Instituto Politécnico Nacional
Escuela Superior de Física y Matemáticas

Cómputo Cúantico y su Afectación a la Criptografía

T E S I S

que, para obtener el título de

Licenciado en Física y Matemáticas,

presenta

Alberto Guzmán Sánchez

ASESOR:
Dr. Luis Carlos Coronado García

México D. F., 19 de junio de 2009.

Índice general

Índice de figuras

Capítulo 1

Introducción

A medida que la tecnología avanza, se nota una tendencia a la miniaturización, desde las primeras computadoras (como ENIAC) hasta las computadoras portátiles actuales, y sigue decreciendo el tamaño de los componentes electrónicos de las computadoras; pero ¿qué sucede con los componentes cuando son demasiado pequeños?.

Las partículas subatómicas no se comportan como lo hacen los objetos cotidianos (como una pelota), pues están sujetas a las leyes de la mecánica cuántica, donde por ejemplo no se puede conocer con certeza en que posición se encuentra un electrón que orbita un átomo, y al tratar de conocer dicha posición se altera el sistema y sólo se conoce en que posición estaba el electrón con cierta probabilidad; pero el sistema ya ha cambiado. Es por esta razón que los componentes electrónicos de las computadoras cuando son demasiado pequeños se comportan de acuerdo a las leyes de la mecánica cuántica.

Debido a lo anterior, se necesita incorporar las leyes de la mecánica cuántica en los modelos computacionales resultando así un nuevo tipo de cómputo: *cómputo cuántico.*

Hace un poco más de una década, el cómputo cuántico era considerado sin utilidad práctica; pero esto cambió en 1994 cuando Peter Shor formuló un algoritmo para una computadora cuántica que podía factorizar enteros enormes (números con 100 cifras por ejemplo) en sus factores primos en un tiempo aceptable, un problema considerado intratable para el cómputo clásico, a raíz de esto, se inviertieron más fondos y recursos en la investigación del cómputo cuántico y algunos resultados importantes son los algoritmos de Grover (algoritmo de búsqueda) y Regev (algoritmo para encontrar el vector más corto en un retículo) que presentan una ventaja sobre el cómputo clásico.

Los objetivos de este trabajo consisten en presentar los elementos básicos de criptografía y mecánica cuántica que permitan el entendimiento del cómputo cuántico, describir al cómputo cuántico mostrando las diferencias y ventajas que existen entre éste y el cómputo "clásico", y, finalmente, estudiar algoritmos cuánticos que se han propuesto en estas dos últimas décadas y tienen una afectación importante en la criptografía, tanto positiva al surgir nuevos algoritmos cuánticos que son criptográficamente seguros, como negativa al dejar inservibles algunos algoritmos criptográficos actuales de uso amplio alrededor del mundo.

Esta tesis se encuentra estructurada del siguiente modo. En el capítulo 2 se describen conceptos criptográficos tales como esquemas de firma digital y esquemas de cifrado, así como funciones hash y sus clasificaciones, también se abarca lo relacionado con la definición de seguridad criptográfica de estos elementos y se ejemplifican estos conceptos. En el capítulo 3 se describen los conceptos de mecánica cuántica necesarios para el entendimiento del cómputo cuántico. En el capítulo 4 se describen los elementos que conforman al cómputo cuántico y su diferenciación respecto al cómputo "clásico". En el capítulo 5 se describen algoritmos cuánticos, dos de ellos de importancia por su superioridad ante cualquier algoritmo "clásico" para la resolución de un problema que ahí mismo se describe, pero que son inocuos a los esquemas criptográficos actuales, también se describen otros tres que sí afectan a la criptografía. En el capítulo 6 se describen dos elementos criptográficos que pueden implementarse solo con el auxilio de computadoras cuánticas y que aprovechan la parte cuántica para garantizar la seguridad de ellos. En el capítulo 7 se mencionan las principales conclusiones de este trabajo.

Capítulo 2

Conceptos preliminares criptográficos

En este capítulo se da una pequeña introducción a la criptografía, se habla sobre conceptos criptográficos también se ve la complejidad de algoritmos así como la seguridad criptográfica.

Entendemos por **Criptografía** al estudio de las técnicas matemáticas relacionadas con aspectos de seguridad de la información, tales como confidencialidad, integridad de datos, autenticación de identidad y autenticación de orígenes de datos.

Es decir que la criptografía no sólo se refiere a la seguridad de la información, sino también al conjunto de técnicas matemáticas utilizadas para asegurar la información.

La criptografía existe desde las primeras civilizaciones que empezaban a comunicar de manera escrita. La criptografía nace de la necesidad de comunicar información de tal forma que los únicos que conozcan lo que se comunica sean quien manda el mensaje y quien debe recibirlo, es decir, a quien va dirigido. Aquí sin darse cuenta conscientemente se está sobreentendiendo algunos conceptos criptográficos, por ejemplo, si uno recibe un mensaje, debe tener una forma de saber que el mensaje realmente viene de una fuente confiable, es decir, se tiene que verificar que el remitente haya mandado dicho mensaje. Esto es conocido como *autenticación de orígenes de datos*. Además se debe saber como descifrar el mensaje para eso se debe tener una *llave* o una *clave*, y además antes de recibir el mensaje debe *identificarse* con el mensajero para poder hacerse del mensaje, es decir, se debe *autenticar* la *identidad*.

2.1. Objetivo de la Criptografía

El objetivo fundamental de la criptografía es hacer que dos entidades puedan comunicarse en un canal inseguro de tal forma que un tercero no pueda entender lo que se está comunicando.

Este objetivo es alcanzado mediante técnicas o métodos matemáticos.Y se logra teniendo en cuenta los siguientes aspectos:
1.- Confidencialidad.
2.- Autenticación.
3.- Integridad de la información.
4.- No-autorización.

El primer aspecto (Confidencialidad), es usado para mantener guardado el contenido de la información de todos excepto de los que tienen autorización para ver el contenido de dicha información. Privacidad es un término sinónimo de confidencialidad. Hay muchas aproximaciones a la provisión de confidencialidad, desde protección física (lectores de huellas digitales, lectores de córneas) hasta algoritmos matemáticos que interpretan los datos como ininteligibles.

La integridad de datos localiza la alteración no autorizada de datos. Para asegurar la integridad de datos, uno debe tener la capacidad de detectar la manipulación de datos por partes no autorizadas. La manipulación de datos incluye la inserción, borrado y sustitución de datos.

La autenticación está relacionada con la identificación. Se aplica tanto a las entidades como a la información. Dos partes que entran en comunicación deben identificarse mutuamente. La información enviada sobre un canal (de información) debe ser autenticada de acuerdo al origen, fecha de origen, contenido de información, tiempo de envío, etc. Por estas razones este aspecto de la criptografía es comúnmente dividida en dos clases: *autenticación de identidad* y *autenticación de origen de datos*. El origen de datos provee implícitamente la integridad de datos (pues si el mensaje ha sido modificado, el origen habrá cambiado).

La no-autorización impide a una entidad denegándole algunas acciones. Cuando hay duda acerca de una entidad se le niega la ejecución de ciertas acciones y es necesaria alguna forma de resolver la situación.

Se conoce como primitivas criptográficas a las herramientas criptográficas que proveen seguridad de la información. Ejemplos de primitivas son esquemas de cifrado, funciones Hash y esquemas de firma digital.

2.2. Elementos criptográficos

En esta sección se dan algunos conceptos básicos en el estudio de la criptografía.

Noción 2.1 *Se le llama "texto llano" al texto que se quiere comunicar, dicho texto debe estar escrito en lenguaje cotidiano.*

Noción 2.2 *Se le llama "texto cifrado" al texto que se comunica sobre un canal inseguro que trata de no ser descifrado más que por la persona a quien va dirigido.*

El texto llano es la parte inteligible para el emisor y el receptor de la información. Esta información podría ser: una imagen codificada en cualquier formato, un programa ejecutable en alguna plataforma en específico, un mensaje como el "hola mundo", una instrucción de compra-venta de títulos financieros, etc. El texto cifrado es la parte ininteligible que va de emisor al receptor y cualquier tercero no pueda obtener de esta parte la inteligible, aún conociendo la información pública del emisor y receptor involucrada en esta creación.

Definición 2.1 *Un sistema criptográfico es una 5-tupla (P, C, K, E, D) donde las siguientes condiciones se satisfacen:*

1. P es un conjunto finito de posibles textos llanos.
2. C es un conjunto finito de posibles textos cifrados.
3. K, el espacio de claves, es un conjunto finito de posibles claves
4. Para cada $k \in K$, hay una regla de cifrado $e_k \in E$ y una correspondiente regla de descifrado $d_k \in D$. Cada $e_k{:}P \longrightarrow C$ y $d_k{:}C \longrightarrow P$ son funciones tales que $d_k(e_k(x))=x$ para todo texto llano $x \in P$.

En esta definición cada elemento $k \in K$ es una biyección de M a C. Puesto que es una biyección, el proceso es reversible y se obtiene un único texto llano para cada texto cifrado. Podemos pedir que $k \in K$ sea inyectiva y entonces la biyección se tiene de M a $Im(k)$, con $Im(k) \subseteq C$
Como se ve las funciones dadas (d y e) al componerlas se obtiene la función identidad, por lo que la función d es inyectiva y la función e es suprayectiva. Es decir, que dados dos textos cifrados que son iguales, tienen asociado el mismo texto llano, y que para cualquier texto cifrado c_1 existe un texto llano p_1 de tal forma que $e(p_1)=c_1$

Noción 2.3 *Una función f de un conjunto X a un conjunto Y se dice* ***en un solo sentido*** *si f(x) es "fácil de calcular" (computacionalmente hablando, que se tenga una implementación de f que permita calcular f(x) en tiempo polinomial respecto de la entrada x) para toda $x \in X$ pero para casi todos los elementos de $y \in Im(f)$ es "imposible calcular" x (computacionalmente hablando, que no exista o que al menos no se conozca un algoritmo que se ejecute en tiempo polinomial para calcular x con el puro conocimiento de f y f(x) escogida x aleatoriamente en X) tal que f(x) = y*

Ejemplo 2.1 *Como ejemplo de una función de un sentido tenemos una basada en el problema del logaritmo discreto. Considere $p \in \mathbb{Z}$ un número primo y $a \in \mathbb{Z}_p^*$ tal que $\langle a \rangle = \mathbb{Z}_p^*$ un generador. Sea $f : \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ definida por $f(x) = a^x$.*

Ya se había hablado de las primitivas, ahora se presenta el cifrado de clave simétrica.

Definición 2.2 *Considere un esquema de cifrado que consiste de conjuntos de transformaciones de cifrado y descifrado $\{E_e : e \in K\}$ y $\{D_d : d \in K\}$, respectivamente, donde K es el espacio de claves. El esquema de cifrado se dice de* ***clave-simétrica*** *si para cada par asociado de claves (e, d), es fácil determinar d conociendo solo e y determinar e de d.*
Puesto que $e = d$ en la mayoría de los esquemas de cifrado de clave-simétrica, el término clave-simétrica se vuelve apropiado. Otros términos usados son clave-simple, una-clave, clave-secreta y cifrado convencional.

Ejemplo 2.2 *Consideremos la codificación ASCII de caracteres. La función de cifrado y de descifrado consiste en hacer xor bit a bit entre el texto llano y la clave empleada.*

```
La longitud de la clave a usar es 32 y conincide con la longitud del
texto llano.
El texto llano es: <<este es un mensaje o texto llano>>
En hexadecimal es:
        65 73 74 65 20 65 73 20 75 6e 20 6d 65 6e 73 61
        6a 65 20 6f 20 74 65 78 74 6f 20 6c 6c 61 6e 6f
La clave a utilizar es:
        67 45 8b 6b c6 23 7b 32 69 98 3c 64 73 48 33 66
        51 dc b0 74 ff 5c 49 19 4a 94 e8 2a ec 58 55 62
El texto cifrado es:
        02 36 ff 0e e6 46 08 12 1c f6 1c 09 16 26 40 07
```

```
        3b b9 90 1b df 28 2c 61 3e fb c8 46 80 39 3b 0d
El texto llano recuperado es:
        65 73 74 65 20 65 73 20 75 6e 20 6d 65 6e 73 61
        6a 65 20 6f 20 74 65 78 74 6f 20 6c 6c 61 6e 6f
```

Dentro del esquema de cifrado de clave-simétrica se encuentran dos subcategorías: cifrado por bloques y cifrado corrido. En el cifrado por bloques el texto llano se fragmenta para ser transmitido en bloques de longitud fija t sobre un alfabeto A y se cifra bloque por bloque. En el cifrado corrido se asocia a cada texto llano una clave de longitud igual al texto cifrado, lo que sería como un cifrado por bloques, donde los bloques son de longitud 1.

Ahora se tiene una primitiva a la que se denomina firma digital.

Noción 2.4 *Una primitiva criptográfica la cual es fundamental en autenticación, autorización es la* ***firma digital****. El propósito de una firma digital es proveer un medio para una entidad de ligar su identidad con un pedazo de información. El proceso de* ***firmado*** *conlleva transformar el mensaje y alguna información secreta propiedad de la entidad en una etiqueta llamada* ***firma***

Nomenclatura.

- M es el conjunto de mensajes que pueden ser firmados.
- S es un conjunto de elementos llamados $firmas$, posiblemente cadenas binarias de longitud fija
- S_A es una transformación del conjunto de mensajes M al conjunto de firmas S, y es llamada una *transformación de firmado* por la entidad A. La transformación S_A es guardada en secreto por A, y será usada para crear firmas para mensajes de M.
- V_A es una transformación del conjunto $M \times S$ al conjunto $\{verdadero, falso\}$. V_A es llamada una *transformación de verificación* para las firmas de A, se conoce públicamente, y es usado por otras entidades para verificar firmas creadas por A.

Definición 2.3 *Las transformaciones S_A y V_A proveen un* ***esquema de firma digital*** *para A. Ocasionalmente se usa el término* ***mecanismo de firma digital****.*

Ejemplo 2.3 *Esquema de firma digital RSA simplificado.*

Generación de claves *: Se escoge de forma aleatoria $p, q \in \mathbb{Z}$ números primos de longitud $\frac{k}{2}$, en donde k es el parámetro de seguridad (en este caso la longitud en bits del módulo a utilizar). Se obtiene $n = pq$, se escoge aleatoriamente $e \in \mathbb{Z}_n$ tal que $(e, (p-1)(q-1)) = 1$, se calcula $d \in \mathbb{Z}_n$ tal que $ed \equiv 1 \mod n$. La clave pública es $PK = (n, e)$, mientras que la clave privada es $SK = (n, e, d, p, q)$.*

Firmado *: Dado un texto a firmar $m \in \mathbb{Z}_n$, su firma es $\sigma = m^d \mod n$.*

Verificación *: Dado un texto $m \in \mathbb{Z}_n$ y su firma $\sigma \in \mathbb{Z}_n$, la verificación resulta válida si $\sigma = m^e \mod n$ y fallida en otro caso.*

La desventaja de este esquema de firma mostrado en el ejemplo 2.3 es que los textos a firmar no pueden tener una codificación arbitraria, sino que deben ser elementos de $\mathbb{Z}_n$. Para poder ampliar el tipo de textos a firmar se requiere de primitivas auxiliares como se verá más adelante.

Ahora se muestra otra primitiva que es la de sistemas criptográficos de clave pública.

Definición 2.4 *Considere un esquema de cifrado que consiste de conjuntos de transformaciones de cifrado y descifrado $\{E_e : e \in K\}$ y $\{D_d : d \in K\}$, respectivamente. El método de cifrado se dice un* ***esquema de cifrado de clave pública*** *si para cada par asociado de cifrado/descifrado (e, d), una clave e (la clave pública) es puesta a disposición del público, mientras que la otra d (la clave privada) se mantienen en secreto.*

Comparando los esquemas de cifrado de clave pública con los de clave simétrica, una ventaja importante del esquema de clave simétrica es la longitud de la clave, pues es de aproximadamente 64 o 128 bits, mientras que para la clave pública se necesitan 1024 bits como es el caso de RSA [RSA78] para considerar el esquema seguro. Y esto se debe a que para poder romper el esquema de clave simétrica se necesita realizar una búsqueda exhaustiva.

Ejemplo 2.4 *Esquema de cifrado RSA simplificado.*

Generación de claves *: Se escoge de forma aleatoria $p, q \in \mathbb{Z}$ números primos de longitud $\frac{k}{2}$, en donde k es el parámetro de seguridad (en este caso la longitud en bits del módulo a utilizar). Se obtiene $n = pq$, se escoge aleatoriamente $e \in \mathbb{Z}_n$ tal que $(e, (p-1)(q-1)) = 1$, se calcula $d \in \mathbb{Z}_n$ tal que $ed \equiv 1 \mod n$. La clave pública es $PK = (n, e)$, mientras que la clave privada es $SK = (n, e, d, p, q)$.*

Cifrado *: Dado un texto llano $m \in \mathbb{Z}_n$, el texto cifrado es $c = m^e \mod n$.*

Descifrado *: Dado un texto cifrado $c \in \mathbb{Z}_n$, el texto llano es $m = c^d \mod n$.*

Ahora se verá una primitiva fundamental, la cual es llamada función Hash, normalmente llamada función Hash de un lado.

Noción 2.5 *Una **función Hash** es una función computacionalmente eficiente que mapea cadenas binarias de longitud arbitraria a cadenas binarias de alguna longitud fija, llamados valores Hash.*

Ejemplo 2.5 *Suponga que se tiene un texto de 10 caracteres, por ejemplo '1010101011' y una función Hash de 5 bits, entonces se puede truncar el texto y obtener '10101' como la codificación de '1010101011' pero es inservible para para los esquemas de firma digital.*

Para una función con salida de valores Hash de n bits y ciertas propiedades, la probabilidad de que se escoja al azar una cadena que sea mapeada a un valor Hash de n bits es 2^{-n}. La idea básica es que un valor Hash sirva como una representación compacta de una cadena de entrada. Para que sea de uso criptográfico, una función Hash h debe ser escogida de tal forma que sea computacionalmente imposible de encontrar dos distintas entradas las cuales se van al mismo valor común(es decir, dos entradas que colisionen x e y tales que $h(x) = h(y)$), y que dado un valor específico Hash y, es computacionalmente imposible encontrar una entrada x tal que $h(x) = y$ (esto último se conoce como resistencia a preimágenes).

Una última primitiva muy importante es la generación aleatoria de números, puesto que se pueden generar claves de cifrado de una forma impredecible para un adversario. Generar aleatoriamente una clave envuelve la selección de números aleatorios o secuencias de bits.

No es sencillo encontrar métodos para generar números aleatoriamente. Llamar a un número aleatorio sin contexto no tiene sentido. Si 60 bolas idénticas etiquetadas con números del 1 al 60 están en un contenedor, y este contenedor revuelve las bolas uniformemente, saca una bola y esta bola está etiquetada con el número 13, entonces se dice que 13 fue generado aleatoriamente de una distribución uniforme. La probabilidad de que salga 13 es 1 en 60 o $\frac{1}{60}$.

2.3. Complejidad de algoritmos

Ahora se trata la complejidad de algoritmos, en esta parte se clasifican los algoritmos dependiendo de su complejidad, medida en tiempo de ejecución.

Para la resolución de un problema por métodos informáticos se necesitan algoritmos y su implementación. Lo importante es poder decidir cuando un algoritmo es "mejor" que otro, un criterio es el tiempo que se tardan en terminar, otro la cantidad de memoria de la que hacen uso. Actualmente el costo de la memoria ha bajado considerablemente, por lo que este criterio no es recomendable para comparar los algoritmos, por lo que solo queda el tiempo que necesitan los algoritmos.

Un algoritmo puede ser clasificado por el número de operaciones y la cantidad de memoria que requiere para calcular una respuesta a una entrada de tamaño n.

Definición 2.5 *Un algoritmo es un procedimiento computacional bien definido que toma variables de entrada y termina con una salida.*

Es decir que un algoritmo es una especie de relación en el sentido de que toma un elemento de entrada y produce una salida.

Definición 2.6 *El **tamaño** de la entrada es el total de bits que se necesitan para representar la entrada en notación binaria ordinaria usando un esquema de codificación apropiado.*

Definición 2.7 *El **tiempo de ejecución** de un algoritmo sobre una entrada particular es el número de operaciones elementales o 'pasos' ejecutados.*

Usualmente un paso significa una operación de bits. Para algunos algoritmos será más conveniente tomar un paso como una comparación, una instrucción, etc.

Definición 2.8 *El **peor de los casos en tiempo de ejecución** de un algoritmo es una cota superior del tiempo de ejecución para cualquier entrada, expresado como una función del tamaño de entrada.*

Definición 2.9 *El **caso promedio de tiempo de ejecución** de un algoritmo es el tiempo de ejecución promedio sobre todas las entradas de un tamaño fijo, expresado como una función del tamaño de entrada.*

Órdenes de complejidad.

Tómese en cuenta el tiempo de ejecución de un programa en función de n (cantidad de datos de entrada) al que se denominará $T(n)$. Esta función $T(n)$ puede medirse físicamente, o calcularse sobre el código, contando las instrucciones que se ejecutarán y multiplicándolas por el tiempo que se necesita para cada instrucción.

La gran mayoría de los programas reales tienen algunas sentencias condicionales, haciendo que dependiendo de los datos de entrada se procesen distintas cantidades de instrucciones, lo que hace que en lugar de tener un valor único para $T(n)$ se tenga un intervalo de ellos. $T_{min}(n) \leq T_p(n) \leq T_{max}(n)$. A $T_{min}(n)$ se le conoce como 'el mejor de los casos' y a $T_{max}(n)$ se le denomina 'el peor de los casos', mientras que $T_p(n)$ es el caso promedio.

Sean $g_i(n)$, $i \in N$ distintas funciones (de tiempo de ejecución en función del tamaño de entrada n), se van a identificar familias de estas funciones, usando como criterio de identificación su comportamiento asintótico es decir cuando $n \longrightarrow \infty$.
El principal objetivo de la teoría de complejidad es dar una clasificación de problemas computacionales de acuerdo a los recursos necesarios para resolverlos. La clasificación no debe depender de un modelo de computación particular, sino que debe medir la dificultad intrínseca del problema. Los recursos para resolver un problema pueden incluir tiempo, espacio de almacenamiento, números de procesadores entre otros. Típicamente se centra la atención en el tiempo, y algunas veces en el espacio.

Definición 2.10 (Notación de Orden)

1. *Cota superior asintótica* $f(n) = O(g(n))$ *si existe una constante positiva* c *y un entero positivo* n_0 *tales que* $0 \leq f(n) \leq cg(n)\ \forall n \geq n_0$

2. *Cota inferior asintótica* $f(n) = \Omega(g(n))$ *si existe una constante positiva* c *y un entero positivo* n_0 *tales que* $0 \leq cg(n) \leq f(n)\ \forall n \geq n_0$

3. *Cota cercana asintótica* $f(n) = \Theta(g(n))$ *si existen constantes positivas* c_1 *y* c_2 *y un entero positivo* n_0 *tales que* $c_1 g(n) \leq f(n) \leq c_2 g(n)\ \forall n \geq n_0$

4. *Notación-o* $f(n) = o(g(n))$ *si para cualquier constante positiva* c *existe una constante* $n_0 > 0$ *tal que* $0 \leq f(n) \leq cg(n)\ \forall n \geq n_0$

Definición 2.11 *Al representante de un conjunto de funciones que comparten un mismo comportamiento asintótico superior se le denominará* ***orden de complejidad****.*

Es decir que para cada uno de estos conjuntos se identificará algún elemento $f(n)$ que será el representante del orden, hablándose del conjunto de funciones 'g' que son del orden de '$f(n)$'.

Se escogerán como representantes de estos órdenes a las funciones $f(n)$ más sencillas de los mismos, así tendremos:

O(1)	constante
O(log n)	logarítmico
O(n)	lineal
O($n \log n$)	
O(n^2)	cuadrático
O(n^a)	polinomial ($a > 2$)
O(a^n)	exponencial ($a > 2$)
O($n!$)	factorial

Se tiene la siguiente relación de orden en el conjunto de ordenes O(f(n)), definida por:

$f < g \iff f(n) \in \overline{g(n)}$, es decir, si $f(n)$ está en la clase de $g(n)$(denotada $\overline{g(n)}$)

Se ve que es un orden parcial, pues:

Es reflexiva $f < f$, ya que $f(n) \in \overline{f(n)}$

Es transitiva, pues si $f < g$ y $g < h$, entonces $g(n) \in \overline{h(n)}$ y $f(n) \in \overline{g(n)}$ $\therefore$ $f(n) \in \overline{h(n)} \Rightarrow f < h$

Y es antisimétrica, ya que si $f < g$ y $g < f$, $f(n) \in \overline{g(n)}$ y $g(n) \in \overline{f(n)} \Rightarrow \overline{f(n)} = \overline{g(n)}$

Dentro de la complejidad de algoritmos, hay que separar correctamente dos ideas que tienden a confundirse, estas ideas son las de **'No se conoce un algoritmo mejor'** y **'No existe un algoritmo para calcular'**, es decir que cuando se tiene un problema 'No factible', es posible que no se haya encontrado un algoritmo para resolverlo en un tiempo polinomial, o bien que NO exista ningún algoritmo que pueda resolver dicho problema, ya sea en un tiempo polinomial o en cualquier cantidad de tiempo.

2.4. Probabilidad

En esta sección se dará una introducción breve a la teoría de la probabilidad, dando algunas nociones, notaciones y los axiomas para el caso finito.

Entendemos por *evento* la ocurrencia o no ocurrencia de un fenómeno. Los eventos se denotan por letras mayúsculas A, B, C, ...

A cada evento A le corresponde un *no* A denotado por A^c y además A^c ocurre si y sólo si A no ocurre. A y A^c son eventos disjuntos, es decir no pueden ocurrir ambos al mismo tiempo.

El evento A o B se denota $A \cup B$. Si A y B son disjuntos $A \cup B = A + B$. El evento A y B se denota por AB y $AB = BA$.

Podemos dividir los eventos en dos tipos, *condiciones* y *resultados*, las condiciones son eventos que son conocidos o que están diseñados para ocurrir. Los resultados son eventos que pueden (o no) ocurrir, ocurren cuando sus condiciones ocurren.

Hay dos combinaciones de eventos que se consideran *eventos frontera*, son el 'primer' y 'último' eventos. Eventos de la forma $A + A^c$ siempre ocurren, y los eventos que resultan son equivalentes, todos ellos identifican un evento el cual es llamado *evento seguro* y se denota por Ω. La otra combinación es AA^c, los cuales son eventos que nunca ocurren e implican el mismo evento llamado *evento imposible* y denotado por $\emptyset$.

Los resultados de un experimento forman un *campo* o un *álgebra* de conjuntos.

Las condiciones de un experimento , junto con su campo de resultados constituyen un ensayo.

Los resultados de un ensayo aleatorio son llamados eventos aleatorios. El número medido por la frecuencia observada de un evento aleatorio A es llamado la probabilidad de A y es denotado por PA Claramente $P\emptyset$=0, $P\Omega$=1, y para cada A, $0 \leq PA \leq 1$

A continuación se dan los axiomas para el caso finito.

Sea Ω (o el evento seguro) un espacio de puntos ω; el conjunto vacío o el evento imposible será denotado por $\emptyset$. Sea $\mathbb{A}$ una clase de conjuntos no vacíos en Ω, llamados eventos aleatorios, o simplemente eventos. Sea P o Probabilidad una función numérica definida sobre $\mathbb{A}$; el valor de P para un evento A será llamada la probabilidad de

A y será denotada por PA. El par $(\mathbb{A}, P)$ es llamado un *campo de probabilidad*, y el triple (Ω,$\mathbb{A}$,P) es llamado un *espacio de probabilidad*.

AXIOMA I.- $\mathbb{A}$ es un álgebra
Si $A \in \mathbb{A} \Longrightarrow A^c \in \mathbb{A}$
Si $\{A_n\}_{n=1}^k \in \mathbb{A} \Longrightarrow \bigcap_{n=1}^k A_n \ \& \ \bigcup_{n=1}^k A_n \in \mathbb{A}$

AXIOMA II.- P sobre $\mathbb{A}$ está normalizada, es no negativa, y finitamente aditiva.
$P\Omega$=1
$PA > 0$
$P\sum_{n=1}^k A_n = \sum_{n=1}^k PA_n$

Ahora se verá lo que son las variables aleatorias simples.

Se toma un campo de probabilidad fijo $(\mathbb{A}, P)$. A cada evento A se le asigna una función I_A sobre Ω con valores $I_A(\omega)$ tales que $I_A(\omega) = 1$ o 0 de acuerdo a si ω pertenece o no a A respectivamente; I_A será llamado el *indicador* de A (en términos de ocurrencia $I_A = 1$ o 0 de acuerdo a si A ocurre o no ocurre). Así $I_A^2 = I_A$ y los casos frontera son aquellos de $I_\emptyset = 0$ e $I_\Omega = 1$.

La función indicador tiene las siguientes propiedades:

Si $A \subseteq B$, entonces $I_A \leq I_B$ y al contrario.
Si $A = B$, entonces $I_A = I_B$ y al contrario.
$I_{A^c} = 1 - I_A$, $I_{AB} = I_A I_B$, $I_{A+B} = I_A + I_B$.

Las combinaciones lineales $X = \sum_{j=1}^m x_j I_{A_j}$ de indicadores de eventos A_j de una partición finita de Ω , donde los x_j son números finitos, son llamadas *variables aleatorias simples* denotadas por mayúsculas $X, Y, ...$ Por convención, cada combinación lineal de indicadores es de indicadores de eventos disjuntos cuya suma es el evento seguro. El conjunto de valores PA_j que corresponden a los valores x_j de X que se asume son distintos, es llamada la *distribución de probabilidad* y los A_j forman la *partición* de X.

La *esperanza* EX de una variable aleatoria simple $X = \sum_{j=1}^m x_j I_{A_j}$ está definida por $EX = \sum_{j=1}^m x_j PA_j$. Además la esperanza de una suma de una cantidad finita de variables aleatorias simples es la suma de sus esperanzas.

La probabilidad es de vital importancia para la mecánica cuántica y por ende para el cómputo cuántico, pues no se puede medir el estado de un qubit $(a|0\rangle + b|1\rangle)$ y obtener un resultado con veracidad, sino que el resultado de la medición es 0 con probabilidad $|a|^2$ o 1 con probabilidad $|b|^2$.

2.5. Seguridad criptográfica

Así como se han creado modelos para preservar la seguridad de la información, han habido ataques para vulnerar la seguridad y poder corromper, copiar o eliminar la información que se transmite, estos ataques se pueden clasificar como ataques pasivos y ataques activos, en los primeros el adversario sólo monitorea el canal de comunicación, un atacante pasivo solo amenaza la confidencialidad de la información, mientras que un ataque activo es aquél donde el adversario intenta borrar, agregar o alterar la transmisión sobre el canal, un atacante activo amenaza la integridad de datos, la autenticación así como la confidencialidad.

Pasemos a los ataques sobre esquemas de cifrado, el objetivo de los ataques es recuperar el texto llano desde el texto cifrado , o más drástico, deducir la clave de descifrado.

- un **ataque de texto cifrado** es uno donde el adversario trata de deducir la clave de descifrado o el texto llano sólo observando el texto cifrado. Cualquier esquema de cifrado vulnerable a este tipo de ataque es considerado completamente inseguro.
- Un ataque de **texto llano conocido** es aquél donde el adversario tiene una cantidad de texto llano y su correspondiente texto cifrado. Este tipo de ataque es típicamente sólo marginalmente más difícil de realizar.
- Un ataque de **texto llano escogido** es uno donde el adversario escoge texto llano y es dado el correspondiente texto cifrado. Subsecuentemente, el adversario usa cualquier información deducida para recuperar el texto llano correspondiente al texto cifrado previo.
- Un ataque de **texto llano escogido adaptado** es un ataque de texto llano escogido donde la selección del texto llano depende el texto cifrado recibido de previos pedidos.
- Un ataque de **texto cifrado escogido** es aquél donde el adversario selecciona el texto cifrado y es dado el correspondiente texto llano. Una forma de montar

tal ataque es que el adversario obtenga acceso al equipo usado para el descifrado (aunque no para la clave de descifrado, la cual puede ser metida en el equipo de forma segura). el objetivo es ser capaz, sin tener acceso al equipo de deducir el texto llano de (distintos) textos cifrados.

- Un ataque de **texto cifrado escogido adaptable** es un ataque de texto cifrado escogido donde la elección del texto cifrado puede depender del texto llano recibido por pedidos previos.

La mayoría de estos ataques también se aplican a esquemas de firma digital.

A continuación se muestran las definiciones de seguridad para funciones Hash.

En lo siguiente, $M \xleftarrow{\$} S$ denota escoger un elemento aleatorio de la distribución S y llamarlo M. A es el adversario y Adv denota la ventaja. El símbolo $\wedge$ significa 'y'.

Definición 2.12 (Tipos de resistencia a preimágenes) *Sea H:K×M→Y una familia de funciones hash y sea m un número tal que $\{0,1\}^m \subseteq M$. Sea A un adversario. Entonces se define:*

$$Adv_H^{Pre[m]}(A) = P\Big[k \xleftarrow{\$} K;\ M \xleftarrow{\$} \{0,1\}^m;\ y \leftarrow H_k(M);\ M \xleftarrow{\$} A(k,y) : H_k(M) = y\Big]$$
$$Adv_H^{ePre[m]}(A) = max_{y \in Y}\ \Big\{P\Big[k \xleftarrow{\$} K;\ M \xleftarrow{\$} A(k) :\ H_k(M) = y\Big]\Big\}$$
$$Adv_H^{aPre[m]}(A) = max_{k \in K}\ \Big\{P\Big[M \xleftarrow{\$} \{0,1\}^m;\ y \leftarrow H_k(M);\ M' \xleftarrow{\$} A(y) :\ H_k(M') = y\Big]\Big\}$$

La primera definición, resistencia a preimagen (Pre) , es la forma usual de definir cuando una familia de funciones Hash es una función de un sólo sentido. La segunda definición, resistencia a preimagen dondequiera (ePre), captura más directamente la intuición de que es imposible encontrar la preimagen de puntos del rango: para cualquier punto del rango seleccionado es computacionalmente difícil de encontrar su preimagen. La definición final, resistente a preimagen siempre (aPre), refuerza la primera definición en el sentido que se necesita para decir que una función como SHA1 [RS04] es de un sólo sentido: uno piensa en SHA1 como una función de una familia de funciones hash y se desea decir que para esta función en particular de la familia permanece siendo difícil de encontrar una preimagen de un punto aleatorio.

Es posible formalizar múltiples definiciones que pueden ser entendidas como de significado técnico para la resistencia de segunda preimagen. En todos los casos un punto

del dominio M y una descripción de la función Hash H_k son conocidas por el adversario, cuyo trabajo es encontrar un M' distinto de M tal que $H(k, M) = H(k, M')$. Tales M y M' son llamadas parejas.

Definición 2.13 (Tipos de resistencia de segunda preimagen) *Sea $H : K \times N \to Y$ una familia de funciones Hash y sea m un número tal que $\{0,1\}^m \subseteq N$. Sea A un adversario. Entonces se define:*

$$Adv_H^{Sec[m]}(A) = P\left[k \xleftarrow{\$} K;\ M \xleftarrow{\$} \{0,1\}^m;\ M' \xleftarrow{\$} A(k, M) : (M \neq M') \wedge (H_k(M) = H_k(M'))\right]$$
$$Adv_H^{eSec[m]}(A) = max_{M \in \{0,1\}^m}\left\{P\left[k \xleftarrow{\$} K;\ M' \xleftarrow{\$} A(k) : (M \neq M') \wedge (H_k(M) = H_k(M'))\right]\right\}$$
$$Adv_H^{aSec[m]}(A) = max_{k \in K}\left\{P\left[M \xleftarrow{\$} \{0,1\}^m;\ M' \xleftarrow{\$} A(M) : (M \neq M') \wedge (H_k(M) = H_k(M'))\right]\right\}$$

La primera definición, resistencia a segunda preimagen (Sec), es la definición estándar. La segunda definición resistencia a segunda preimagen dondequiera (eSec), formaliza de manera directa que es difícil de encontrar una pareja para cualquier punto del dominio. Esta noción es llamada también una familia universal de funciones Hash en un sólo sentido (UOWHF) y fue definida por primera vez por Naor y Yung [NY89]. La definición final, resistencia a preimagen siempre (aSec), fortalece la primera en el sentido de que se necesita decir que una función como SHA1 es resistente a segunda preimagen, se piensa a SHA1 como una función de una familia de funciones Has y se desea decir que para esta función particular es difícil de encontrar una pareja para un punto aleatorio.

Finalmente, se hablará de la dificultad con la cual un adversario es capaz de encontrar dos puntos distintos en el dominio de una función Hash para el mismo punto del rango.

Definición 2.14 (Resistencia a colisiones) *Sea $H : K \times N \to Y$ una familia de funciones Hash y sea A un adversario. Entonces se define:*

$$Adv_H^{Coll}(A) = P\Big[k \xleftarrow{\$} K;\ (M, M') \xleftarrow{\$} A(k) : (M \neq M') \wedge (H_k(M) = H_k(M'))\Big]$$

No tiene sentido pensar en reforzar esta definición maximizando sobre todos los $k \in K$: para cualquier función fija $h : M \to Y$ con $|M| > |Y|$ existe un algoritmo eficiente que devuelve una M y una M' que colisionan bajo h. Mientras que este

programa podría ser difícil de encontrar en la práctica, no se conoce ninguna forma en que pueda ser formalizado.

Las funciones Hash como se verá son las funciones que soportan mejor los avances de cómputo cuántico, pues para poder romper funciones Hash se debe hacer una búsqueda exhaustiva, y hasta el momento no se conoce algún algoritmo cuántico que disminuya drásticamente el tiempo de búsqueda.

La seguridad de primitivas criptográficas puede ser evaluada bajo varios modelos diferentes. Las métricas más prácticas de seguridad son computacional, probable y ad hoc, aunque esta última es peligrosa. El nivel de confianza en la cantidad de seguridad dada por una primitiva basada en seguridad computacional o ad hoc se incrementa con el tiempo e investigación del esquema.

Seguridad incondicional
La medida más estricta es esta. Se cree que un adversario tiene recursos computacionales ilimitados, y el caso es cuando hay o no suficiente información al alcance para tirar el sistema.

Ejemplo 2.6 *Una condición necesaria para que un esquema de cifrado de clave-simétrica sea seguro incondicionalmentes es que la clave sea al menos tan larga como el mensaje.*

Seguridad teórico-compleja
Un modelo de computación apropiado es definido y los adversarios son modelados como si tuvieran poder de cómputo polinomial. Se construye entonces una prueba de seguridad relativa al modelo. Un objetivo es designar un método criptográfico basado en las posibles debilidades anticipando un adversario poderoso. Se usa un análisis asintótico y usualmente el análisis del peor de los casos para determinar cuando las pruebas tengan importancia práctica. En contraste, ataques polinomiales los cuales con factibles bajo el modelo, en la práctica todavía no son factibles computacionalmente.

Seguridad demostrable
Un método criptográfico se dice seguro demostrable si puede mostrarse que la dificultad para romperlo es esencialmente igual que la dificultad de resolver un problema ya conocido y supuestamente difícil, tal como factorización de enteros o el logaritmo discreto.

Ejemplo 2.7 *El esquema de cifrado simplificado RSA se basa en la dificultad para encontrar los factores primos de un entero grande por lo que es un ejemplo de un esquema seguro demostrable.*

Seguridad computacional
Este modelo mide la cantidad de esfuerzo computacional que se requiere por el mejor método para tirar un sistema; se debe asumir que el sistema ha sido bien estudiado para determinar que ataques son relevantes. Una técnica propuesta se dice computacionalmente segura si el nivel de cómputo necesario para tirar el sistema (usando el mejor ataque conocido) excede, por un margen confortable, los recursos de cómputo del adversario.

Seguridad Ad hoc
Esta aproximación consiste de una variedad de argumentos convincentes de que cada ataque exitoso requiere un nivel de recursos más grande que el fijado para un adversario. La primitivas criptográficas que pasan tal análisis se dicen tener seguridad heurística, con seguridad en el sentido computacional.

La seguridad de un esquema está dado por la complejidad del problema que hay que resolver para romper el esquema, como en 2.7.

Capítulo 3

Espacios de Hilbert y Mecánica Cuántica

En este capítulo se exponen los espacios de Hilbert y cómo se relacionan con la mecánica cuántica.

Se empieza definiendo lo que es un espacio de Hilbert.

Definición 3.1 *Un **espacio de Hilbert** es un espacio vectorial con producto interior que es completo con respecto a la norma vectorial definida por el producto interior*

Se dice que H es un espacio de Hilbert si es completo con respecto a dicha norma, es decir si cualquier sucesión de Cauchy en H converge en H.

El producto interior es denotado por $< \cdot , \cdot >$ y da lugar a una norma $|| \cdot ||$ inducida como $||x|| = \sqrt{< x, x >}$

Dentro de la formulación matemática de la mecánica cuántica se encuentran dos formas, una de ellas es mediante el cálculo matricial y la otra mediante los espacios de Hilbert.

En mecánica clásica, la posición de una partícula está descrita por un vector que tiene tres números reales. Existe una descripción análoga en mecánica cuántica aunque hay muchas diferencias importantes. El **estado** de un sistema mecánico cuántico está dado por un vector en un espacio vectorial llamado **ket** y denotado (con la notación de Dirac) por $| \rangle$. Al espacio se le conoce como el **espacio de estados**.

Definición 3.2 *Un **operador lineal** en un espacio vectorial H sobre un campo F es un mapeo T de H en H que cumple la siguiente condición:*

- *T(αv + βu) = αT(v) + βT(u) ∀ α, β ∈ F y u,v ∈ H*

Tomemos $H = \mathbb{R}$, $F = \mathbb{R}$ y T tal que $T(x) = 2x$, vemos que T es un operador lineal.

Además, el conjunto de todos los operadores lineales con las operaciones + y · definidas como

(T+S)(w) = T(w)+S(w) ∀ w ∈ H y

(T·S)(w) = T[S(w)] ∀ w ∈ H

forman un álgebra sobre el campo de los complejos, si $F = \mathbb{C}$

Definición 3.3 *Un operador lineal Hermitiano sobre un espacio de Hilbert es un operador lineal que sobre cierto dominio coincide con su propio operador adjunto.*

La identidad es un operador lineal Hermitiano, pues su operador adjunto es él mismo. Se dice que un operador T^* es el operador adjunto de T si se cumple que $\langle T(x), y\rangle = \langle x, T^*(y)\rangle$

Definición 3.4 *Si A es una matriz de tamaño $n \times n$ con entradas en un campo $\mathbb{F}$, y λ es un eigenvector de A, entonces la unión del vector 0 y el conjunto de todos los eigenvectores correspondientes al eigenvalor λ es un subespacio de $\mathbb{F}^n$ conocido como el eigenespacio de λ*

Definición 3.5 *Una Eigenbase es una base para un Eigenespacio.*

Ahora se introduce el concepto de observable.

Definición 3.6 *Un observable es un operador lineal Hermitiano para el cual se puede encontrar una base ortonormal del espacio de estados que consiste de los eigenvectores del operador.*

Si el espacio de estados es de dimensión finita, entonces cualquier operador Hermitiano es un observable. En la notación de Dirac, un operador es representado por una letra. Puesto que la acción de un operador sobre elementos de un espacio vectorial es un vector, una expresión de la forma $A|\Psi\rangle$ también representa un ket.

Recuérdese que un **funcional lineal** es un mapeo de un espacio vectorial al campo. El espacio dual del espacio de estados E consiste de todos los funcionales lineales que actúan sobre E y es denotado por E*. En la notación de Dirac, un elemento de E* es llamado un **bra**, y está designado por el símbolo $\langle\ |$. Se puede asociar con cualquier ket $|\Phi\rangle$ de E un elemento de E*, denotado por $\langle\Phi|$. La acción de un bra $\langle\Psi|$ sobre un ket $|\chi\rangle$ es expresado concatenando los dos símbolos $\langle\Psi|\chi\rangle$. Por definición, esta expresión es un número complejo. La correspondencia entre E y E* está relacionada a la existencia de un producto escalar en E.

Las propiedades básicas de dicho producto escalar son presentadas a continuación:

- $\langle\Phi|\Psi\rangle = \langle\Psi|\Phi\rangle^*$
- $\langle\Psi|\lambda_1\Phi_1 + \lambda_2\Phi_2\rangle = \lambda_1\langle\Psi|\Phi_1\rangle + \lambda_2\langle\Psi|\Phi_2\rangle$
- $\langle\lambda_1\Psi_1 + \lambda_2\Psi_2|\Phi\rangle = \lambda_1{}^*\langle\Psi_1|\Phi\rangle + \lambda_2{}^*\langle\Psi_2|\Phi\rangle$
- $\langle\Psi|\Psi\rangle$ es real y positivo, es cero si y sólo si $|\Psi\rangle = 0$

3.1. Axiomas de la mecánica cuántica

Para la mecánica cuántica se tiene una formulación matemática rigurosa que fue desarrollada por Paul Adrien Maurice Dirac y John von Neumann [Neu55] . Se basa en los siguientes postulados:

Postulado I. El estado de un sistema físico en un tiempo t_0 está definido por un ket $|\Psi(t_0)\rangle$ perteneciente al espacio de estados E.

Postulado II. Una cantidad medible física A está descrita por un observable A actuando sobre E.

Postulado III. Los posibles resultados en la medición de una cantidad física son los eigenvalores del observable correspondiente.

Postulado IV. Sea A una cantidad física con observable A. Suponga que el sistema está en un estado normalizado $|\Psi\rangle$. Cuando A es medido, la probabilidad $\mathrm{P}(\mathrm{a}_n)$ de obtener el eigenvalor a_n de A es

$$P(a_n) = \sum_{i=1}^{g_n} |\langle u_n^i | \Psi \rangle|^2,$$

donde g_n es la degeneración de a_n y $|u_n^1\rangle$, $|u_n^2\rangle$, . . ., $|u_n^{g_n}\rangle$ forman una base ortonormal del subespacio E_n que consiste de los eigenvectores de A con eigenvalores a_n.

Postulado V Si la medición de una cantidad A sobre un sistema físico en el estado $|\Psi\rangle$ da el resultado a_n, inmediatamente después de la medición el estado está dado por la proyección normalizada de $|\Psi\rangle$ sobre el eigenespacio E_n asociado con a_n; esto es $\frac{1}{\sqrt{\langle \Psi | P_n | \Psi \rangle}} \mathrm{P}_n |\Psi\rangle$, donde P_n es la proyección sobre E_n.

3.2. Evolución de estados cuánticos

La evolución temporal de los estados cuánticos puede obtenerse a partir del Hamiltoniano a través de la ecuación de Schrödinger. Si $|\psi(t)\rangle$ es el estado del sistema a tiempo t, tenemos:

$$H|\psi(t)\rangle = \mathrm{i}\hbar \frac{\partial}{\partial t} |\psi(t)\rangle.$$

donde $\hbar$ es la constante de Planck dividida entre 2π. Dado el estado a un tiempo inicial (t = 0), podemos integrarla para obtener el estado en cualquier tiempo subsiguiente. Si H además de operador autoadjunto no depende explícitamente del tiempo podemos encontrar una familia de operadores unitarios definidos sobre el espacio de Hilbert que da una solución formal de la anterior ecuación:

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle \qquad U(t) := \mathrm{e}^{(-\mathrm{i}Ht/\hbar)}$$

Donde la exponencial del operador Hamiltoniano se calcula usualmente mediante serie de potencias. Es un operador unitario, y es la forma común de operador de evolución temporal o propagador.

3.3. Estados entrelazados

Se les conoce como estados entrelazados a aquellos estados que están ligados de tal forma que un estado no puede ser descrito adecuadamente sin mencionar a todos los demás estados, aún cuando dichos objetos estén separados espacialmente. Esta interconección nos lleva a correlaciones entre propiedades físicas observables de sistemas remotos. Por ejemplo, la mecánica cuántica sostiene que estados tales como el spin están indeterminados hasta que alguna intervención física es hecha para medir el spin del objeto en cuestión. Es lo mismo que si cualquier partícula se vea como spin-arriba al mismo tiempo que spin-abajo. Medir cualquier cantidad de partículas resulta en una serie impredecible de medidas, que tienden más y más a la mitad arriba y la mitad abajo. De cualquier manera, si este experimento se hace con partículas entrelazadas los resultados son un poco diferentes. Cuando dos miembros de un par entrelazado son medidos, uno siempre será spin arriba y el otro spin abajo. La distancia entre las dos partículas es irrelevante. Para poder explicar este resultado, se ha teorizado que existen variables ocultas que cuentan para el spin de cada partícula y que estas variables ocultas están determinadas cuando el par entrelazado es creado.

Considérese dos sistemas que no interactúan A y B con sus respectivos espacios de Hilbert H_A y H_B. El espacio de Hilbert del sistema compuesto es el producto tensorial $H_A \otimes H_B$

Si el primer sistema está en el estado $|\psi\rangle_A$ y el segundo en el estado $|\phi\rangle_B$, el estado del sistema compuesto es $|\psi\rangle_A \otimes |\phi\rangle_B$.

Los estados del sistema compuesto que pueden ser representados en esta forma son llamados estados separables, o estados producto.

No todos los estados son estados producto. Fíjese una base $\{|i\rangle_A\}$ para H_A y una base $\{|j\rangle_B\}$ para H_B. El estado más general en $H_A \otimes H_B$ es de la forma

$$|\psi\rangle_{AB} = \sum_{i,j} c_{ij} |i\rangle_A \otimes |j\rangle_B$$

Este estado es separable si $c_{ij} = c_i^A c_j^B$, con $|\psi\rangle_A = \sum_i c_i^A |i\rangle_A$ y $|\phi\rangle_B = \sum_j c_j^B |j\rangle_B$. Es inseparable si $c_{ij} \neq c_i^A c_j^B$. Si un estado es inseparable, se le conoce como un estado entrelazado. Por ejemplo, dados dos vectores base $\{|0\rangle_A, |1\rangle_A\}$ de H_A y dos vectores base $\{|0\rangle_B, |1\rangle_B\}$ de H_B el siguiente es un estado entrelazado:

$$\frac{1}{\sqrt{2}}\left(|0\rangle_A \otimes |1\rangle_B - |1\rangle_A \otimes |0\rangle_B\right)$$

Si el sistema compuesto está en este estado, es imposible atribuir al sistema A o al sistema B un estado puro definido. En lugar de eso, sus estados están superpuestos con otros. En este sentido, los sistemas están entrelazados.

Ahora suponga que Alice es un observador para el sistema A, y Bob es un observador para el sistema B. Si Alice hace una medición en la eigenbase $\{|0\rangle, |1\rangle\}$ de A, hay dos posibles resultados con probabilidades iguales:

1. Alice mide 0, y el estado del sistema se colapsa a $|0\rangle_A|1\rangle_B$
2. Alice mide 1, y el estado del sistema se colapsa a $|1\rangle_A|0\rangle_B$

Si el primero ocurre, entonces cualquier medición siguiente que haga Bob, en la misma base dará 1. Si la segunda ocurre, entonces las mediciones de Bob darán 0. Así, el sistema B ha sido alterado por la medición local hecha por Alice en el sistema A. Esto se mantiene verdadero aún si los sistemas A y B están separados espacialmente. Esto es el fundamento de la paradoja EPR.

3.4. Teorema de no clonación y consecuencias

Una operación cuántica que copie estados sería muy útil. Por ejemplo, dado un estado cuántico desconocido, ya sea $|\phi\rangle$ o $|\psi\rangle$, realice una medición adivinar cual es. Si $|\phi\rangle$ y $|\psi\rangle$ no son ortogonales, entonces ninguna medición los distingue perfectamente, y siempre se tendrá alguna constante de probabilidad de error. Como sea, si se pudieran hacer varias copias del estado desconocido, entonces, se pueden repetir las mediciones óptimamente varias veces y hacer que la probabilidad de error sea arbitrariamente pequeña. El teorema de no clonación nos dice que esto no es físicamente posible. Solo conjuntos de estados mutuamente ortogonales pueden ser copiados por un operador simple unitario.

Asúmase que se tiene un operador unitario U_{cl} y dos estados cuánticos $|\phi\rangle$ y $|\psi\rangle$ los cuales copia U_{cl}, es decir

$$|\phi\rangle \otimes |0\rangle \xrightarrow{U_{cl}} |\phi\rangle \otimes |\phi\rangle$$

$$|\psi\rangle \otimes |0\rangle \xrightarrow{U_{cl}} |\psi\rangle \otimes |\psi\rangle$$

Entonces $\langle\phi|\psi\rangle$ es 0 o 1.

$\langle\phi|\psi\rangle$=$(\langle\phi|\otimes\langle 0|)(|\psi\rangle\otimes|0\rangle) = (\langle\phi|\otimes\langle\phi)(|\psi\rangle\otimes|\psi\rangle) = \langle\phi|\psi\rangle^2$.
En la segunda igualdad se usa el hecho de que U, siendo unitario preserva productos internos.

Dentro de las consecuencias del teorema de no clonación, se pueden mencionar, el no ser posible el uso de técnicas clásicas de corrección de errores sobre estados cuánticos, por ejemplo, no se pueden crear copias de seguridad de un estado en medio de un cálculo cuántico, en contraste, el teorema de no clonación es vital para la criptografía cuántica, ya que prohíbe a algún interceptor de crear copias de una clave criptográfica cuántica transmitida. El teorema también protege el principio de incertidumbre en mecánica cuántica. Si se pudiera clonar un estado desconocido, entonces se podrían hacer tantas copias como se deseara, y medir cada variable dinámica con precisión arbitraria, violando el principio de incertidumbre. Aún cuando es imposible realizar copias perfectas de un estado cuántico, es posible producir copias imperfectas. Esto puede ser hecho juntando una cantidad grande de sistemas auxiliares al sistema que será clonado, y aplicando una transformación unitaria al sistema combinado. Si la transformación unitaria es escogida correctamente, muchos componentes del sistema combinado evolucionarán en copias aproximadas del sistema original.

3.5. Paradoja EPR

La paradoja EPR es un experimento mental propuesto por Albert Einstein, Boris Podolsky y Nathan Rosen en [AER35], en el artículo presentado se cuestiona la completitud de la mecánica cuántica como una teoría que explique la realidad física.

El experimento planteado consiste en dos partículas que interactuaron en el pasado y que terminan en un estado entrelazado. Dos observadores reciben cada una de las partículas. Si un observador mide el momento de una de ellas, sabe cuál es el momento de la otra. Si mide la posición, gracias al entrelazamiento cuántico y al principio de incertidumbre de Heisenberg puede saber la posición de la otra partícula de forma instantánea.

La paradoja EPR está en contradicción con la teoría de la relatividad, ya que se transmite información de forma instantánea entre las dos partículas. De acuerdo a la paradoja EPR esta teoría predice un fenómeno (el de la acción a distancia instantánea) pero no permite hacer predicciones deterministas sobre él; por lo tanto, la mecánica cuántica es una teoría incompleta.

Imagínese que se realiza el experimento, Charlie prepara dos partículas. No importa como las prepara, sólo que sea capaz de repetir el procedimiento. Una vez que hizo la preparación, se manda una partícula a Alice y la segunda a Bob.

Una vez que Alice recibe su partícula, hace una medición. Suponga que se tienen dos distintas mediciones que se pueden hacer. Estas mediciones son de propiedades físicas que llamaremos P_Q y P_R, respectivamente. Alice no conoce que medición va a escoger. Cuando ella recibe la partícula, lanza una moneda para decidir que medición realizará. Se supone por simplicidad que las mediciones tienen dos posibles resultados +1 o -1. Supóngase que la partícula de Alice tiene un valor Q para la propiedad P_Q. Se asume que Q es una propiedad objetiva de la partícula de Alice, la cual es conocida por la medición. Similarmente, sea R el valor revelado por la medición de la propiedad P_R.

Similarmente suponga que Bob es capaz de medir una de dos propiedades, P_S o P_T, de nuevo revelando un valor objetivo S o T de la propiedad, tomando valores +1 o -1. Bob no decide que propiedad medirá, sino que espera a que llegue la partícula y entonces escoge al azar. Los tiempos de los experimentos están arreglados de tal forma que Alice y Bob realizan sus mediciones al mismo tiempo. Por lo tanto, la medición que Alice realice no interfiere el resultado de la medición de Bob (o vice versa), puesto que las influencias físicas no se pueden propagar más rápido que la luz.

Ahora, se realiza simple álgebra con las cantidades QS+RS+RT-QT. Nótese que

$$\text{QS+RS+RT-QT=(Q+R)S+(R-Q)T}$$

puesto que R, Q $=\pm 1$ se sigue que (Q+R)S=0 o (R-Q)T=0. En cualquiera de los casos, es fácil ver que QS+RS+RT-QT$=\pm 2$. Supóngase ahora que p(q, r, s, t) es la probabilidad que, antes de que se hagan las mediciones, el sistema esté en un estado donde Q=q, R=r, S=s y T=t. Estas probabilidades pueden depender de como Charlie haya hecho su preparación, y del ruido experimental. Sea $\mathbb{E}(\cdot)$ el valor medio de una cantidad, se tiene:

$$\mathbb{E}(QS + RS + RT - QT) = \sum_{qrst} p(q, r, s, t)(qs + rs + rt - qt)$$

$$\mathbb{E}(QS + RS + RT - QT) \leq 2 \sum_{qrst} p(q, r, s, t) = 2$$

También,

$$\mathbb{E}(QS + RS + RT - QT) = \sum_{qrst} p(q,r,s,t)qs + \sum_{qrst} p(q,r,s,t)rs$$

$$+ \sum_{qrst} p(q,r,s,t)rt - \sum_{q,r,s,t} p(q,r,s,t)qt$$

$$\mathbb{E}(QS + RS + RT - QT) = \mathbb{E}(QS) + \mathbb{E}(RS) + \mathbb{E}(RT) - \mathbb{E}(QT)$$

Comparando las ecuaciones anteriores, se obtiene la desigualdad de Bell,

$$\mathbb{E}(QS) + \mathbb{E}(RS) + \mathbb{E}(RT) - \mathbb{E}(QT) \leq 2$$

La naturaleza no obedece la desigualdad de Bell, por lo que la mecánica cuántica es una teoría completa, contradiciendo la paradoja EPR.[AER35].

Este trabajo de Einstein fue hecho para tratar de desacreditar a la mecánica cuántica, pues el famoso físico creía que las leyes del universo no deberían involucrar probabilidades, que deberían ser exactas y armónicas, esto por sus creencias religiosas.

Capítulo 4

Cómputo Cuántico

En este capítulo se ven las máquinas de Turing cuánticas, circuitos y operaciones básicas entre qubits y clases de complejidad cuánticas.

Una computadora cuántica opera manipulando los qubits con un conjunto de puertas cuánticas lógicas. Pero ¿Qué es un qubit?, así como el bit clásico tiene un estado - ya sea 0 o 1- un qubit también tiene un estado. Dos posibles estados para un qubit son los estados $|0\rangle$ y $|1\rangle$, los cuales son análogos a los estados 0 y 1 para un bit clásico. La diferencia entre los bits y los qubits es que un qubit puede estar en otro estado, distinto de 0 o 1. Es posible formar combinaciones lineales de estados, llamados superposiciones:

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

Los números α y β son números complejos, aunque para muchos propósitos no se pierde generalidad pensando en ellos como números reales. Puesto en otra forma, el estado de un qubit es un vector en un espacio vectorial de dimensión dos. Los estados $|0\rangle$ y $|1\rangle$ son conocidos como los estados base, y forman una base ortonormal para este espacio vectorial.

Se puede examinar un bit para determinar cual es su estado. Por ejemplo las computadoras hacen esto todo el tiempo cuando leen el contenido de sus memorias. Algo importante es que no se puede examinar un qubit para determinar su estado cuántico, esto es, los valores de α y β. En lugar de eso, la mecánica cuántica dice que se puede obtener solamente información mucho más restringida acerca del estado cuántico. Cuando se mide un qubit se obtiene el resultado 0, con probabilidad $|\alpha|^2$, o el resultado 1, con probabilidad $|\beta|^2$. Naturalmente, $|\alpha|^2 + |\beta|^2 = 1$, puesto que las probabilidades deben sumar 1. Geométricamente, se puede interpretar esto como

la condición de que el estado del qubit está normalizado. Así, en general el estado de un qubit es un vector unitario en un espacio vectorial complejo de dimensión dos.

Esta dicotomía entre los estados no observables de un qubit y las observaciones que se puedan hacer yace en el corazón del cómputo cuántico y la información cuántica. En la mayoría de los modelos abstractos del mundo, existe una correspondencia directa entre los elementos de la abstracción y el mundo real.

Con esta correspondencia directa a la mecánica cuántica, es difícil intuir el comportamiento de sistemas cuánticos; como sea, existe una correspondencia indirecta para que los estados de qubits puedan ser manipulados y transformados en formas que permitan medir los resultados que dependen de las diferentes propiedades del estado. Así, estos estados cuánticos tienen consecuencias reales, experimentalmente verificables, las cuales como se verá son la parte esencial del poder del cómputo cuántico y de la información cuántica.

La propiedad de un qubit de estar en un estado de superposición va en contra del 'sentido común' del entendimiento del mundo físico. Un qubit puede existir en un continuo de estados entre $|0\rangle$ y $|1\rangle$ al menos hasta que es medido. Nótese nuevamente que cuando un qubit es medido, sólo da $|0\rangle$ o $|1\rangle$ como resultado de la medición, con cierta probabilidad. Por ejemplo, un qubit puede estar en el estado:

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

el cual cuando es medido da resultado 0 el 50 % del tiempo y el resultado 1 el otro 50 % del tiempo. Este estado es denotado por $|+\rangle$.

Aún con esta extrañeza, los qubits son reales, su existencia y comportamiento validados ampliamente por experimentos. Por ejemplo, dos estados de un electrón orbitando un átomo simple, el electrón puede existir en los estados 'base' o 'excitado', los cuales serán $|0\rangle$ y $|1\rangle$ respectivamente. Irradiando el átomo con energía apropiada por un tiempo apropiado, es posible mover el electrón del estado $|0\rangle$ al estado $|1\rangle$ y vice versa. Pero algo más interesante es que reduciendo el tiempo de la radiación, un electrón inicialmente en el estado $|0\rangle$ puede ser movido 'la mitad del camino' entre $|0\rangle$ y $|1\rangle$, en el estado $|+\rangle$.

Una imagen útil para pensar en qubits es la siguiente representación geométrica. Puesto que $|\alpha|^2 + |\beta|^2 = 1$ se puede reescribir $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ como:

$$|\psi\rangle = e^{i\gamma}\left(cos\frac{\theta}{2}|0\rangle + e^{i\varphi}sen\frac{\theta}{2}|1\rangle\right)$$

donde θ, φ y γ son números reales. Puesto que $e^{i\gamma}$ no tiene efectos observables, se puede escribir

$$|\psi\rangle = cos\frac{\theta}{2}|0\rangle + e^{i\varphi}sen\frac{\theta}{2}|1\rangle$$

Los números θ y φ definen un punto sobre la esfera unitaria tridimensional, como se muestra en la figura siguiente.

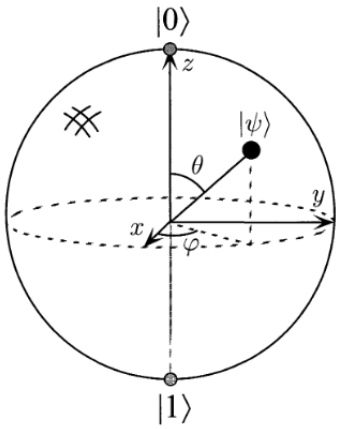

Figura 4.1: Representación de un qubit en la esfera de Bloch

Esta esfera es llamada la esfera de Bloch; da un significado útil de visualización del estado de un qubit simple, y sirve para probar ideas acerca del cómputo cuántico e información cuántica. Como sea, se debe tener en mente que esta intuición es limitada, pues no hay una generalización simple conocida de la esfera de Bloch para varios qubits, [NC00].

4.1. Máquinas de Turing cuánticas

En 1985, David Deutsch presentó el diseño de la primera Máquina Cuántica basada en una máquina de Turing. Con este fin enunció una nueva variante de la tesis de Church, dando lugar al denominado *'Principio de Church-Turing-Deutsch'*.

La estructura de una máquina de Turing cuántica es muy similar a la de una máquina de Turing clásica.

Una máquina de Turing Cuántica es a *grosso* modo una máquina de Turing probabilística con amplitudes de transición complejas en lugar de reales.

En lo siguiente, la máquina de Turing clásica es una máquina con una cinta infinita con dos lados empezando en la posición 0 de la cinta. Una máquina de Turing cuántica correspondiente trabaja como sigue:

1.- El espacio de estados cuánticos de la máquina está generado por una base consisitiendo de estados $|h\rangle|q_c\rangle|x_c\rangle|T_c\rangle$, donde $h \in \{0, 1\}$ y (q_c, x_c, T_c) es una configuración de la máquina clásica correspondiente, donde x_c denota la posición del cabezal, q_c el estado interno de la máquina y T_c el contenido no blanco de la cinta. T_c incluye una indicación de la posición absoluta del contenido de la cinta.

2.- Los estados internos inicial y terminal se identifican.

3.- La regla de transición es ahora un operador unitario el cual, en cada paso, mapea cada básico $|h\rangle|q\rangle|x\rangle|T\rangle$ a una superposición finita de elementos de la forma $|h'\rangle|q'\rangle|x'\rangle|T'\rangle$, donde

- T' y T difieren a lo más en posición x;
- $|x' - x| \leq 1$;
- $h' = 1$ siempre que q' es el estado donde se detiene la máquina clásica.
- $T' = T$, $q' = q$ y $h' = h$ siempre que $h = 1$

4.- La máquina es iniciada con una superposición finita de entradas en el estado inicial. Por la forma en que trabaja la regla de transición, la máquina estará en la superposición de sólo una cantidad finita de estados básicos $|h\rangle|q\rangle|x\rangle|T\rangle$ en cualquier paso durante todo el cálculo.

4.2. Circuitos y operaciones básicas entre qubits

El desarrollo de las herramientas de cómputo cuántico empiezan con operaciones en el sistema más simple de todos, un qubit. un qubit es un vector $|\psi\rangle = a|0\rangle + b|1\rangle$ con $a, b \in \mathbb{C}$ y que satisfacen $|\alpha|^2 + |\beta|^2 = 1$. Las operaciones sobre un qubit deben preservar su norma, así que son descritas por matrices unitarias de tamaño 2×2. De estas, algunas de las más importantes son las matrices de Pauli:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}; Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Otras tres puertas cuánticas que se usan ampliamente son la puerta de Hadamard (H), la puerta de fase (S), y la puerta $\pi/8$ (T):

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}; T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}$$

Es útil notar que $H = (X + Z)/\sqrt{2}$ y $S = T^2$. También recuérdese que un qubit en el estado $a|0\rangle + b|1\rangle$ puede ser visualizado como un punto (θ, φ) sobre la esfera unitaria, donde $a = \cos(\theta/2)$, $b = e^{i\varphi}\text{sen}(\theta/2)$, y a puede ser real. Esta forma es llamada la representación de la esfera de Bloch, y el vector $(\cos\varphi\text{sen}\theta, \text{sen}\varphi\text{sen}\theta, \cos\theta)$ es llamado el vector Bloch.

Las matrices de Pauli dan pie a tres clases de matrices unitarias muy útiles, los *operadores de rotación* alrededor de los ejes x, y y z definidos por las ecuaciones:

$$R_x(\theta) = e^{-i\theta X/2} = cos\frac{\theta}{2}I - isen\frac{\theta}{2}X = \begin{bmatrix} cos\frac{\theta}{2} & -isen\frac{\theta}{2} \\ -isen\frac{\theta}{2} & cos\frac{\theta}{2} \end{bmatrix}$$

$$R_y(\theta) = e^{-i\theta Y/2} = cos\frac{\theta}{2}I - isen\frac{\theta}{2}Y = \begin{bmatrix} cos\frac{\theta}{2} & -sen\frac{\theta}{2} \\ sen\frac{\theta}{2} & cos\frac{\theta}{2} \end{bmatrix}$$

$$R_z(\theta) = e^{-i\theta Z/2} = cos\frac{\theta}{2}I - isen\frac{\theta}{2}Z = \begin{bmatrix} e^{\frac{-i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{bmatrix}$$

Un operador unitario arbitrario sobre un qubit puede escribirse de muchas formas como combinación de rotaciones, con un cambio de fase global sobre el qubit.

Teorema 4.1 ***(Descomposición en Z-Y para un qubit)*** *Suponga que U es una operación unitaria sobre un qubit. Entonces existen números reales α, β, γ y δ tales que*

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

La utilidad del teorema anterior yace en el siguiente corolario, el cual es la clave para la construcción de operaciones unitarias controladas sobre varios qubits.

Corolario 4.2 *Suponga que U es una puerta unitaria sobre un qubit. Entonces existen operadores unitarios A, B, C sobre un qubit, tales que $ABC=I$ y $U=e^{i\alpha}AXBXC$, donde α es algún factor de fase global.*

Los símbolos para las puertas comunes de un qubit se muestran a continuación. Recuerde las propiedades básicas de los circuitos cuánticos, el tiempo transcurre de izquierda a derecha; los cables representan qubits, y una '/' puede usarse para indicar un paquete de qubits.

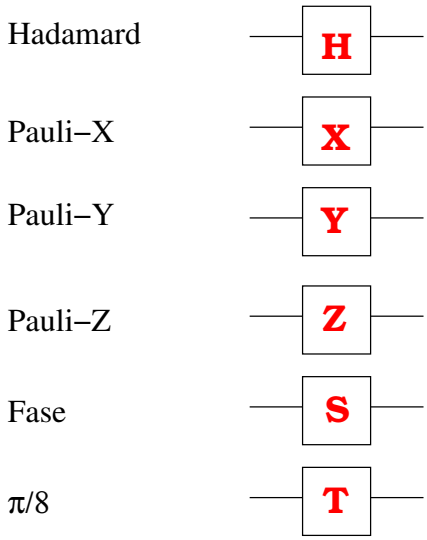

Figura 4.2: Nombre y representación de las puertas unitarias comunes sobre un qubit

'Si A es verdad, entonces haz B'. Este tipo de *operación controlada* es una de las más usadas en cómputo, ambos clásico y cuántico.

La operación controlada típica es la operación CNOT, la cual tiene como entrada dos qubits, conocidos como el *qubit de control* y el *qubit blanco*, respectivamente. En términos de la base computacional, la acción de CNOT está dada por $|c\rangle|t\rangle \rightarrow |c\rangle|t \oplus c\rangle$; esto es, si el qubit de control se coloca en $|1\rangle$ entonces el qubit blanco es

cambiado, de otra forma el qubit blanco se deja solo. Así, en la base computacional la representación matricial de CNOT es:

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

Más generalmente, suponga que U es una operación unitaria sobre un qubit. Una operación U *controlada* es una operación sobre dos qubits, de nuevo con un bit de control y un bit blanco. Si el qubit de control es fijado entonces U es aplicada sobre el qubit blanco, de otra forma el qubit blanco se deja solo; esto es, $|c\rangle|t\rangle \rightarrow |c\rangle U^c|t\rangle$. La operación controlada U se representa a continuación.

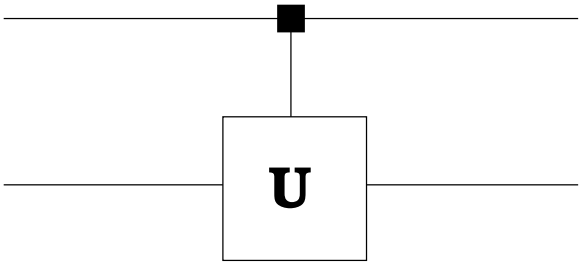

Figura 4.3: La línea superior es el qubit de control y la inferior es el qubit blanco

Un pequeño conjunto de puertas (por ejemplo AND, OR y NOT) pueden ser usadas para calcular una función clásica arbitraria. Se dice que tal conjunto de puertas es *universal* para el cómputo clásico. Un resultado de universalidad similar se tiene para cómputo cuántico, donde el conjunto de puertas se dice *universal para cómputo cuántico* si cualquier operación unitaria puede ser aproximada con precisión arbitraria por un circuito que conlleve tales puertas. Ahora se describen tres construcciones universales para el cómputo cuántico. Estas construcciones aproximan a cualquier operación unitaria, las cuales usan las puertas: Hadamard, fase, CNOT y $\pi/8$.

La primera construcción muestra que un operador unitario arbitrario puede ser expresado *exactamente* como un producto de operadores unitarios donde cada uno actúa de forma no trivial solo sobre un subespacio generado por dos estados base computacionales. La segunda construcción utiliza la primera construcción para mostrar que un operador unitario arbitrario puede ser expresado *exactamente* usando qubit simples y puertas CNOT. La tercera construcción combina la segunda construcción con

una demostración que una operación de qubits simples puede ser aproximada con precisión arbitraria usando las puertas de Hadamard, fase y $\pi/8$. Esto implica que cualquier operación unitaria puede ser aproximada con precisión arbitraria usando las puertas Hadamard, fase, CNOT y $\pi/8$.[NC00]

4.3. Clases de complejidad cuántica

Al igual que en cómputo clásico, en cómputo cuántico existen clases de complejidad, donde se clasifican los algoritmos de acuerdo a su complejidad.

Usaremos el alfabeto Σ=$\{0, 1\}$. El conjunto Σ^* denotará todas las cadenas de longitud finita sobre el alfabeto Σ. Un lenguaje L es un subconjunto de Σ^*.

Un algoritmo 'resuelve el problema de reconocimiento del lenguaje L' si acepta cualquier cadena $x \in L$ y rechaza cualquier cadena $x \notin L$

La clase BPP (del inglés bounded-error probabilistic polynomial time) consiste de todos los lenguajes L para los cuales existe un algoritmo clásico aleatorio A con tiempo polinomial en el peor de los casos tal que para cualquier entrada x$\in \Sigma^*$ se tiene:

- Si $x \in L$ entonces la probabilidad de que A acepte x es al menos $\frac{2}{3}$
- Si $x \notin L$ entonces la probabilidad de que A acepte x es a lo más $\frac{1}{3}$

La clase BPQ (bounded-error quantum polynomial time) consiste de todos los lenguajes L para los cuales existe un algoritmo cuántico A corriendo en tiempo polinomial en el peor de los casos tal que para cualquier entrada $x \in \Sigma^*$ se tiene:

- Si $x \in L$ entonces la probabilidad de que A acepte x es al menos $\frac{2}{3}$
- Si $x \notin L$ entonces la probabilidad de que A acepte x es a lo más $\frac{1}{3}$

A continuación se muestra la relación conocida entre las clases de complejidad clásicas y cuánticas.

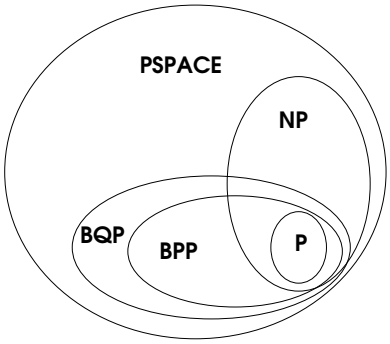

Figura 4.4: Relación conocida entre clases de complejidad clásicas y cuánticas

La clase PSPACE es la clase de los lenguajes para los que existe una máquina de Turing clásica determinista que 'lo decide' en **espacio polinomial** [Sip97].

El reto más grande de los algoritmos cuánticos es encontrar problemas que están en BQP pero no en BPP; esto es, encontrar problemas que son solubles eficientemente sobre una computadora cuántica pero no en una computadora clásica. El estudio de estas clases de complejidad y las relaciones entre ellas puede ser muy útil para entender la dificultad de estos problemas.

Capítulo 5

Algoritmos Cuánticos y su afectación a la criptografía

En el presente capítulo se explican algunos algoritmos cuánticos y como reducen el tiempo para resolver algunos problemas que en cómputo clásico crecen exponencialmente. Los algoritmos descritos en este capítulo son Deutsch, Deutsch-Jozsa, Shor, Grover y Regev, donde los que más ventaja obtienen de la mecánica cuántica para mostrar que una computadora cuántica es más poderosa que una clásica son Shor y Grover, el primero puede romper el esquema RSA y el segundo reduce el tiempo de búsqueda al intentar romper un esquema de cifrado simétrico.

En el año de 1982 el físico Richard Feynman pensó en la idea de una 'computadora cuántica', una computadora que usa los efectos de la mecánica cuántica como ventaja. Por algún tiempo, la noción de una computadora cuántica fue por interés meramente teórico; pero avances recientes han atraído la atención tanto de investigadores como de inversionistas. Uno de tales avances fue la invención de un algoritmo para factorizar números enormes sobre una computadora cuántica, por Peter Shor (Laboratorios Bell). Usando este algoritmo, una computadora cuántica sería capaz de romper códigos mucho más rápidamente que una computadora ordinaria (o clásica).

El esquema RSA es en la actualidad el método usado más comúnmente para enviar datos cifrados. Trabaja usando dos claves, una pública y otra privada. La clave pública es usada para cifrar los datos, mientras que la privada es usada para descifrar los datos. La clave pública puede ser generada por la clave privada; pero no viceversa. Un interceptor que haya adquirido la clave pública puede en principio calcular la clave privada, puesto que están relacionadas matemáticamente. Para hacerlo es necesario

factorizar la clave pública, un proceso que es considerado intratable, es decir, que requiere de una cantidad considerable de tiempo de cómputo.

Por ejemplo, multiplicar 1234 por 3433 es fácil de calcular, pero calcular los factores de 4236322 no es tan fácil. La dificultad de factorizar un número crece rápidamente con dígitos adicionales. Toma 8 meses y 1600 usuarios de internet para romper RSA129 (un número de 129 dígitos); pero tomaría más que la edad del universo factorizar RSA140. De cualquier forma, usando una computadora cuántica, corriendo el algoritmo de Shor, la cantidad de dígitos en la clave tiene un pequeño efecto sobre la dificultad del problema. Romper RSA140 tomaría sólo unos cuantos segundos.

Por otro lado, el algoritmo de Grover tiene también una aplicación muy útil en criptografía. Es posible teóricamente usar este algoritmo para romper el esquema DES, un estándar que es usado para proteger, entre otras cosas transacciones financieras entre bancos. El estándar se basa en un número de 56 bits que ambos participantes deben conocer con anterioridad, el número es usado como clave para cifrar y descifrar los datos.

Si un documento cifrado y su fuente pueden obtenerse, es posible tratar de encontrar la clave de 56 bits. Una búsqueda exhaustiva por métodos convencionales haría necesario buscar 2^{55} claves antes de encontrar la correcta. Esto tomaría más de un año aún cuando se probaran un billón de claves cada segundo, en comparación, el algoritmo de Grover podría encontrar la clave después de sólo 185 búsquedas.

5.1. Deutsch

El problema resuelto por el algoritmo de Deutsch es el siguiente:

Supóngase que se tiene una función desconocida $f : \{0,1\} \rightarrow \{0,1\}$, donde esta función es como un oráculo, es decir, se puede aplicar varias veces la función para obtener distintos valores de $f(x)$ dado un valor x; pero no se puede saber como trabaja la función. El problema es determinar el valor de $f(0) \oplus f(1)$. Si se tiene que $f(0) \oplus f(1) = 0$, entonces se sabe que $f(0) = f(1)$ aunque no se sepa que valor es, y se dice que f es constante. Por otro lado, si se obtiene que $f(0) \oplus f(1) = 1$, entonces se sabe que $f(0) \neq f(1)$, y se dice que la función está balanceada. Así, determinar $f(0) \oplus f(1)$ es equivalente a determinar si la función f es constante o está balanceada.

Entonces el problema de Deutsch se reduce a determinar el valor de $f(0) \oplus f(1)$ haciéndole preguntas a f. ¿Cuántas preguntas se le deben hacer al oráculo clásico

de f para determinar $f(0) \oplus f(1)$?, claramente la respuesta es dos, supóngase que calculamos $f(0)$ usando una pregunta (clásica), entonces el valor de $f(1)$ puede ser 0 o 1, por lo que no se puede deducir cuánto vale $f(0) \oplus f(1)$. El algoritmo de Deutsch es un algoritmo cuántico capaz de determinar el valor de $f(0) \oplus f(1)$ haciéndole sólo una pregunta a un oráculo cuántico para f.

El circuito reversible dado para f puede convertirse en un circuito cuántico reemplazando cada puerta clásica reversible en el circuito por una puerta cuántica unitaria análoga. Este circuito cuántico puede ser expresado como un operador unitario

$$U_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$$

Teniendo la versión cuántica del circuito para f se pueden proveer qubits como entradas. Se define U_f tal que si se queda fijo el segundo qubit de entrada en el estado $|y\rangle = |0\rangle$, entonces $|0\rangle = |0\rangle$ en el primer qubit de entrada nos dará $|0 \oplus f(0)| = |f(0)\rangle$ en el segundo qubit de salida y $|x\rangle = |1\rangle$ en el primer qubit de entrada dará $|0 \oplus f(1)\rangle = |f(1)\rangle$ en el segundo qubit de salida. Así se puede pensar en $|x\rangle = |0\rangle$ como la versión cuántica de la entrada clásica 0, y análogamente con $|1\rangle$ y 1.

Supóngase que se deja el segundo qubit de entrada en el estado $|y\rangle = |0\rangle$ y se coloca el primer qubit de entrada en el estado con superposición

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

Entonces los dos qubits de entrada para U_f son

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

La salida de U_f será el estado

$$\begin{aligned} U_f\left(\tfrac{1}{\sqrt{2}}|0\rangle|0\rangle + \tfrac{1}{\sqrt{2}}|1\rangle|0\rangle\right) &= \tfrac{1}{\sqrt{2}}U_f|0\rangle|0\rangle + \tfrac{1}{\sqrt{2}}U_f|1\rangle|0\rangle \\ &= \tfrac{1}{\sqrt{2}}|0\rangle|0 \oplus f(0)\rangle + \tfrac{1}{\sqrt{2}}|1\rangle|0 \oplus f(1)\rangle \\ &= \tfrac{1}{\sqrt{2}}|0\rangle|f(0)\rangle + \tfrac{1}{\sqrt{2}}|1\rangle|f(1)\rangle \end{aligned}$$

En algún sentido, U_f ha calculado simultáneamente el valor de f en las dos posibles entradas 0 y 1 en superposición, ahora, si se mide el estado de salida en la base

computacional, se tiene $|0\rangle|f(0)\rangle$ (con probabilidad 1/2) o $|1\rangle|f(1)\rangle$ (con probabilidad 1/2), después de la medición el estado de salida será $|f(0)\rangle$ o $|f(1)\rangle$ y así las siguientes mediciones del estado de salida darán $|f(0)\rangle$ o $|f(1)\rangle$, esto significa que aunque se hayan calculado dos valores en superposición, solo uno de estos valores estará disponible a través de una medición.

Recuérdese que para el problema de Deutsch no estamos interesados en los valores $f(0)$ ni $f(1)$ sino en $f(0) \oplus f(1)$. El algoritmo de Deutsch muestra como se usa la interferencia cuántica para obtener información global acerca de la función f y de forma mucho más eficiente que la clásica. El algoritmo de Deutsch se ha implementado en el circuito de la figura 5.1.

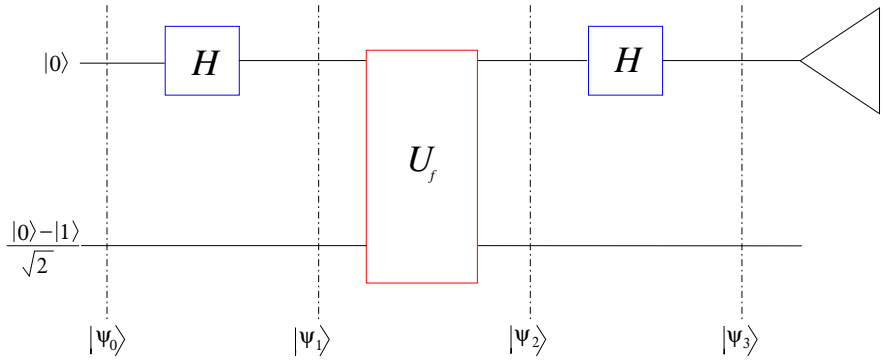

Figura 5.1: Un circuito para implementar el algoritmo de Deutsch

Nótese que el segundo qubit de entrada se ha fijado en el estado $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$, este estado se puede generar al aplicar una puerta simple de Hadamard al estado $|1\rangle$. Se analizará el comportamiento de los estados en cada puerta del circuito.

Primero, el estado de entrada es:

$$|\psi_0\rangle = |0\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

Después de la primera puerta de Hadamard aplicada al primer qubit, el estado se convierte en:

$$\begin{aligned} |\psi_1\rangle = & \left(\tfrac{1}{\sqrt{2}}|0\rangle + \tfrac{1}{\sqrt{2}}|1\rangle \right) \left(\tfrac{|0\rangle-|1\rangle}{\sqrt{2}} \right) \\ = & \tfrac{1}{\sqrt{2}}|0\rangle \left(\tfrac{|0\rangle-|1\rangle}{\sqrt{2}} \right) + \tfrac{1}{\sqrt{2}}|1\rangle \left(\tfrac{|0\rangle-|1\rangle}{\sqrt{2}} \right) \end{aligned}$$

Luego aplicando U_f obtenemos:

$$\begin{aligned}|\psi_2\rangle = & \frac{(-1)^{f(0)}}{\sqrt{2}}|0\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)+\frac{(-1)^{f(1)}}{\sqrt{2}}|1\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\\ = & \left(\frac{(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\\ = & (-1)^{f(0)}\left(\frac{|0\rangle+(-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\end{aligned}$$

donde la última igualdad usa los hechos de que $(-1)^{f(0)}(-1)^{f(1)} = (-1)^{f(0)\oplus f(1)}$ y $f(0)+f(0) = f(1)+f(1) = 0$, pues $f(0)$ puede ser 0 o 1, si es 0, 0+0=0 y si es 1, 1+1=0, análogamente para $f(1)$.

Si f es una función constante, (es decir $f(0)\oplus f(1)$=0), entonces tenemos:

$$|\psi_2\rangle = (-1)^{f(0)}\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

y así la última puerta de Hadamard sobre el primer qubit transforma el estado en

$$|\psi_3\rangle = (-1)^{f(0)}|0\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

La norma al cuadrado del estado básico $|0\rangle$ en el primer qubit es 1, esto significa que para una función constante la medición del primer qubit da 0.

Si f es una función balanceada (es decir $f(0)\oplus f(1) = 1$, entonces se tiene

$$|\psi_2\rangle = (-1)^{f(0)}\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

y la última puerta de Hadamard transforma el estado en

$$|\psi_3\rangle = (-1)^{f(0)}|1\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

En este caso la norma al cuadrado del estado básico $|1\rangle$ en el primer qubit da como resultado 1, esto significa que para una función balanceada, la medición del primer qubit es 1.

Así, la medición del primer qubit al final del circuito para el algoritmo de Deutsch da el valor de $f(0)\oplus f(1)$ y así se sabe si la función es constante o está balanceada.[PKM07]

Aunque este algoritmo no afecta a la criptografía, es un ejemplo claro de una diferencia que existe entre el cómputo clásico y el cómputo cuántico, pues, mientras que en el cómputo clásico para poder obtener el resultado se tiene que hacer dos preguntas al oráculo, usando la superposición de estados cuánticos, al cómputo cuántico le basta hacer sólo una pregunta.

5.2. Deutsch-Jozsa

El algoritmo de Deutsch-Jozsa [DJ92] resuelve un problema que es una generalización directa del problema resuelto por el algoritmo de Deutsch. Se tiene un circuito reversible que implementa una función desconocida f, pero esta vez f es una función de cadenas de n bits a un sólo bit, esto es:

$$f : \{0,1\}^n \to \{0,1\}$$

También se nos da el hecho de que f es constante (es decir $f(x)$ es el mismo para toda x), o está balanceada (es decir $f(x) = 0$ exactamente para la mitad de las cadenas de entrada x y $f(x) = 1$ para la otra mitad de las entradas). El problema es determinar si f es constante o balanceada haciendo preguntas al circuito de f.

Considere resolver este problema mediante un algoritmo clásico, supóngase que hemos usado el oráculo para determinar $f(x)$ para exactamente la mitad de las posibles entradas (es decir se han hecho 2^{n-1} preguntas a f), y todas las preguntas han dado como respuesta $f(x) = 0$. En este punto se puede sospechar que f es constante; sin embargo es posible que si se preguntase a f de las entradas restantes se pueda obtener $f(x) = 1$ cada vez. Así que es posible decir que f está balanceada. Así en el peor caso, usando un algoritmo clásico no se puede decidir con certeza si f es constante o está balanceada con menos de $2^{n-1} + 1$ preguntas. La propiedad de ser constante o estar balanceada es una propiedad global de f. El algoritmo de Deutsch-Jozsa determinará la propiedad de f haciendo sólo una pregunta a una versión cuántica del circuito de f.

Análogamente al algoritmo de Deutsch, se define la operación cuántica

$$U_f : |\hat{x}\rangle|y\rangle \mapsto |\hat{x}\rangle|y \oplus f(x)\rangle$$

Esta vez ponemos $\hat{x}$ por tratarse de una cadena de n bits, como antes, se llama U_f como un operador $\hat{U}_{f(\hat{x})}$ de un qubit, esta vez controlada por el registro de qubits

en el estado $|\hat{x}\rangle$. Se puede notar que $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ es un eigenvalor de $\hat{U}_{f(\hat{x})}$ con eigenvalor $(-1)^{f(\hat{x})}$.

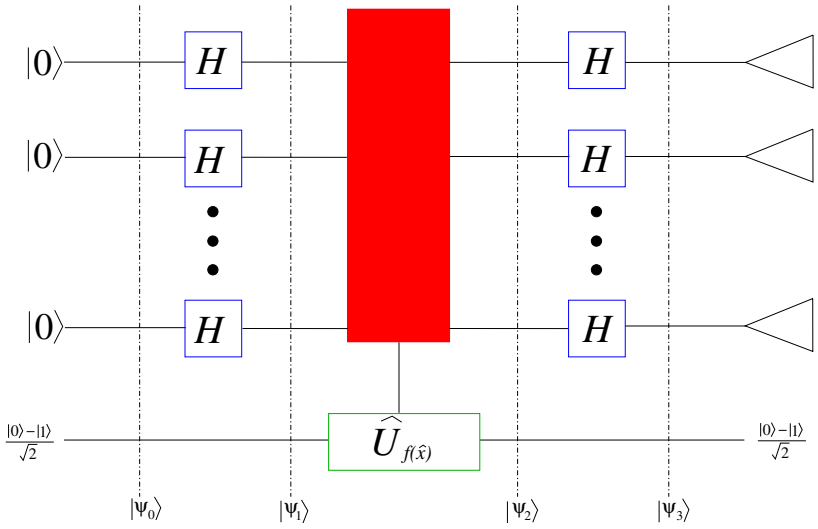

Figura 5.2: Un circuito para implementar el algoritmo de Deutsch-Jozsa

Nótese la similitud entre el circuito para el algoritmo de Deutsch y el circuito para el algoritmo de Deutsch-Jozsa (figura 5.2). En lugar de una puerta de Hadamard de un qubit se tienen productos tensoriales de n puertas de Hadamard de un qubit (actuando en paralelo). Esto se denota por $H^{\otimes n}$, se usa $|0\rangle^{\otimes n}$, o $|\hat{0}\rangle$ para denotar el estado que es el producto tensorial de n qubits, cada uno en el estado $|0\rangle$

Como se hizo para el algoritmo de Deutsch, se sigue el estado a través del circuito. Al inicio el estado es

$$|\psi_0\rangle = |0\rangle^{\otimes n}\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Considerando la acción de una transformación de Hadamard de n qubits sobre el estado $|0\rangle^{\otimes n}$ se tiene

$$|H^{\otimes n}\rangle|0\rangle^{\otimes n} = \left(\frac{1}{\sqrt{2}}\right)^n \underbrace{(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes \cdots (|0\rangle + |1\rangle)}_{n-veces}$$

esto se reescribe como

$$|H^{\otimes n}\rangle|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{\hat{x}\in\{0,1\}^n} |\hat{x}\rangle$$

Esta es una forma muy común y útil de escribir este estado, la puerta de Hadamard de n qubits actuando sobre el estado de n qubits siendo todos cero nos da una superposición de todos los estados básicos de n qubits, todos con la misma amplitud, a saber $\frac{1}{\sqrt{2^n}}$. Así el estado inmediato después de la primera puerta $H^{\otimes n}$ en el algoritmo de Deutsch-Jozsa es

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{\hat{x}\in\{0,1\}^n} |\hat{x}\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Nótese que el registro de la pregunta es ahora una superposición de igual peso para todas las posibles cadenas de entrada de n qubits, se considera ahora el estado inmediato después de la puerta $\hat{U}_f$, el estado es

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \hat{U}_f \left(\sum_{\hat{x}\in\{0,1\}^n} |\hat{x}\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\right)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{\hat{x}\in\{0,1\}^n} (-1)^{f(\hat{x})}|\hat{x}\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

donde se ha asociado el cambio de fase de $(-1)^{f(\hat{x})}$ con el primer qubit.

El efecto de la puerta de Hadamard de un qubit sobre un estado básico $|y\rangle$ puede escribirse como:

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

$$H|y\rangle = \frac{1}{\sqrt{2}} \sum_{z\in\{0,1\}} (-1)^{yz}|z\rangle$$

Entonces la acción de la transformación de Hadamard sobre un estado básico de n qubits $|\hat{x}\rangle = |x_1\rangle|x_2\rangle...|x_n\rangle$ está dada por

$$\begin{aligned} H^{\otimes n}|\hat{x}\rangle &= H^{\otimes n}\left(|x_1\rangle|x_2\rangle...|x_n\rangle\right) \\ &= H|x_1\rangle H|x_2\rangle...H|x_n\rangle \\ &= \tfrac{1}{\sqrt{2}}\left(|0\rangle+(-1)^{x_1}|1\rangle\right)\tfrac{1}{\sqrt{2}}\left(|0\rangle+(-1)^{x_2}|1\rangle\right)\cdots\tfrac{1}{\sqrt{2}}\left(|0\rangle+(-1)^{x_n}|1\rangle\right) \end{aligned}$$

por lo que tenemos finalmente

$$H^{\otimes n}|\hat{x}\rangle = \frac{1}{\sqrt{2^n}}\sum_{z_1z_2...z_n\in\{0,1\}^n}(-1)^{x_1z_1+x_2z_2+...+x_nz_n}|z_1\rangle|z_2\rangle\cdots|z_n\rangle$$

La ecuación anterior puede ser escrita como:

$$H^{\otimes n}|\hat{x}\rangle = \frac{1}{\sqrt{2^n}}\sum_{\hat{z}\in\{0,1\}^n}(-1)^{\hat{x}\cdot\hat{z}}|\hat{z}\rangle$$

donde $\hat{x}\cdot\hat{z}$ denota el producto interno bit a bit de $\hat{x}$ y $\hat{z}$ módulo 2 (puesto que $(-1)^2=1$). Nótese que la adición módulo 2 es lo mismo que la operación XOR. El estado después de la última puerta de Hadamard en el algoritmo de Deutsch-Jozsa es:

$$|\psi_3\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{\hat{x}\in\{0,1\}^n}(-1)^{\hat{x}\cdot\hat{z}}|\hat{z}\rangle\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

que nos resulta

$$|\psi_3\rangle = \frac{1}{2^n}\sum_{\hat{z}\in\{0,1\}^n}\left(\sum_{\hat{x}\in\{0,1\}^n}(-1)^{f(\hat{x})+\hat{x}\cdot\hat{z}}\right)|\hat{z}\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

Al final del algoritmo una medición del primer registro es hecho, para ver que pasa, considere la amplitud total (coeficiente) de $|\hat{z}\rangle = |0\rangle^{\otimes n}$ en el primer registro del estado $|\psi_3\rangle$. Esta amplitud es:

$$\frac{1}{2^n}\sum_{\hat{x}\in\{0,1\}^n}(-1)^{f(\hat{x})}$$

Considere esta amplitud en los dos casos, f constante y f balanceada. Si f es constante, la amplitud $|0\rangle^{\otimes n}$ es $+1$ o -1 (dependiendo de los valores que tome $f(x)$) así, si f es constante, una medición del primer registro regresará únicamente ceros (es decir la cadena binaria $00\cdots0$). Por otro lado, si f está balanceada, entonces las contribuciones positivas y negativas se cancelan, y la amplitud total de $|0\rangle^{\otimes n}$ es 0. Así, si f está balanceada una medición del primer registro no regresará todos cero, para determinar si f es constante o balanceada, el primer registro es medido, si el resultado de la medición es todo 0, entonces el algoritmo da como resultado 'constante', y de otra forma da 'balanceada'.

Consecuencias: Este algoritmo fue propuesto por David Deutsch y Richard Jozsa en 1992, fue uno de los primeros algoritmos diseñados para ejecutar sobre una computadora cuántica y que tiene el potencial de ser más eficiente que los algoritmos clásicos al aprovechar el paralelismo inherente de los estados de superposición cuánticos. Este algoritmo ni el anterior (Deutsch) conllevan una reducción de tiempo de forma exponencial como el que nos da el siguiente algoritmo.

5.3. Shor

Un método común de criptografía de clave pública actual se basa en la dificultad que existe para factorizar números enteros (en sus factores primos) lo suficientemente grandes en un tiempo aceptable. El algoritmo de Shor se basa en propiedades cuánticas para obtener un algoritmo que factoriza números enteros grandes en un tiempo aceptable.

El mejor algoritmo clásico (publicado) necesita $O(e^{\frac{64}{9}\frac{1}{3}N^{\frac{1}{3}}(ln(N))^{\frac{2}{3}}})$ operaciones para poder factorizar un número entero de N bits, y como vemos este tiempo varía de forma exponencial respecto al tamaño en bits de N.

A continuación se muestra como trabaja el algoritmo de Shor. Sea $N = n_1 n_2$ con $(n_1, n_2) = 1$, el número que será factorizado, x un número seleccionado aleatoriamente primo relativo con N, es decir $(x, N) = 1$ y expn la función de exponenciación modular con periodo r,

$$\text{expn}(k, N) = x^k \mod N,\ \text{expn}(k + r, N) = \text{expn}(k, N),\ x^r \equiv 1 \mod N$$

El periodo r es el orden de $x \mod N$. Si r es par, se puede definir $y = x^{\frac{r}{2}}$, que satisface la condición $y^2 \equiv 1 \mod N$ y por lo tanto es solución de uno de los siguientes sistemas de ecuaciones:

$$\begin{aligned} y_1 &\equiv \quad 1 \mod n_1 \quad 1 \mod n_2 \\ y_2 &\equiv -1 \mod n_1 \quad -1 \mod n_2 \\ y_3 &\equiv \quad 1 \mod n_1 \quad -1 \mod n_2 \\ y_4 &\equiv -1 \mod n_1 \quad 1 \mod n_2 \end{aligned}$$

Los dos primeros sistemas tienen las soluciones triviales $y_1 = 1$ e $y_2 = -1$. Los últimos dos sistemas tienen soluciones no triviales $y_3 = a$ e $y_4 = -a$, por el Teorema Chino del residuo, el cual dice que un sistema de k congruencias simultáneas (es decir un sistema de ecuaciones de la forma $y \equiv a_i \mod m_i$) con módulos coprimos entre sí a pares tiene una única solución y con $0 \leq y < m_1 m_2 \cdots m_k$

Si r es par e $y = \pm a$ con $a \neq 1$ y $a \neq N-1$, entonces $a+1$ o $a-1$ debe tener un divisor común con N porque $a^2 \equiv 1 \mod N$, es decir $a^2 = cN + 1$ con $c \in \mathbb{N}$ y por lo tanto $a^2 - 1 = (a+1)(a-1) = cN$. Un factor de N puede ser encontrado por el algoritmo de Euclides para determinar $(N, a+1)$ y $(N, a-1)$, el cual está definido como:

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

Se puede mostrar que un x aleatorio coincide con las condiciones mencionadas con probabilidad $p > \frac{1}{2}$ si N no es de la forma $N = p^\alpha$ o $N = 2p^\alpha$. Puesto que hay algoritmos clásicos eficientes para factorizar potencias puras de primos, se puede encontrar un algoritmo probabilístico eficiente de factorización si el periodo r de la exponenciación modular puede ser determinado en un tiempo polinomial.

Sea F una función cuántica $F : |x, 0\rangle \rightarrow |x, f(x)\rangle$ de la función entera $f : \mathbb{Z} \rightarrow \mathbb{Z}_{2^n}$ con periodo desconocido $r < 2^n$.

Para determinar r, se necesitan dos registros, con los tamaños $2n$ y m qubits, los cuales deben ser inicializados a $|0, 0\rangle$.

Como primer paso se debe producir una superposición homogénea de todos los vectores base en el primer registro aplicando un operador U con

$$U|0,0\rangle = \sum_{i=0}^{N-1} c_i |i, 0\rangle \; con\, |c_i| = \frac{1}{\sqrt{N}} \, y \, N = 2^{2n}$$

Esta superposición puede ser alcanzada por ejemplo con la transformación de Hadamard H. Aplicando F a los estados resultantes tenemos

$$|\psi\rangle = FH|0,0\rangle = F\frac{1}{2^n}\sum_{i=0}^{N-1}|i,0\rangle = \frac{1}{2^n}\sum_{i=0}^{N-1}|i,f(i)\rangle$$

Una medición del segundo registro con resultado $k = f(s)$ con $s < r$ reduce el estado a

$$|\psi'\rangle = \sum_{j=0}^{\frac{N}{r}-1} c'_j|rj+s,k\rangle \; con \; c'_j = \left\lceil \frac{N}{r} \right\rceil^{-\frac{1}{2}}$$

La medición del estado $|\psi'\rangle$ del primer registro consiste únicamente de vectores base de la forma $|rj+s\rangle$ puesto que $f(rj+s) = f(s)$ para toda j, entonces se tiene un espectro homogéneo.

No es posible extraer directamente el periodo r o un múltiplo de él midiendo el primer registro por el desfase aleatorio s. Este problema es resuelto mediante el uso de una transformación discreta de Fourier sobre el registro

$$TDF : |x\rangle \mapsto \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{\frac{2\pi i}{N}xy}|y\rangle$$

como la probabilidad del espectro del estado transformado es invariante al desfase (es decir solo la fase pero no el valor absoluto de la amplitud compleja son afectados)

$$|\psi'\rangle = TDF|\psi'\rangle = \sum_{i=0}^{N-1}\bar{c}'_i|i,k\rangle$$

$$\bar{c}'_i = \frac{\sqrt{r}}{N}\sum_{j=0}^{p-1}\text{expn}\left(\frac{2\pi i}{N}i(jr+s)\right) = \frac{\sqrt{r}}{N}e^{\phi_i}\sum_{j=0}^{p-1}\text{expn}\left(\frac{2\pi i}{N}ijr\right)$$

$$\text{con } \phi_i = 2\pi i\frac{is}{N} \text{ y } p = \lceil\tfrac{N}{r}\rceil$$

Si $N = 2^n$ es un múltiplo de r entonces $\bar{c}'_i = \frac{e^{\phi_i}}{\sqrt{r}}$ si i es un múltiplo de $\frac{N}{r}$ y 0 de otra forma. Pero aún si r no es una potencia de 2, el espectro de $|\bar{\psi}'\rangle$ muestra distintos picos con un periodo de $\frac{N}{r}$ porque

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} e^{2\pi i k\alpha} = \begin{cases} 1 & si\ \alpha \in \mathbb{Z} \\ 0 & si\ \alpha \notin \mathbb{Z} \end{cases}$$

Esta es también la razón del porque se usa un primer registro con $2n$ qubits cuando $r < 2^n$ porque garantiza al menos 2^n elementos en la suma anterior y así un pico de orden $O(1)$

Si se mide ahora el primer registro, se tendrá un valor c cercano a $\lambda N/r$ con $\lambda \in \mathbb{Z}_r$. Esto puede ser escrito como $c/N = c \cdot 2^{-2n} \approxeq \lambda/r$. Se puede pensar esto como encontrar una aproximación racional a/b con $a, b < 2^n$ para el número binario fijo $c \cdot 2^{-2n}$. Un algoritmo clásico eficiente para resolver este problema se puede realizar usando fracciones continuas.

Puesto que la forma de un número racional no es única, λ y r son solo determinados por $a/b = \lambda/r$ si $(\lambda, r) = 1$. La probabilidad de que λ y r sean coprimos es más grande que $1/ln(r)$, así sólo $O(n)$ intentos son necesarios para una probabilidad constante de éxito, tan cerca de 1 como se desee.

Consecuencias: Este algoritmo es el que dio paso a la investigación del cómputo cuántico, pues hasta antes de que se formulara, no se creía que el cómputo cuántico tuviera alguna ventaja sobre los sistemas de computo clásicos, es importante también porque puede en teoría ser usado para romper el esquema de cifrado de clave pública conocido como RSA. RSA está basado en la creencia de que factorizar enteros grandes es imposible computacionalmente. Hasta donde se conoce esta creencia es válida para computadoras clásicas; no se ha implementado ningún algoritmo clásico que pueda factorizar en tiempo polinomial. El algoritmo de Shor muestra que factorizar enteros grandes se puede hacer de manera eficiente en computadoras cuánticas, así que una computadora cuántica lo suficientemente grande puede romper el esquema RSA. También el algoritmo fue un motivante poderoso para el diseño y construcción de computadoras cuánticas y para el estudio de nuevos algoritmos para cómputo cuántico.

En el 2001, un grupo en IBM (International Bussines Machines) mostró el algoritmo de Shor que factorizó 15 en 3×5, usando una implementación NMR (Resonancia Magnética Nuclear) de una computadora cuántica con 7 qubits [VSS$^+$01].

También es considerado un algoritmo de velocidad exponencial pues reduce el tiempo de ejecución para la resolución de un problema de ser subexponencial a ser polinomial.

5.4. Grover

El algoritmo de búsqueda cuántica realiza una búsqueda genérica para solucionar una gran cantidad de problemas [Gro96]. Considérese cualquier problema donde puede reconocer una buena solución y desea buscar de entre una lista de posibles soluciones la mejor solución. Por ejemplo, dado un entero grande N, se puede decir si p es un factor no trivial de N, y así una estrategia simple para encontrar factores no triviales de N es buscar en el conjunto $\{2,\ 3,\ ...\ \lfloor N \rfloor\}$ hasta que se encuentre un factor. Usualmente el mejor algoritmo de búsqueda clásico conocido hace un uso limitado de la estructura del problema, quizá para evitar candidatos que son imposibles de manera obvia o para dar prioridad a algunos candidatos, pero la complejidad total de la búsqueda es todavía exponencial.

La búsqueda cuántica es una herramienta para agilizar este tipo de búsquedas genéricas a través de un espacio de posibles soluciones.

Es útil notar que si se tiene una forma de reconocer una solución a un problema y conociendo las posibles soluciones, en algún sentido se 'sabe' la solución. Sin embargo, no necesariamente se puede producir eficientemente una solución.

Se le da a este problema una estructura matemática más general como sigue. Se asume que las soluciones se pueden expresar como cadenas binarias de longitud n. Defínase una función $f : \{0,1\}^n \to \{0,1\}$ tal que $f(x) = 1$ si x es la codificación binaria de la solución al problema de búsqueda y $f(x) = 0$ de otra forma.

Entonces se puede pensar este problema con una caja negra U_f para calcular una función desconocida $f : \{0,1\}^n \to \{0,1\}$ y se desea encontrar una entrada x tal que $x \in \{0,1\}^n$ tal que $f(x) = 1$.

Por conveniencia se restringe la atención a funciones con exactamente una solución $x = w$. Se asume que se desea que el procedimiento encuentre la solución con probabilidad al menos de p con $\frac{1}{2} < p \leq 1$ para cada función f.

Si sólo se puede hacer una pregunta, lo mejor que puede hacer el algoritmo es proponer una solución x_1 aleatoria uniformemente, y luego usar la pregunta para checar si $f(x_1) = 1$. Si x_1 es la respuesta correcta, regresar x_1, de otra forma, proponer una cadena x_2 aleatoria uniformemente del conjunto $\{0,1\}^n - \{x_1\}$ y regresar x_2. Nótese que este procedimiento regresa el valor correcto $x = w$ con probabilidad $\frac{2}{2^n}$.

Si se tienen dos preguntas, lo mejor que se puede hacer es continuar con el procedimiento anterior y se usa la segunda pregunta para probar si $f(x_2) = 1$. Si $f(x_2) = 1$, regresar x_2 y de otra forma, se propone una cadena x_3 aleatoria uniformemente de $\{0,1\}^n - \{x_1, x_2\}$ y regresar x_3. Este procedimiento regresa $x = w$ con probabilidad $\frac{3}{2^n}$.

Si se continúa con el proceso anterior, con k preguntas, para $k < 2^n$, el procedimiento regresará el valor correcto $x = w$ con probabilidad $\frac{k+1}{2^n}$. Nótese que se puede proponer la respuesta correcta con probabilidad $\frac{1}{2^n}$ sin ninguna pregunta y cada pregunta adicional aumenta la probabilidad de regresar la respuesta correcta en $\frac{1}{2^n}$.

Considere una versión cuántica del algoritmo simple que haga una propuesta sin hacer ninguna pregunta. El procedimiento anterior propone la respuesta correcta con una probabilidad de $\frac{1}{2^n}$, y así la versión cuántica lo hace con una amplitud de probabilidad de $\frac{1}{\sqrt{2^n}}$ si hubiera una forma de incrementar la amplitud en $\frac{1}{\sqrt{2^n}}$ después de cada pregunta, entonces se podría resolver el problema de búsqueda con sólo $O(\sqrt{2^n})$ preguntas. Encontrar tal algoritmo cuántico no es tan inmediato puesto que se está restringido por las leyes de la mecánica cuántica; por lo tanto no se es capaz de utilizar herramientas como la clonación de estados, Grover ideó un algoritmo cuántico que alcanza este aumento de amplitud.

El algoritmo de Grover realiza la búsqueda cuadráticamente más rápido de lo que se puede hacer con cómputo clásico. Si existe exactamente una solución, una búsqueda de fuerza bruta clásica toma $2^n - 1$ preguntas en el peor caso. De hecho cualquier algoritmo clásico que para cualquier función f encuentra una solución con probabilidad de al menos p, $\frac{1}{2} < p \leq 1$, debe hacer $\Omega(2^n)$ preguntas en el peor caso. El algoritmo de búsqueda cuántica de Grover toma solo $O(\sqrt{2^n}) = O(2^{\frac{n}{2}})$ preguntas.

Aunque este aumento no es tan dramático como la ventaja cuántica exponencial que tiene el algoritmo de Shor para factorizar, la enorme aplicabilidad del problema de búsqueda de Grover hacer a este algoritmo interesante e importante. En particular el algoritmo de Grover da una velocidad cuadrática en la solución de problemas NP-completos, los cuales son muchos de los problemas difíciles e importantes en la ciencia de la computación.

Ahora se expone el algoritmo de Grover, se asume que se tiene forma de reconocer una solución, y por lo tanto, se puede asumir que se tiene una caja negra U_f para f como sigue

$$U_f : |x\rangle|b\rangle \mapsto |x\rangle|b \oplus f(x)\rangle$$

Suponga que se coloca el registro blanco $|b\rangle$ el cual consiste de un qubit a $|0\rangle$. Entonces, dado un valor de una pregunta x codificada en el registro de pregunta como $|x\rangle$, supóngase que se pregunta a U_f. El resultado es:

$$|x\rangle|0\rangle \xrightarrow{U_f} |x\rangle|f(x)\rangle$$

y midiendo el qubit blanco, se obtiene la respuesta a la pregunta para f. Pero esto no es mejor que aplicar la pregunta clásica, así que para tomar 'ventaja cuántica', se necesita el uso de superposiciones cuánticas.

Se puede preparar fácilmente el primer registro en una superposición de todos los valores posibles de pregunta,

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \quad (donde\, N = 2^n)$$

Se puede separar la suma anterior en dos partes. La primera parte es una suma sobre todos los x para los cuales $f(x) = 0$; esto es los x 'malos' que no son soluciones para el problema de búsqueda. Sea X_{malo} el conjunto de tales x. La segunda parte es una suma sobre todos los x tales que $f(x) = 1$; esto es los x 'buenos' que son soluciones al problema de búsqueda. Sea X_{buenos} el conjunto de tales x. Por conveniencia, se asume que existe sólo una solución w, así $X_{buenos} = \{x\}$.

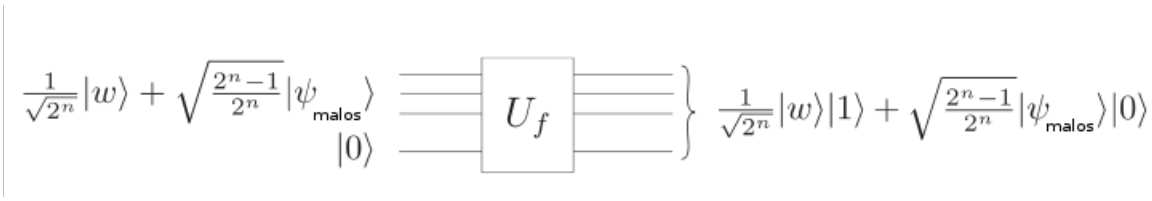

Figura 5.3: El oráculo U_f para la búsqueda cuántica

Se definen los estados

$$|\psi_{buenos}\rangle = |w\rangle$$
$$|\psi_{malos}\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \in X_{malos}} |x\rangle$$

Supóngase que se prepara el qubit blanco de U_f en el estado $|0\rangle$, y el registro pregunta en superposición de la forma:

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle = \frac{1}{\sqrt{N}}|w\rangle + \sqrt{\frac{N-1}{N}}|\psi_{malos}\rangle$$

Ahora con probabilidad $\frac{1}{N}$ una medición del qubit blanco dará $|1\rangle$, y los qubits pregunta se dejaran en el estado bueno $|w\rangle$. Aunque este procedimiento usa el principio de superposición cuántica, no hace ningún uso de la interferencia cuántica y puede ser fácilmente simulado usando aleatoriedad clásica. Este procedimiento es equivalente a muestrear una entrada x aleatoria uniformemente y calcular $f(x)$.

El algoritmo de búsqueda cuántica es un procedimiento iterativo que usa interferencia cuántica para aumentar la amplitud del estado bueno $|w\rangle$ antes de medir el registro pregunta.

Si se coloca el registro pregunta en algún índice de pregunta $|x\rangle$ y se coloca el qubit blanco en $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ el efecto del oráculo es:

$$|x\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} (-1)^{f(x)}|x\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Puesto que el segundo estado es un eigenestado, se puede ignorar, considerando sólo el efecto sobre el primer registro.

$$U_f : |x\rangle \mapsto (-1)^{f(x)}|x\rangle$$

Así el efecto es codificar la respuesta a la pregunta del oráculo en un cambio de fase. Es conveniente redefinir U_f como el operador de n-qubits que realiza la transformación

$$U_f : |x\rangle \to (-1)^{f(x)}|x\rangle$$

También se define un operador de cambio de fase de n-qubits $U_{0^\perp}$ que actúa como sigue:

$$U_{0^\perp} : \begin{cases} |x\rangle \mapsto -|x\rangle,\ x \neq 0 \\ |0\rangle \mapsto\ \ |0\rangle \end{cases}$$

Este operador aplica un cambio de fase de -1 a todos los estados de n-qubits ortogonales al estado $|00..,0\rangle$. Si se denota el espacio vectorial generado por el estado básico $|0\rangle$ por V_0, entonces el espacio ortogonal a V_0 es el espacio generado por todos los estados básicos $|x\rangle \neq |00..,0\rangle$ y puede ser denotado por $V_0^{\perp}$. El operador $U_{0^{\perp}}$ aplica un cambio de fase a los vectores en $V_0^{\perp}$.

Ahora se puede definir el operador que hace el trabajo de incrementar la amplitud de $|\psi_{buenos}\rangle = |w\rangle$. Este operador $G = HU_{0^{\perp}}HU_f$ es llamado el iterado de Grover o el iterado de la búsqueda cuántica. Está definido por la siguiente sucesión de transformaciones.

- Aplicar el oráculo U_f.
- Aplicar la puerta de Hadamard de n-qubits H.
- Aplicar $U_{0^{\perp}}$
- Aplicar la puerta de Hadamard de n-qubits H.

A continuación se muestra un circuito para implementar el iterado de Grover.

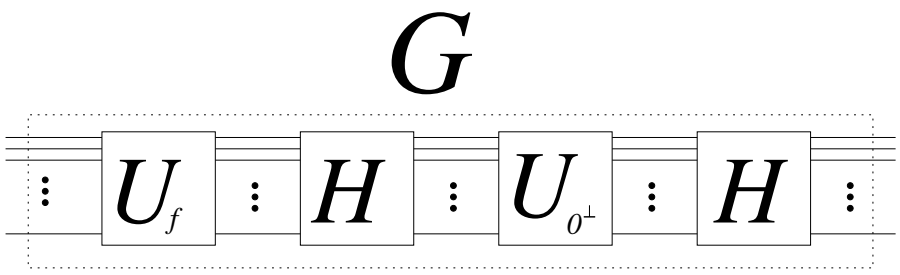

Figura 5.4: El iterado de Grover

Nótese que el qubit blanco para el oráculo U_f se omite en la figura, puesto que se está trabajando con la definición simplificada de U_f.

Ahora que se ha definido el iterado de Grover, el algoritmo de búsqueda cuántica de Grover puede ser escrito como sigue:

- Iniciar con el estado de n-qubits $|00..,0\rangle$.

- Aplicar la puerta de Hadamard de n-qubits H para preparar el estado $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$ (donde $N = 2^n$).
- Aplicar el iterado de Grover un total de $\lfloor \frac{\pi}{4} \frac{1}{\sqrt{N}} \rfloor$ veces.
- Medir el estado resultante.

Consecuencias: Este algoritmo no conlleva un golpe fuerte a la criptografía actual, pues las primitivas que se verían afectadas son las funciones Hash, pero puesto que no reduce de manera exponencial el tiempo de búsqueda, estas funciones siguen siendo seguras, y puesto que está demostrado que este es el mejor algoritmo, pues las funciones Hash siguen siendo de alguna forma seguras. Las funciones hash tienen una caducidad natural debido al avance del poder de cómputo, en ese sentido, el algoritmo de Grover no afecta dramáticamente la seguridad de estas funciones, pero sí hace que se requiera aumentar los parámetros de seguridad de tales funciones.

5.5. Regev

La actual búsqueda de algoritmos cuánticos se concentra en problemas que no se sabe que sean NP-difíciles [Reg03]. Estos incluyen problemas de retículos e isomorfismos de grafos. Se está interesado en problemas de retículos o específicamente el problema del único vector más corto (SVP). Un retículo es el conjunto de todas las combinaciones lineales enteras de un conjunto de n vectores linealmente independientes en $\mathbb{R}^n$. Este conjunto de n vectores es conocido como una base del retículo. En el SVP el objetivo es encontrar el vector más corto no cero en un retículo. En el $f(n)$-único-SVP se da la 'promesa' que el vector más corto lo es por un factor de al menos $f(n)$ de todos los otros vectores no paralelos.

Un problema en cómputo cuántico es el problema del subgrupo escondido (HSP). Aquí, se da una caja negra que calcula una función sobre los elementos de un grupo G. La función se sabe que es constante y distinta sobre clases izquierdas de un subgrupo $H \leq G$ y el objetivo es encontrar H.

Se centrará la atención en el HSP sobre el grupo dihédrico. El grupo dihédrico de orden $2N$, denotado por D_N, es el grupo de simetrías de un polígono regular de N lados. Es isomorfo al grupo generado por el elemento ρ de orden n y el elemento τ de orden 2, sujetos a la relación $\rho\tau = \tau\rho^{-1}$. Además el grupo dihédrico tiene una

estructura mucho más simple que el grupo simétrico, no se conoce alguna solución eficiente al HSP sobre el grupo dihédrico.

Teorema 5.1 *Si existe una solución al problema de clase dihédrica con parámetro de error* $\boldsymbol{f}$*, entonces existe un algoritmo cuántico que resuelva el* $\theta(n^{\frac{1}{2}+2f})$*-único-SVP.*

La entrada para el problema de clase dihédrica (DCP, Dihedrical Coset Problem) es un producto tensorial de un número polinomial de registros. Cada registro está en el estado $|0\rangle|x\rangle + |1\rangle|x + d \mod N\rangle$ para algunos $x \in \{0, 1, ..., N-1\}$ y d es el mismo para todos los registros. Esto puede ser pensado como clases del subgrupo $\{(0,0),(1,d)\}$ en D_N. El objetivo es encontrar el valor de d. Además se dice que el DCP tiene un parámetro de error f, si cada uno de los registros está en el estado $|b\rangle|x\rangle$ con probabilidad a lo más $\frac{1}{(\log N)^{\mathbf{f}}}$, con b, x arbitrarios, en lugar de un estado de la clase. Se nota que cualquier algoritmo que resuelva el HSP dihédrico por muestreo de clases también resuelve el DCP para algún parámetro de error **f**. La razón es que puesto que el algoritmo compara sólo un número polinomial de clases se puede tomar **f** lo suficientemente grande tal que con alta probabilidad todos los registros son estados de clases.

Teorema 5.2 *Si existe una solución al HSP dihédrico que muestrea clases (e. g., cualquier solución usando el 'método estándar') entonces existe un algoritmo cuántico que resuelve poli(n)-único-SVP*

Corolario 5.3 *Si existe un algoritmo S que resuelva* $\frac{1}{poli(\log N)}$ *de las entradas válidas del subconjunto suma con parámetro N, entonces existe una solución al HSP dihédrico*

Corolario 5.4 *Si existe algún algoritmo que resuelva* $\frac{1}{poli(\log N)}$ *de las entradas válidas para la suma de subconjuntos con parámetro N entonces existe un algoritmo cuántico para el* $\Theta(n^{2,5})$*-único-SVP*

El problema de dos puntos

Definición 5.1 *La entrada del problema de dos puntos con parámetro de error* $\boldsymbol{f}$ *consiste de poli(*$n \log M$*) registros. Cada registro está en el estado* $\frac{1}{\sqrt{2}}(|0, \bar{a}\rangle + |1, \bar{a'}\rangle)$ *con probabilidad al menos* $1 - \frac{1}{(n \log(2M)))^f}$ *sobre* $1 + n\lceil \log M \rceil$ *qubits donde* $\bar{a}, \bar{a'} \in$

{0,1,...,M-1}^{n} son arbitrarios tales que $\bar{a'} - \bar{a}$ es fijo. De otra forma su estado es $|b, \bar{a}\rangle$ con probabilidad a lo más $\frac{1}{(n\log(2M))^f}$ donde $b \in \{0,1\}$ y $\bar{a} \in \{0,...,M\text{-}1\}^n$ son arbitrarios. Se dice que un algoritmo resuelve el problema de dos puntos si se obtiene $\bar{a'} - \bar{a}$ con probabilidad poli $\left(\frac{1}{n\log M}\right)$ y tiempo poli$(n\log M)$.

Lema 5.5 *Si existe un algoritmo que resuelva el DCP con parámetro de error $\mathtt{f}$ entonces existe un algoritmo que resuelve el problema de dos puntos con parámetro de error $\mathtt{f}$*

Lema 5.6 *Considere la representación del vector más corto $\bar{u}$ en la base del retículo LLL-reducida $\bar{u} = \sum_{i=1}^{n} u_i \bar{b_i}$. Entonces , $|u_i| \leq 2^{2n}$ para $i \in [n]$.*

Sea $p > n^{2+2\mathtt{f}}$ un primo fijo. El siguiente lema es el lema principal del algoritmo de Regev.

Lema 5.7 *Para cualquier $\mathtt{f}$> 0 sea $\bar{u} = \sum_{i=1}^{n} u_i \bar{b_i}$ el vector más corto del retículo en un retículo ($c_{unq} n^{1+2\mathtt{f}}$)-único, donde c_{unq} >0 es una constante. Si existe una solución al problema de los dos puntos con parámetro de error $\mathtt{f}$ entonces existe un algoritmo cuántico que dado este retículo y tres enteros l,m,i_0 regresa $(u_1, ..., u_{i_0-1}, \frac{u_{i_0}-m}{p}, u_{i_o+1}, ..., u_n)$ con probabilidad 1/poli(n) si las siguientes condiciones se cumplen: $\|\bar{u}\| \leq l \leq 2\|\bar{u}\|$, $u_{i_0} \equiv m \pmod{p}$ y $1 \leq m \leq p-1$.*

Se demuestra que este lema implica el Teorema 5.1 con $\Theta(n^{1+2\mathtt{f}})$ describiendo el algoritmo SVP. De acuerdo con los lemas anteriores, existe una solución al problema de los dos puntos con parámetro de error f. Es decir que existe un algoritmo que dados los valores correctos de l, m, i_0 se obtiene $(u_1, ..., u_{i_0-1}, \frac{u_{i_0}-m}{p}, u_{i_o+1}, ..., u_n)$. El valor l es un estimado de la longitud del vector más corto u. Puesto que el algoritmo LLL da una $2^{(n-1)/2}$ aproximación a la longitud del vector más corto, uno de los $(n-1)/2$ valores distintos de l es el que se requiere. Además, puesto que u es el vector más corto, u/p no puede ser un vector del retículo y por lo tanto existe un i_0 tal que $u_{i_0} \not\equiv 0 \pmod{p}$. Aquí, hay solo $O(pn^2)$ posibles valores para l, m, i_0. Con cada uno de estos valores el algoritmo para SVP llama al algoritmo del lema 5.7 una cantidad polinomial de veces. Con alta probabilidad en uno de esos llamados el algoritmo devuelve el vector ya descrito de donde u puede ser extraído. Los resultados de las otras llamadas pueden ser fácilmente descartados porque son o vectores más largos o vectores que no están en el retículo.

Consecuencias: Con este algoritmo, toda la parte criptográfica basada en la dificultad del problema del vector más corto para retículos queda vulnerable a cualquier computadora cuántica corriendo el algoritmo de Regev. Un ejemplo de este tipo de esquemas es el NTRU(N-th degree truncated polynomial ring) [JHS98], el cual no se sabe si es seguro demostrable, pero al estar basado en problemas de retículos, es vulnerable al algoritmo de Regev.

Capítulo 6

Criptografía Cuántica

En este capítulo se ve lo que es la criptografía cuántica, lo que es un esquema de cifrado cuántico de clave pública y el protocolo de intercambio de clave cuántico.

Como ya se sabe, algunas primitivas actuales de seguridad están comprometidas, tal es el caso del esquema RSA y basados en curvas elípticas [Sho94] [PZ03] por lo que si se lograra realizar una computadora cuántica, practicamente todos los sistemas criptográficos se verían comprometidos, por lo que se necesitan nuevas medidas de protección, es decir otras primitivas de seguridad, esta vez utilizando las ventajas del cómputo cuántico. Tatsuaki Okamoto, Keisuke Tanaka y Shigenori Uchiyama presentan una solución a este problema en [TOU00].

Para dar la solución, primero se muestra una extensión natural del concepto de sistemas criptográficos de clave pública para el modelo de la máquina de Turing cuántica, el sistema criptográfico cuántico de clave pública. Donde se utilizan canales clásicos. Después se muestra un esquema de cifrado cuántico de clave pública, donde la seguridad del esquema se basa en la suposición computacional (sobre máquinas de Turing cuánticas probabilísticas) que una clase de problemas de suma de subconjuntos (cuya densidad es al menos 1) no es soluble en contra de adversarios con máquinas de Turing cuánticas. En este esquema el mecanismo cuántico subyacente es solo el algoritmo de Shor para logaritmos discretos, el cual es usado en la generación de claves. El cifrado y descifrado requieren solamente mecanismos clásicos y así son muy eficientes.

6.1. Esquema de cifrado cuántico de clave pública

Se define lo que es un sistema criptográfico de clave pública y las nociones relacionadas. En la siguiente definición $P[\cdots]$ denota la probabilidad y *Adv* denotal el adversario.

Definición 6.1 *Una función f es llamada cuántica en un sólo sentido si las siguientes condiciones se cumplen:*
1. Existe una máquina de Turing cuántica de tiempo polinomial A, tal que sobre una entrada x, A produzca f(x) (i.e. A(x)=f(x))
2. Para cada máquina de Turing cuántica probabilística de tiempo polinomial, Adv, cada polinomio poly, y n lo suficientemente grande,

$$P[Adv(f(x)) \in f^{-1}(f(x))] < 1/poly(n)$$

La probabilidad es tomada sobre la distribución clásica de x.

Nótese que todas las variables en esta definición son cadenas clásicas y ningún canal cuántico entre pares de componentes se supone.

Definición 6.2 *Un esquema de cifrado cuántico de clave pública consiste de tres máquinas de Turing cuánticas probabilísticas de tiempo polinomial (G,E,D), como sigue:*
1.- G es una máquina de Turing cuántica probabilística de tiempo polinomial para generar claves. Esto es G, sobre una entrada 1^n, produce (e,d) con enorme probabilidad en n, donde e es una clave pública, d es la clave secreta, y n es un parámetro de seguridad.

2.- E es una función de cifrado que produce texto cifrado c, y D es una función de descifrado. Para cada mensaje m de tamaño $|m| = n$, cada polinomio poly y n lo suficientemente grande:

$$P(D[E(m,e),d)=m > 1-1/poly(n)$$

donde nuevamente la probabilidad es clásica.

Definición 6.3 *Un esquema de firma digital cuántico consiste de tres máquinas de Turing cuánticas probabilísticas de tiempo polinomial, (G,S,V), como sigue:*
1.-G genera claves. Esto es G con entrada 1^n, obtiene (s,v), donde s es una clave de firma (secreta), v es una clave de verificación (pública), y n es un parámetro de seguridad.

2.-S es una función de firma que produce la firma σ, y V es una función de verificación. Para cada mensaje m de tamaño $|m| = n$, cada polinomio poly, y n suficientemente grande,

$$P[(V(m,S(m,s),v)=1]>1\text{-}1/poly(n)$$

Ahora se dan dos teoremas que sirven para generar el esquema.

Aquí K es un campo algebraico, O_k es el anillo de enteros de K y $\mathtt{N}(I)$ la norma de I.

Teorema 6.1 *Si K es un campo algebraico y p es un ideal primo de O_k, entonces O_k/p es un campo finito, $\mathbb{F}_{p^f}$, y $\mathtt{N}(p)=p^f$. Existe una base entera $[w_1, ..., w_l]$ tal que cada clase de residuos de O_k/p está representado de manera única por*

$$a_1w_1 + ... + a_lw_l \quad ,$$

donde l es el grado de K, $0 \leq a_i < e_i$ (i=1,...,l) y $[e_1w_1, ..., e_lw_l]$ es una base entera de p. Note que $\prod_{I=1}^{l} e_i=p^f$

Teorema 6.2 *(**Teorema pequeño de Fermat**) Sea p un ideal primo de O_k. y un elemento no cero de O_k/p. Entonces se tiene*

$$g^{N(p)-1} \equiv 1 \ (\mod p)$$

El esquema propuesto se compone de la siguiente manera.

Generación de la clave.

1. Fije un conjunto $\mathbb{K}$ de campos algebraicos, disponibles para el sistema.
2. Escoja al azar un campo algebraico, K de $\mathbb{K}$. Sea O_K su anillo de enteros
3. Fije los parámetros de tamaño n, k de $\mathbb{Z}$
4. Escoja un ideal primo $\wp$ de O_K, y seleccione aleatoriamente un elemento, g, de O_K tal que g es un generador del grupo multiplicativo del campo finito $O_K/\wp$. Aquí un elemento en $O_K/\wp$ está representado de manera única por una base $\{1, w_2, ..., w_l\}$ y una tupla de enteros $(e_1, e_2, ..., e_l)$ (donde e_1=p). Esto es, para cualquier $x \in O_K$, existen enteros $x_1, x_2, .., x_l \in \mathbb{Z}$ ($0 \leq x_i \leq e_i$) tales que $x \equiv x_1 + x_2w_2 + ... + x_lw_l \ (\mod \wp)$.

5. Escoja n enteros $p_1, p_2, ..., p_n$ de $O_K/\wp$ con la condición de que $\mathtt{N}(p_1), ..., \mathtt{N}(p_n)$ son coprimos, y para cualquier subconjunto $\{p_{i_1}, p_{i_2}, ..., p_{i_k}\}$ de $\{p_1, p_2, ..., p_n\}$, existen enteros $a_1, a_2, ..., a_l$ $(0 \leq a_i < e_i)$ tales que $j = 1\ p_{i_j} = a_1 + a_2 w_2 + ... + a_l w_l$

6. Use el algoritmo de Shor para encontrar logaritmos discretos para obtener $a_1, a_2, ..., a_n$ tales que $p_i \equiv g^{a_i} \pmod{\wp}$, donde $a_i \in \mathbb{Z}\ /\ (\mathtt{N}(\wp) - 1)\mathbb{Z}$, y $1 \leq i \leq n$.

7. Escoja un entero racional aleatorio, d, en $\mathbb{Z}/(\mathtt{N}(\wp) - 1)\mathbb{Z}$.

8. Calcule $b_i = (a_i + d) \mod (\mathtt{N}(\wp) - 1)$ para cada $1 \leq i \leq n$.

9. La clave pública es ($\mathbb{K}$, n, b_1, b_2, . . ., b_n), y la clave privada es ($\mathbb{K}$, g, d, p, p_1, p_2, . . ., p_n).

Cifrado

1. Fije la longitud del texto llano M a $\left\lfloor log \begin{pmatrix} n \\ k \end{pmatrix} \right\rfloor$

2. Cifre M en una cadena binaria $m = (m_1, m_2, ..., m_n)$ de longitud n y de peso Hamming k (i.e., que tengan exactamente k 1's) como sigue:

   - Póngase $l \leftarrow k$.
   - Para i de 1 a n haga lo siguiente:
     Si $M \geq \begin{pmatrix} n-i \\ l \end{pmatrix}$ entonces póngase $m_i \leftarrow 1, M \leftarrow M - \begin{pmatrix} n-i \\ l \end{pmatrix}$, $l \leftarrow l - 1$. De otra forma,
     ponga $m_i \leftarrow 0$ (Note que $\begin{pmatrix} l \\ 0 \end{pmatrix} = 1$ para $l \geq 0$, y $\begin{pmatrix} 0 \\ l \end{pmatrix} = 0$ para $l \geq 1$)

3. Calcule el texto cifrado c por

$$c = \sum_{i=1}^{n} m_i b_i$$

Descifrado

1. Calcule $r = (c - kd) \mod (\mathtt{N}(\wp) - 1)$
2. Calcule $u \equiv g^r \mod \wp$
3. Encuentre m como sigue: Si $p_i|u$, entonces ponga $m_i \leftarrow 1$. De otra forma ponga $m_i \leftarrow 0$. Después de completar este procedimiento para todos los p_i's $(1 \leq i \leq n)$, ponga $m = (m_1, m_2, ..., m_n)$
4. Descifre m al texto llano M como sigue:
   - Ponga $M \leftarrow 0$, $l \leftarrow k$
   - Para i desde 1 hasta n haga lo siguiente: Si $m_i = 1$, entonces ponga $M \leftarrow M + \begin{pmatrix} n-i \\ l \end{pmatrix}$ y $l \leftarrow l - 1$

6.2. Protocolo de intercambio de clave cuántico

Uno de los protocolos importantes y el primero en darse a luz es el protocolo **BB84** el cual es un esquema de distribución de clave cuántico desarrollado por Charles Bennet y Gilles Brassard en 1984. El protocolo es seguro demostrable, confiando en la propiedad cuántica de que obtener la información es solo posible perturbando la señal si los estados que se está tratando de distinguir no son ortogonales (teorema de no clonación). Es explicado usualmente como un método seguro de comunicar una clave privada de una parte a otra.

Protocolo BB84

En el esquema BB84 [NC00], Alice desea enviar una clave privada a Bob. Ella empieza con dos cadenas de bits, a y b, cada una de longitud n. Ella entonces cifra estas dos cadenas como una cadena de n qubits,

$$|\psi\rangle = \bigotimes_{i=1}^{n} |\psi_{a_i b_i}$$

a_i y b_i son los i-ésimos bits de a y b respectivamente. Juntos, $a_i b_i$ nos da un índice en los siguientes cuatro estados de qubits:

$|\psi_{00}\rangle = |0\rangle$
$|\psi_{10}\rangle = |1\rangle$

$|\psi_{01}\rangle = |+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$
$|\psi_{11}\rangle = |-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$

Note que el bit b_i es el que decide en que base se codifica a_i. Los qubits están ahora en estados que no son mutuamente ortogonales, y así es imposible distinguir todos ellos con veracidad sin conocer b.

Alice envía $|\psi\rangle$ sobre un canal cuántico público a Bob. Bob recibe un estado $\epsilon\rho = \epsilon|\psi\rangle\langle\psi|$, donde ϵ representa los efectos del ruido en el canal, así como la intercepción por una tercera parte que llamaremos Eve. Después de que Bob recibe la cadena de qubits, las tres partes Alice, Bob y Eve, tienen sus propios estados.

Como sea, puesto que sólo Alice conoce b es virtualmente imposible que Bob o Eve distingan los estados de los qubits. También, después de que Bob haya recibido los qubits, se sabe que Eve no puede tener una copia de los qubits enviados a Bob, aunque ella haya hecho mediciones. Sus mediciones, como sea, corren el riesgo de perturbar un qubit particular con probabilidad $\frac{1}{2}$ si ella adivina la base incorrecta.

Bob procede a generar una cadena de bits aleatorios b' de la misma longitud que b, y entonces mide la cadena que ha recibido de Alice a'. En este punto, Bob anuncia públicamente que ha recibido la transmisión de Alice. Alice sabe entonces que puede anunciar b. Bob comunica sobre un canal público con Alice para determinar que b_i y b'_i no son iguales. Ambos Alice y Bob ahora descartan los qubits a y a' donde b yb' no coinciden.

Para los k bits restantes donde ambos Bob y Alice midieron en la misma base, Alice escoge aleatoriamente $k/2$ bits y revela sus elecciones sobre un canal público. Ambos Alice y Bob anuncian estos bits de manera pública y hacen un chequeo para ver si más que un cierto número de ellos coinciden. Si el chequeo pasa, Alice y Bob proceden a usar un ajuste de información y técnicas de amplificación privadas para crear alguna cantidad de claves secretas compartidas. De otra forma, ellos cancelan todo y empiezan de nuevo.

Protocolo B92

El protocolo BB84 puede ser generalizado para usar otros estados y bases, y se mantienen las conclusiones. De hecho, un protocolo simple existe en el cual solo dos estados son usados. Por simplicidad, es suficiente considerar que le pasa a un bit simple; la descripción generaliza fácilmente a pruebas de bloques así como se hace en BB84.

Suponga que Alice prepara un bit aleatorio clásico a, y, dependiendo del resultado envía a Bob

$$|\psi\rangle = \begin{cases} |0\rangle & si\ a = 0 \\ \frac{|0\rangle+|1\rangle}{\sqrt{2}} & si\ a = 1 \end{cases}$$

Dependiendo de un bit clásico aleatorio a' el cual el genera, Bob mide subsecuentemente el qubit que recibe de Alice en cualquiera de las bases, ya sea Z $|0\rangle$, $|1\rangle$ (si a'=0), o en la base X $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$. De su medición, el obtiene el resultado b, el cual es 0 o 1, correspondiendo a los eigenestados -1 y +1 de X y Z. Bob entonces anuncia públicamente b (pero mantiene en secreto a'), y Alice y Bob llevan una discusión pública manteniendo en secreto sólo los pares $\{a, a'\}$ para el cual $b = 1$. Note que cuando $a = a'$, entonces $b = 0$ siempre. Solo si $a' = 1 - a$ Bob podrá obtener $b = 1$, y eso ocurre con probabilidad 1/2. La clave final es a para Alice y 1-a' para Bob.

Este protocolo, conocido como B92[NC00], remarca como la imposibilidad de la distinción perfecta entre estados no ortogonales cae en el corazón de la criptografía cuántica. Como en el protocolo BB84, puesto que es imposible para cualquier interceptor distinguir entre los estados de Alice sin romper la correlación entre los bits que Alice y Bob guardan, este protocolo permite a Alice y Bob crear bits clave compartidos mientras que colocan también una cota superior del ruido e intercepción durante su comunicación. Ellos pueden entonces aplicar reconciliación de información y amplificación privada para extraer los bits secretos de sus cadenas de bits aleatorias correlacionadas.

El protocolo EPR

Los bits clave generados en los protocolos BB84 y B92 son originados por Alice. Como sea, se ve que la clave puede generarse de un proceso aleatorio fundamentalmente que envuelve las propiedades del entrelazados de qubits. Esto se ilustra por el protocolo EPR.

Suponga que Alice y Bob comparten un conjunto de n pares entrelazados de qubits en el estado

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Estos estados son conocidos como pares EPR. Obtener estos estados se puede hacer de diferentes formas, por ejemplo, Alice pudo haber preparado los pares y entonces mandar la mitad de cada uno a Bob y vice versa. Una tercera entidad pudo preparar los pares y mandar mitades a Alice y Bob, o ellos pudieron haberse conocido mucho tiempo atrás y compartido los pares, guardándolos hasta este momento. Alice y Bob entonces seleccionan un subconjunto aleatorio de los pares EPR, y prueban para ver si violan la desigualdad de Bell o alguna otra prueba de fidelidad. Pasando la prueba certifican que continúan teniendo estados cuánticos entrelazados suficientemente puros, colocando una cota inferior sobre la fidelidad de los pares restantes EPR (y así cualquier ruido o intervención). Y cuando ellos miden estos de manera conjunta determinan bases aleatorias, Alice y Bob obtienen cadenas de bits clásicos correlacionados de los cuales pueden obtener bits de clave secreta como en B92 y BB84.

Puesto que el protocolo es simétrico, no se puede decir si Alice o Bob generaron la clave. Mejor dicho, la clave es aleatoria de verdad. De hecho, lo mismo se aplica al protocolo BB84, puesto que puede ser reducido a una instancia de la versión generalizada del protocolo EPR. Suponga que Alice prepara un bit b aleatorio clásico, y de acuerdo a eso, mide su mitad del par EPR en ya sea la base $|0\rangle|1\rangle$ o $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$, obteniendo a. Supóngase que Bob hace lo mismo, midiendo en una base b' (escogida aleatoriamente) y obtiene a'. Ahora ellos se comunican b y b' sobre un canal público clásico, y guardan como sus claves sólo aquellos $\{a, a'\}$ para los cuales $b = b'$ Note que esta clave es indeterminada hasta que Alice y Bob realizan una medición en sus mitades de pares EPR. Por esta razón, la criptografía cuántica es algunas veces vista no como una forma secreta de cambiar claves o transferirlas, sino más bien como una forma de generar claves secretas, puesto que fundamentalmente ni Alice ni Bob pueden predeterminar la clave hasta terminar el protocolo.

Capítulo 7

Conclusiones

Aún cuando los componentes electrónicos se minimizan de tamaño, se tiene un límite, un momento en el cual los componentes electrónicos no se comportan como los objetos que observamos cotidianamente, sino que las leyes de la mecánica cuántica entran en juego, estas leyes de la mecánica cuántica a veces van en contra del sentido común.

Como se ha visto, el cómputo cuántico presenta una seria amenaza a la criptografía clásica y una ventaja sobre el cómputo clásico, hay que entender que no es una ventaja que haga que los problemas resueltos por un modelo de cómputo se incrementen, sino el hecho de que problemas de complejidad exponencial se ven reducidos a problemas de complejidad polinomial, situación que no se había logrado con cómputo clásico.

Dentro de la realización física de una computadora cuántica, apenas se tienen algunos esbozos, algunos modelos que se aproximan; pero se han encontrado con problemas tales como la medición del resultado, la interacción con el medio, etc. Algunas técnicas que han funcionado con una cantidad pequeña de qubits son NMR (Nuclear Magnetic Resonance) que fue donde se probó el algoritmo de Shor con 7 qubits y se factorizó el número 15 en 3×5, otra técnica es la de QED (Quantum Electro Dynamics) Cavities donde se ha probado el algoritmo de Grover con 3 qubits.

Respecto a computadoras cuánticas lo suficientemente grandes, no se ha encontrado la técnica que permita construir alguna con k-qubits y k lo suficientemente grande como para poder vulnerar esquemas de cifrado, basados en RSA, curvas elípticas y NTRU.

Aún así, supongamos que se tiene una computadora cuántica lo suficientemente grande, entonces, la mayoría de los sistemas criptográficos quedarían vulnerables, por

ejemplo RSA quedaría completamente vulnerable, la seguridad basada en funciones Hash tendría que incrementar sus parámetros de seguridad para evitar quedar completamente vulnerable, el esquema NTRU quedaría completamente vulnerable al igual que cualquier esquema basado en la dificultad de hallar el vector más corto en un retículo.

Entonces **los sistemas criptográficos actuales usados comúnmente quedan vulnerables al cómputo cuántico.**

El estudio de los algoritmos cuánticos y sus consecuencias nos lleva a decir que cuando aparezcan las computadoras cuánticas, los esquemas de cifrado y toda la criptografía actual (clásica) quedará vulnerable, por lo que será necesario rediseñar o en su defecto reemplazar los esquemas de cifrado para soportar ataques de computadoras cuánticas, al igual que los protocolos de intercambio de llaves, lo mismo que las firmas digitales.

Aún quedan incógnitas en este campo, por ejemplo dentro de las clases de complejidad, el problema más grande es encontrar algoritmos que resuelvan problemas en la clase BQP pero que no estén en la clase BPP, es decir, problemas que se pueden resolver sólo en computadoras cuánticas, otro problema enorme es la realización física de una computadora cuántica.

Bibliografía

[AER35] B. Podolsky A. Einstein and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 47:4, 1935.

[Bea03] Stéphane Beauregard. Circuit for shor's algorithm using $2n + 3$ qubits. *arXiv:quant-ph/0205095*, February 2003.

[BHT88] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In A. V. Moura C. L. Lucchesi, editor, *LATIN'98, LNCS*, volume 1380, pages 163–169. Springer-Verlag Berlin Heidelberg, 1988.

[Bon99] Dan Boneh. Twenty years of attacks on the rsa cryptosystem. *In Notices of the American Mathematical Society*, 1(2):203–213, 1999.

[Coh96] Henri Cohen. *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics. Springer-Verlag, 1996.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. In *IEEE transactions on Information Theory*, volume 22 of *6*, pages 644–654, November 1976.

[DJ92] David Deutsch and Richard Jozsa. Rapid solutions of problems by quantum computation. In *he Royal Society of London*, volume 439, page 553, 1992.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman and company, 1979.

[GMR88] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. In *Siam Journal on Computing*, pages 281–308, 1988.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pages 212–219, 1996.

[Hug97] Richard J. Hughes. Cryptography, quantum computation and trapped ions. *http://arxiv.org/abs/quant-ph/9712054*, November 1997.

[JHS98] J. Pipher J. Hoffstein and J. H. Silverman. Ntru: a ring based public key cryptosystem. *Lecture Notes in Computer Science*, June 1998.

[Knu97] Donald E. Knuth. *The Art of Computer Programming*, volume 2. Addison-Wesley, 1997.

[LA99] Steve Lloyd and Carlisle Adams. *Understanding the Public-Key Infrastructure: Concepts, Standards, and Deployment Considerations*. Sams, 1st edition, November 1999.

[LV01] A.K. Lenstra and E.R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology*, 14(4):255–293, 2001.

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[Neu55] John Von Neuman. *Mathematical Foundations of Quantum Mechanics*. Princeton University Press, 1955.

[NY89] M.Ñaor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual Symposium on the Theory of Computing*. ACM, 1989.

[PKM07] Raymond Laflamme Phillip Kaye and Michele Mosca. *An Introduction to Quantum Computing*. Oxford, 2007.

[PZ03] John Proos and Christof Zalka. Shor's discrete logarithm quantum algorithm for elliptic curves. Technical report, arXiv.org, 2003. http://arxiv.org/abs/quant-ph/0301141v2.

[Reg03] O. Regev. New lattice based cryptographic constructions. In *Proc. 35th ACM Symp. on Theory of Computing (STOC)*, pages 407–416, 2003.

[RS04] P. Rogaway and T. Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal Roy and Willi Meier, editors, *Lecture Notes in Computer Science*, volume 3017, pages 371 – 388. Springer-Verlag Heidelberg, 2004.

[RSA78] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. In *Communications of the ACM*, volume 21 of *2*, pages 120–126, 1978.

[Sho94] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings, 35th Annual Symposium n Fundamentals of Comp. Science (FOCS)*, pages 124–134, 1994.

[Sip97] Michael Sipser. *Introduction to the theory of computation*. PWS publishing company, 1997.

[TOU00] Keisuke Tanaka Tatsuaki Okamoto and Shigenori Uchiyama. Quantum public-key cryptosystems. *CRYPTO 2000 : advances in cryptology*, 1880:147–165, 2000.

[VSS+00] M. K. Vandersypen, M. Steffen, M. H. Sherwood, C. S. Yannoni, G. Breyta, and I. L. Chuang. Implementation of a three-quantum-bit search algorithm. *Applied Physics Letters*, 76(5), 2000.

[VSS+01] M. K. Vandersypen, M. Steffen, M. H. Sherwood, C. S. Yannoni, G. Breyta, and I. L. Chuang. Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883–887, 2001.