hadamard arriba y no se tiene nada abajo, se hace el producto tensor de la matriz de hadamard con la matriz de identidad.

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1&1\\1&-1\end{bmatrix} \otimes \begin{bmatrix}1&0\\0&1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1&0&1&0\\0&1&0&1\\1&0&-1&0\\0&1&0&-1\end{bmatrix}$$

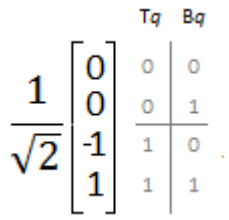
$$H \otimes I = (H \otimes I)$$

Como resultado de las operaciones en los tiempos φ_0, φ_1, φ_2 y φ_3 se tiene la multiplicación de los productos tensores $|0,1\rangle * (H \otimes H) * Uf * (H \otimes I)$, que se puede escribir en términos de matrices

$$\begin{bmatrix}0\\1\\0\\0\end{bmatrix} * \frac{1}{2}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix} * \begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} * \frac{1}{\sqrt{2}}\begin{bmatrix}1&0&1&0\\0&1&0&1\\1&0&-1&0\\0&1&0&-1\end{bmatrix}$$

$$|0,1\rangle \quad * \quad (H \otimes H) \quad * \quad Uf \quad * \quad (H \otimes I)$$

Finalmente, para obtener el resultado del circuito cuántico, toca medir el qubit de arriba el cual originalmente era un vector de 2 posiciones, sin embargo, debido a las operaciones que se realizaron con el qubit de abajo, como resultado se obtiene un vector de 4 estados, en el que las 2 primeras posiciones corresponden al estado 0 del qubit de arriba y las 2 últimas posiciones corresponden al estado 1.



En la imagen anterior, los valores de la columna T_q corresponden a los valores del qubit de arriba, y la columna B_q corresponden a los valores del qubit de abajo.
Según el algoritmo de Deutsch si al medir el qubit de arriba, este se encuentra en estado 0 significaría que la función Uf que se utilizo es constante, de lo contrario, si se encuentra en estado 1, significaría que Uf es balanceada.
Para el ejemplo utilizado, los valores del vector aparecen en las 2 últimas posiciones, lo que significa que el qubit de arriba se encuentra en estado 1 y por esta razón, la función Uf que se utilizo es balanceada.

4.2. Autenticación cuántica

La criptografía cuántica proporciona una distribución de claves incondicionalmente segura, se han propuesto varios protocolos de distribución como lo son el protocolo BB84 propuesto por Bennet y Brassard como el primer protocolo de distribución de clave cuántica y en 1991 Ekert propuso un plan para lograr comunicaciones seguras por medio del entrelazamiento cuántico-[13], estos protocolos son capaces de suministrar una clave secreta compartida aleatoria a dos usuarios, cuyo secreto está garantizado por las leyes fundamentales de la mecánica cuántica que provoca que cualquier intento de escucha secreta introducirá automáticamente un alto nivel de error en el bit cuántico.
Aunque los proponentes consideran que los protocolos de distribución de clave cuántica son incondicionalmente seguros, existe un defecto fundamental que es la falta de un mecanismos de autenticación, por lo tanto ni BB84 ni Ekert pueden garantizarle a Alice que la persona con quien comparte la llave es quien el cree que es. En este contexto se propusieron esquemas de autenticación cuántica
A continuación se mostrara un esquema para crear una clave de autenticación entre 2 partes, Alice y Bob, las dos partes compartirán un estado de dos qubits entrelazados, Cada uno poseerá una de las mitades del qubit entrelazado, luego, al azar, Bob realizara una operación con σ_z o iσ_y y le devolverá su partícula a Alice, que realizara una medición de estados de Bell. De esta forma, Alice se asegurará de que las partículas sean realmente de Bob.

4.2.1. Esquema de autenticación cuántica

La tarea general de la autenticación es verificar la identidad entre si de dos partes (Alice y Bob) en la comunicación, Los protocolos son tales que si Alice y Bob pueden completar con éxito alguno, Alice estará convencida de quien

está al otro lado del canal de comunicación cuántica. De manera clásica, esto se puede lograr haciendo que Bob le revele, atreves de un canal clásico, un secreto que hallan compartido con anterioridad. El esquema presentado usa estados entrelazados compartidos como la contra-parte de la clave secreta compartida y además puede proporcionar tokens de autenticación reutilizables
Primero, usando los estados de Bell como base, Alice y Bob habrán compartido K parejas de qubits entrelazados preparados en la forma

$$|\Phi^-\rangle = \frac{|0_A 0_B\rangle - |1_A 1_B\rangle}{\sqrt{2}}$$

- Primero, Bob aplica al azar una de las dos puertas cuánticas unitarias σ_z o iσ_y en su qubit, donde

$$\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

- Luego, Bob le regresa su partícula a Alice para que ella realice una medición de estados de Bell sobre la partícula de Bob y la suya.
- Si Bob utilizo el operador cuántico unitario σ_z en su partícula, entonces el estado medido por Alice seria Φ^+
- Sin embargo, si Bob utilizo iσ_y, entonces el estado inicial Φ^- será transformado en el estado Ψ^+. entonces si no hubiera nadie mas escuchando el canal, el resultado de Alice al medir será Φ^+ o Ψ^+

Consecuentemente Alice puede al mismo tiempo autenticar a Bob y compartir una clave secreta con él, puesto que al haber transmitido las K parejas de qubits y se halla confirmado la identidad de Bob estos pueden considerar el estado Φ^+ como un 0 binario y el estado Ψ^+ como un 1 Binario.

Shared EPR pair	Bob's unitary operation	Bell state measurement	Binary number
$\|\Phi^-\rangle = \frac{\|0_A 0_B\rangle - \|1_A 1_B\rangle}{\sqrt{2}}$	σ_z	$\|\Phi^+\rangle = \frac{\|0_A 0_B\rangle + \|1_A 1_B\rangle}{\sqrt{2}}$	0
	iσ_y	$\|\Psi^+\rangle = \frac{\|0_A 1_B\rangle + \|1_A 0_B\rangle}{\sqrt{2}}$	1

4.2.2. Análisis de seguridad

Para validar que este esquema de autenticación funciona es necesario que analicemos su seguridad, Cuando Bob le regresa la partícula a Alice, un espía podría interceptarla y enviar una partícula falsa, la cual se construye en el estado

$$\phi_E = a\,|0\rangle + b\,|1\rangle$$

lo que significa que el sistema tiene la probabilidad de $|a^2|$ de estar en el estado $|0)$ y la probabilidad de $|b^2|$ de estar en el estado $|1)$. Cuando Alice reciba la partícula falsa y realice la medición de estados de Bell, el estado de estas partículas seria

$$\frac{1}{2}\{|0_A 0_B\rangle + |1_A 1_B\rangle\} \otimes \{a^2 |0_A 0_B\rangle + ab^* |0_A 1_B\rangle + a^* b |1_A 0_B\rangle + b^2 |1_A 1_B\rangle\}$$

El resultado de la medición de los estados de Bell de las 2 partículas, debería ser uno de los 4 estados de Bell. Lo que significa que si Alice obtiene como resultado el estado Ψ^- o Φ^-, ella deberá creer que esas partículas no son de Bob y por ende finalizara la comunicación. En consecuencia, por cada estado compartido, la probabilidad de detectar a un espía es de $1/2$, y para k estados compartidos, la probabilidad de detectar al espía es de $(1 - (1/2)^k)$. Cuando k se vuelve lo suficientemente grande, la probabilidad de detectar al espía se vuelve aproximadamente del 100 % y por esto el espía no podrá hacerse pasar por Bob.

5. Conclusiones

Muchas criptografías de clave pública, tales como RSA, llegarían a ser obsoletas si el algoritmo de Shor es implementado alguna vez en una computadora cuántica práctica. Un mensaje cifrado con RSA puede ser descifrado descomponiendo en factores la llave pública N, que es el producto de dos números primos. Los algoritmos clásicos conocidos no pueden hacer esto en tiempo O((log N)k) para ningún k, así que llegan a ser rápidamente poco prácticos a medida que se aumenta N. Por el contrario, el algoritmo de Shor puede romper RSA en tiempo polinómico.

Se propuso un esquema de autenticación con distribución de clave cuántica, que ayuda a autenticar a un usuario externo al compartir qubits entrelazados entre las 2 partes, realizando algunas operaciones unitarias y mediciones de estados de Bell. Además el análisis de seguridad muestra que para un gran número de qubits entrelazados compartidos la probabilidad de detección de espionaje aumenta y el esquema se vuelve más seguro

La computación cuántica será el "boom" de las décadas venideras, siendo la mayor innovación y revolución informática de los últimos años, presentando un nuevo paradigma computacional donde se deberá estar en un permanente estudio y análisis de la ventajas como desventajas que podrá traer, incluyendo las diferentes medidas de control y mitigación ante las diversas amenazas que surgirán.

Referencias

[1] David Beckman y col. "Efficient networks for quantum factoring". En: *Phys. Rev. A* 54 (2 ago. de 1996), págs. 1034-1063. DOI: 10.1103/PhysRevA.54.1034.

[2] John S. Bell. *ON THE EINSTEIN PODOLSKY ROSEN PARADOX*. 1964.

[3] Vicente Moret Bonillo. *Principios Fundamentales de Computación Cuántica*. 2013.

[4] Hernando Efraín Caicedo-Ortiz. *Algoritmo de factorización para un computador cuántico*. 2010.

[5] Ludmil Hadjiivanov e Ivan Todorov. "Quantum entanglement". En: (jun. de 2015).

[6] Burt Kaliski. "RSA factoring challenge". En: *Encyclopedia of Cryptography and Security*. Ed. por Henk C. A. van Tilborg. Boston, MA: Springer US, 2005, págs. 531-532. ISBN: 978-0-387-23483-0. DOI: 10.1007/0-387-23483-7_362. URL: https://doi.org/10.1007/0-387-23483-7_362.

[7] Vanessa Marsh. "Fuerte impulso a la criptografía cuántica y los ordenadores cuánticos". En: (mayo de 2010).

[8] Segio. Martínez. *El problema de la medicion en mecánica cuántica*. 2014.

[9] Segio. Martínez. *Mediciones Ideales En La Mecánica Cuántica*. [Revista Hispanoamericana De Filosofía, vol. 20, no. 60]. 1988.

[10] Hernán Ortiz Rojas. *Fundamentos de Criptografía Cuántica*. 2007.

[11] Gary Stix. *Criptografía cuántica comercial*. [Investigacion y Ciencia. 2005, Pag 55-59]. 2013.

[12] Wikipedia. *Criptografía cuántica — Wikipedia, La enciclopedia libre*. [Internet; descargado 26-mayo-2018]. 2017.

[13] Noson S. Yanofsky y Mirco A. Mannucci. *Quantum Computing for Computer Scientists*. 1.ª ed. New York, NY, USA: Cambridge University Press, 2008. ISBN: 0521879965, 9780521879965.