# UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

***FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y MECÁNICA***

***ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS***


---

## DESCIFRAR RSA CON COMPUTACIÓN CUÁNTICA

---

**Asignatura:** Computación Cuántica

**Docente:** HANS HARLEY CCACYAHUILLCA BEJAR


| **Integrantes:** | **Código** |
|---|---|
| • CARDENAS HUAMAN FABRICIO YARED | 192998 |
| • ESPIRILLA MACHACA JOSEPH ODE | 145004 |
| • HUAMAN QUISPE ANDY MARCELO | 192422 |
| • QUISPE CLEMENTE SAMAN | 150495 |
| • VALENCIA ÑAUPA MARKO LEONEL | 191874 |


**Cusco - Perú**

**2024 - I**

**ÍNDICE:**

## 1. RESUMEN/ABSTRACT

Este trabajo explora la implementación del algoritmo de Shor para la factorización de números enteros utilizando computación cuántica, con el objetivo de descifrar el sistema de encriptación RSA.

El estudio incluye la construcción detallada del circuito cuántico, la aplicación de la QFT, y la simulación del sistema. Los resultados de la simulación se visualizan mediante un histograma, mostrando la distribución de los estados medidos. A partir de estos resultados, se identifican las raíces cuadradas y se calculan los factores primos utilizando el máximo común divisor.

Adicionalmente, se implementa un código clásico para la generación de claves RSA, demostrando el proceso de creación de claves públicas y privadas. Este enfoque permite contrastar los métodos clásicos de encriptación RSA con las capacidades de factorización de la computación cuántica.

## 2.MARCO TEÓRICO

### Criptografía RSA

RSA (Rivest-Shamir-Adleman) es un sistema criptográfico de clave pública que se basa en la dificultad de la factorización de números grandes. La seguridad de RSA depende de que, aunque sea sencillo multiplicar dos números primos grandes, es extremadamente difícil factorizar el producto de estos números sin conocerlos.

- Generación de claves:
    - Se seleccionan dos números primos grandes $p$ y $q$.
    - Se calcula su producto $n = pq$
    - Se elige un exponente de cifrado $e$ que sea coprimo con $(p-1)(q-1)$
    - Se calcula el exponente de descifrado $d$, tal que :
    - $ed \equiv 1 mod (p-1)(q-1)$
    - La clave pública es $(e,n)$ y la clave privada es $(d,n)$
- Cifrado y Descifrado:
    - *Cifrado:*
      $C = Memodn$
    - *Descifrado:*
      $M = Cdmodn$

**Computación Cuántica**
La computación cuántica utiliza principios de la mecánica cuántica, como la superposición y el entrelazamiento, para procesar información de manera más eficiente que las computadoras clásicas. El computador cuántico es una idea que crece a medida que se desarrolla la teoría cuántica de la información. Hoy en día el computador cuántico encuentra enormes dificultades para ser construído en la práctica, hasta el punto que determinados autores opinan que es inviable. Si bien se opta por una postura más optimista, lo cierto es que a medida que vayamos afrontando los problemas que surgen a la hora de crearlo, aprenderemos más y más sobre cómo funcionan los sistemas cuánticos y sobre cómo se comporta la información somos capaces de obtener.

**Qubits:**

Qubits: Unidad Básica de Información Cuántica

- Estados: $|0\rangle$, $|1\rangle$, y superposición de ambos
- Pueden representar múltiples valores simultáneamente
- Permiten procesamiento paralelo masivo
- Pierden superposición al ser medidos
- Tipos: superconductores, iones atrapados, fotones, etc.
- Desafío: mantener coherencia cuántica

**Shor (SHOR'S ALGORITHM)**

El algoritmo de Shor es un algoritmo cuántico que permite factorizar números enteros grandes de manera eficiente, lo cual tiene importantes implicaciones para la criptografía. Fue desarrollado por Peter Shor en 1994 y es uno de los algoritmos cuánticos más conocidos. Mientras que los algoritmos clásicos requieren un tiempo exponencial para factorizar números grandes, el algoritmo de Shor puede hacerlo en tiempo polinómico.

Implicaciones para RSA: El algoritmo de Shor puede romper el RSA al factorizar eficientemente el número nnn, permitiendo así calcular las claves privadas desde las claves públicas. Esto es una amenaza significativa para los sistemas de seguridad actuales que se basan en la dificultad de la factorización.

## 3.INTRODUCCIÓN

La criptografía RSA, pilar fundamental en la seguridad de datos en internet, es uno de los métodos más utilizados para asegurar la comunicación digital. Basada en la complejidad de factorizar grandes números primos, RSA ha sido durante décadas un baluarte de la ciberseguridad. Sin embargo, el avance de la computación cuántica plantea desafíos significativos a su integridad. Esta tecnología emergente promete resolver problemas matemáticos complejos, como la factorización de enteros grandes, de manera exponencialmente más eficiente que los métodos clásicos, poniendo en entredicho la inviolabilidad del RSA.

El presente documento está estructurado para abordar esta problemática de manera exhaustiva. Comienza con un abstract que proporciona un resumen conciso del proyecto y sus objetivos. El marco teórico subsiguiente profundiza en los temas fundamentales, explicando el funcionamiento del algoritmo RSA en el contexto de la computación cuántica, partiendo de los qubits y destacando las características cruciales a considerar.

En la sección de metodología, se describe detalladamente el funcionamiento de RSA y los métodos para descifrarlo. El desarrollo presenta, paso a paso, el algoritmo creado para este proyecto, desglosando minuciosamente los módulos utilizados y los resultados obtenidos. Finalmente, se ofrecen conclusiones sobre cada tema expuesto, seguidas de una bibliografía exhaustiva que respalda la investigación realizada a lo largo del proyecto.

## 4. Metodología

- **Descripción del algoritmo**

  El algoritmo RSA se utiliza para lograr la seguridad en la comunicación mediante el cifrado asimétrico. Este tipo de cifrado emplea dos claves distintas: una clave pública y una clave privada. Sus objetivos principales son:

  a) **Confidencialidad**

  La confidencialidad asegura que solo el destinatario previsto pueda descifrar y leer el mensaje. Cuando una persona desea enviar un mensaje seguro utilizando RSA, cifra el mensaje con la clave pública del destinatario. Solo el destinatario, que posee la clave privada correspondiente, puede descifrar el mensaje. Esto garantiza que, incluso si el mensaje es interceptado durante la transmisión, no podrá ser leído por nadie más que el destinatario legítimo.

  b) **Autenticidad**

  La autenticidad verifica que el mensaje proviene realmente del remitente legítimo. Esto se logra mediante el uso de firmas digitales. El remitente puede firmar un mensaje con su clave privada, y cualquier persona que tenga acceso a la clave pública del remitente puede verificar la firma. Si la verificación es exitosa, se garantiza que el mensaje fue enviado por el remitente y no por un impostor. Esto es crucial en muchas aplicaciones, como en transacciones bancarias y comunicaciones oficiales.

  c) **Integridad**

  La integridad garantiza que el mensaje no ha sido alterado durante la transmisión. En el proceso de firma digital, además de verificar la autenticidad, también se asegura que el mensaje no ha sido modificado. Cualquier cambio en el mensaje después de que ha sido firmado invalidará la firma, lo que alertará al destinatario de que la integridad del mensaje ha sido comprometida.

  RSA logra esto mediante el uso de un par de claves, una pública para el cifrado y una privada para el descifrado.

- **Funcionamiento del Algoritmo RSA**
  - **Generación de Claves**:

    Se generan dos números primos grandes y se utilizan para producir dos claves: una pública y una privada. La clave pública puede ser compartida libremente, mientras que la clave privada debe mantenerse en secreto.

  - **Cifrado**:

Para enviar un mensaje seguro, el remitente cifra el mensaje utilizando la clave pública del destinatario. Este proceso convierte el mensaje de texto en un texto cifrado, que parece ser una secuencia de datos aleatorios.

- **Descifrado**:

  El destinatario utiliza su clave privada para descifrar el texto cifrado y recuperar el mensaje original. La clave privada es necesaria para revertir el cifrado, y sólo el destinatario tiene acceso a ella.

- **Funciones del Algoritmo de Shor para descifrar**

Shor puede descomponer un número entero grande en sus factores primos en tiempo polinómico, algo que es intratable para los algoritmos clásicos.

1. **Preparación del Estado Cuántico:**
   - Se inicia con un estado cuántico que consiste en dos registros. El primer registro se prepara en una superposición uniforme de todos los posibles valores de entrada, y el segundo registro se inicializa en el estado $|0\rangle$.
2. **Transformada de Fourier Cuántica (QFT):**
   - Después de aplicar la función, se realiza una medición del segundo registro, lo que colapsa el estado del sistema. Luego se aplica la QFT al primer registro.
3. **Medición y Obtención del Período:**
   - Tras la aplicación de la QFT, se mide el primer registro. El resultado de la medición está relacionado con el período rrr.
   - Utilizando técnicas de teoría de números, se determina rrr a partir del resultado medido.
4. **Factorización:**
   - Una vez conocido el período rrr, se utilizan relaciones matemáticas para encontrar un divisor no trivial de NNN. Si rrr es par, entonces aplica el maximo comun divisor
   - Si rrr es impar o no proporciona factores útiles, se elige un nuevo aaa y se repite el proceso.

El algoritmo de Shor, por lo tanto, no descifra directamente el mensaje, sino que hace posible recuperar la clave privada utilizada en RSA, lo que permite luego descifrar cualquier mensaje cifrado con la correspondiente clave pública.

## 5.DESARROLLO

**Circuito Cuántico:**

- Sea N el segundo elemento de la tupla de la clave pública
- $T = 2(\log2(N) + 1)$
- Se crea un circuito cuántico con T qubits donde se implementa el módulo “a” con respecto a N.
- Se aplica una compuerta X en los qubits pertinentes.
- Se aplican compuertas de control CX entre varios qubits para establecer el cálculo modular.
- Finalmente, se crean compuertas de control CCX (Toffoli) para los qubits de medicion los cuales son la mitad.
- Se convierte en una compuerta cuántica personalizada y se nombra "a^x mod N".

**Transformada de Fourier Cuántica (QFT):**

- Se crea un circuito cuántico con n qubits donde se implementa la QFT.
- Se aplican compuertas Hadamard H y compuertas de fase controlada CP.
- Se intercambian los qubits (swap) para completar la QFT.
- Se convierte en una compuerta cuántica personalizada y se nombra "QFT" seguido por el número de qubits.

**Construcción del Circuito:**

- Se crea un circuito cuántico con T qubits con la mitad para las mediciones.
- Se aplican compuertas Hadamard H a los primeros T/2 qubits.
- Se añade la compuerta "a^x mod N" al circuito.
- Se miden los qubits T/2 a T en los bits clásicos.
- Se aplica una barrera para separar las operaciones.
- Se añade la compuerta QFT de T/2 qubits.
- Se miden los primeros T/2 qubits en los bits clásicos.
- Se dibuja el circuito.

**Simulación del Circuito:**

- Se selecciona un backend de simulación cuántica.
- Se compila y transpira el circuito para el backend seleccionado.
- Se ejecuta el circuito y se obtienen los resultados.
- Se visualizan los resultados mediante un histograma.

**Verificación:**

- Se identifican las raíces cuadradas (4 en este caso) y se calcula el máximo común divisor (mcd) para obtener los factores primos.
- Los factores primos se calculan usando el mcd de $4-1$ y $4+1$ con respecto a 15, obteniendo 3 y 5.

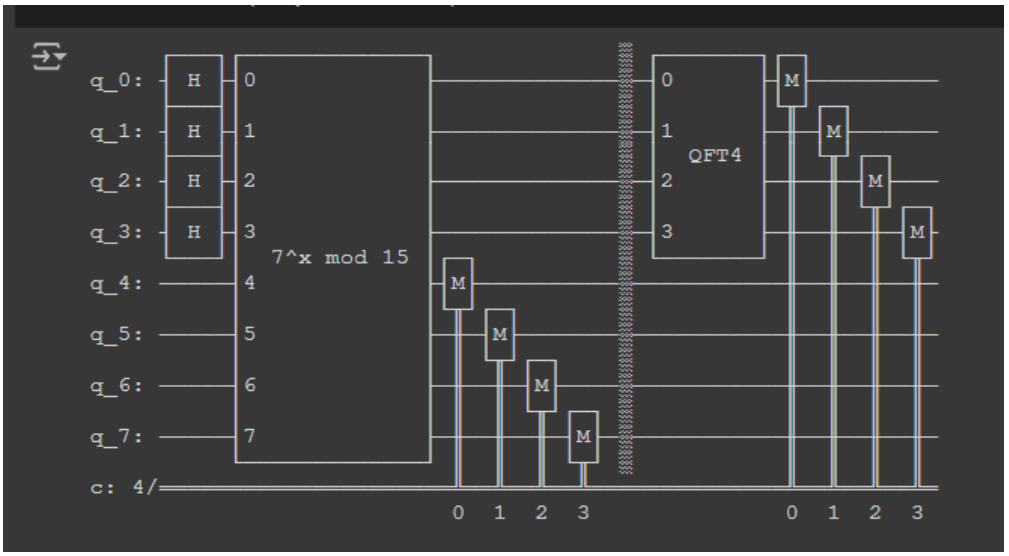
**Código RSA:**

- Se generan números primos aleatorios de un número determinado de bits.
- Se asegura que los números primos p y q sean diferentes.
- Se calculan n y $\phi$ para los números primos.
- Se elige un exponente público común e=65537 y se calcula el inverso modular d.

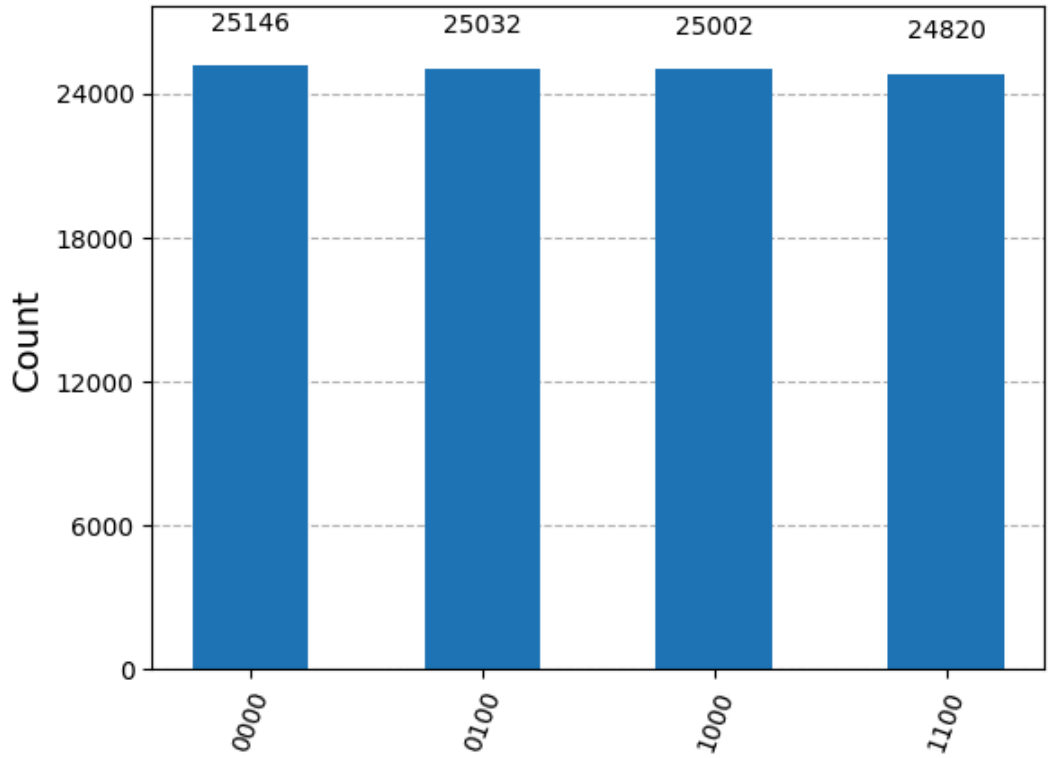
- Se generan y retornan las claves pública y privada.

## 6. RESULTADOS

**Circuito Cuántico:** El circuito cuántico implementa el cálculo de 7^x mod 15 y realiza una QFT para obtener los resultados de medición. Esto se simula en un backend cuántico.




**Simulación:**El histograma muestra los resultados de la simulación del circuito cuántico descrito anteriormente. Cada barra representa la cantidad de veces que se midió un estado específico de los qubits después de ejecutar el circuito.



**Factores Primos:** A partir de la raíz cuadrada 444, se obtienen los factores primos de 15 como 3 y 5 utilizando el mcd.

Verificacion

```
[ ]  1 # con esto sabemos que 4 es raiz cuadrada ya que el mcd de 4 8 12 es 4
     2 # sabiendo esto podemos calcular los factores primos de la siguiente manera:
     3
     4 import math
     5
     6 primer_factor = math.gcd(4-1, 15)
     7 segundo_factor = math.gcd(4+1, 15)
     8 print("factores primos:", primer_factor, segundo_factor)
```

```
factores primos: 3 5
```

**Código RSA:**

```
Clave publica: (65537, 35)
Clave privada: (17, 35)
Ciphertext: 34
Desencriptar mensaje: "
```

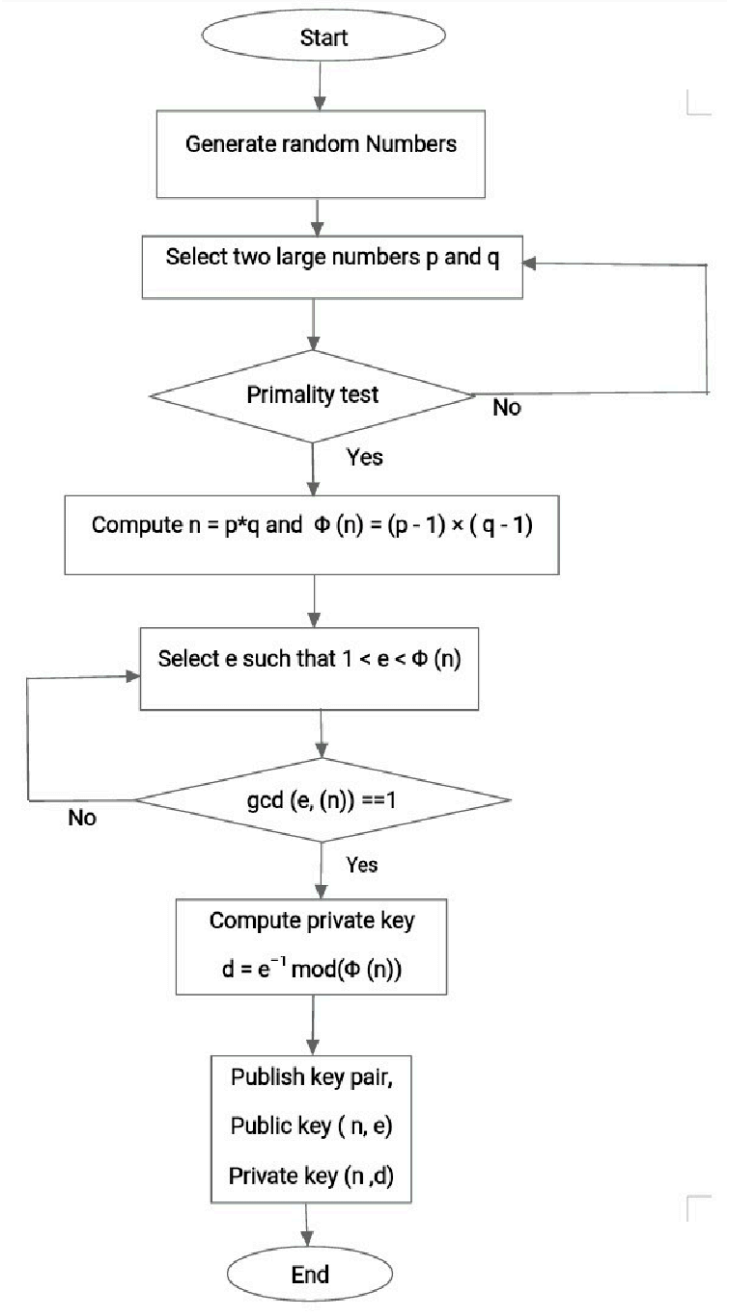
## 7. CONCLUSIONES

1. El circuito cuántico implementado es capaz de calcular 7^x mod 15 y utilizar la QFT para obtener resultados útiles en el algoritmo de Shor, que se puede usar para la factorización de números enteros.
2. La simulación cuántica confirma la precisión del circuito diseñado.
3. Usando el método descrito, se pueden calcular los factores primos de un número compuesto como 15, lo cual es fundamental para el algoritmo de Shor y otros algoritmos cuánticos de factorización.
4. La generación de claves RSA es correcta y el cifrado/descifrado funciona como se espera, demostrando la implementación efectiva del algoritmo RSA.
5. La elección de claves pequeñas en el ejemplo muestra el funcionamiento básico del RSA, aunque en la práctica se usan claves mucho más grandes para mayor seguridad.

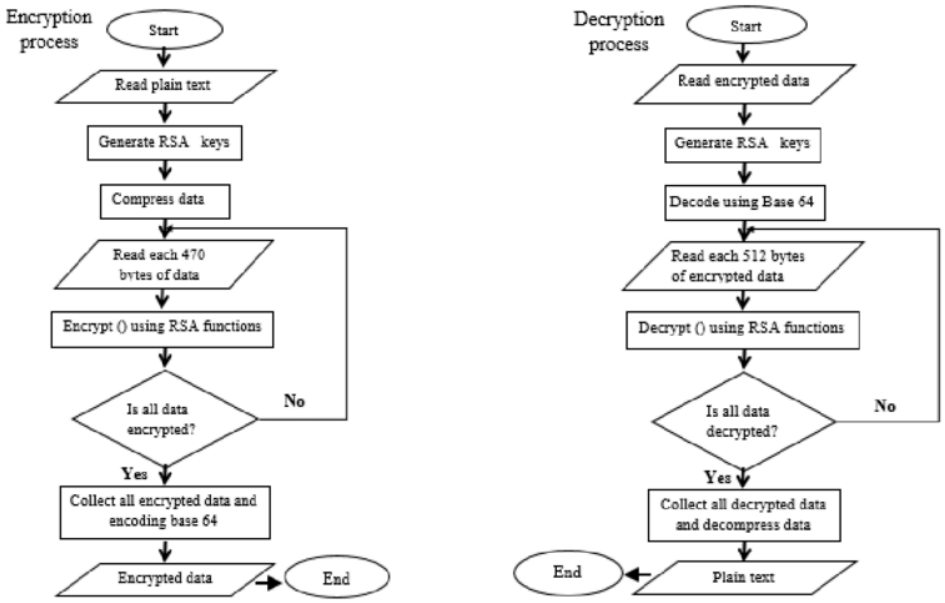
## 8. ANEXOS

### Anexo 1: Diagramas explicativos

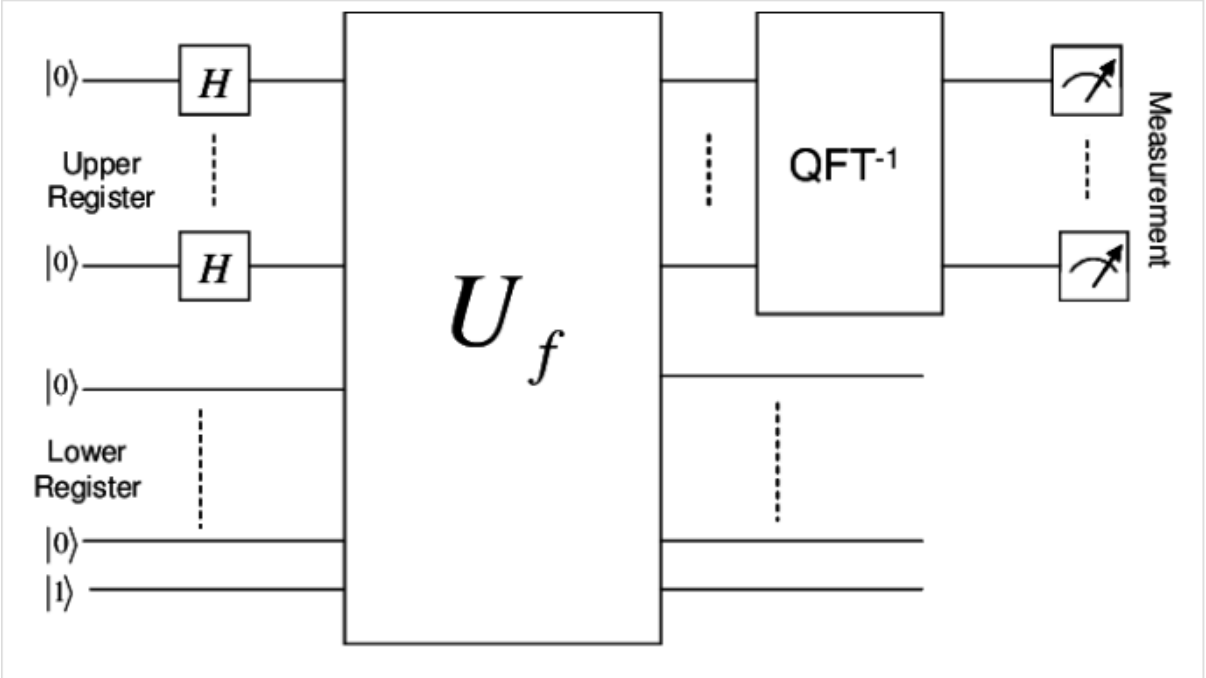
- **Figura A1.1: Proceso de generación de claves RSA**



- **Figura A1.2: Cifrado y descifrado RSA**




- **Figura A1.3: Algoritmo Shor**




**Anexo 2: Tablas informativas**

- **Tabla A2.1: Tiempos de factorización: Computadoras clásicas vs. cuánticas**

| Tamaño de clave RSA | Tiempo estimado (Clásica) | Tiempo estimado (Cuántica) |
|---|---|---|
| 512 bits | 1 semana | 3.5 horas |

| | | |
|---|---|---|
| 1024 bits | 1 año | 10 horas |
| 2048 bits | 300 trillones de años | 8 días |

Nota: Estos son tiempos hipotéticos para ilustrar la diferencia. Los tiempos reales pueden variar.

- **Tabla A2.2: Evolución del tamaño de claves RSA recomendadas**

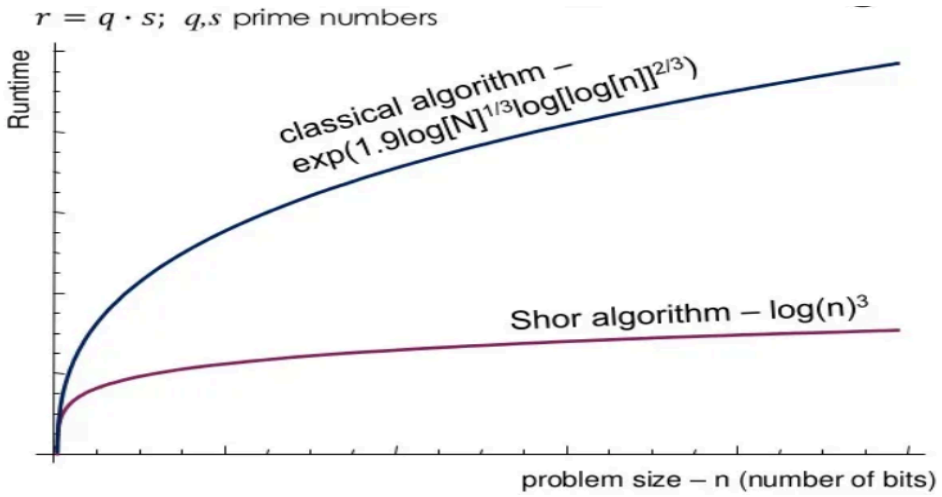
| **Año** | **Tamaño de clave recomendado** |
|---|---|
| **1977** | **512 bits** |
| **1999** | **1024 bits** |
| **2015** | **2048 bits** |
| **2030** | **4096 bits (proyección)** |

- **Tabla A2.3: Shor vs Algoritmo clásico**

| Num dígitos | Alg. Clásico | Alg. De Shor |
|---|---|---|
| 129 | 1.85 años | 45.9 minutos |
| 250 | 2.1 x 10^6 años | 3.4 horas |
| 1000 | 4.5 x 10^25 años | 3.07 días |

**Anexo 3: Gráficos**

- **Figura A3.1: Comparación entre el tiempo que le lleva factorizar un número al mejor algoritmo clásico, comparado con el algoritmo cuántico de Shor**




**Anexo 4: Imágenes**

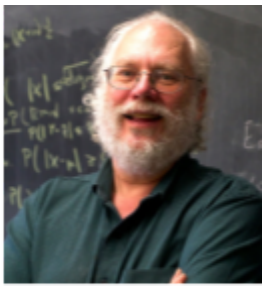
- **Figura A4.1: Computadora cuántica IBM Q System One**


- **Figura A4.2: Pioneros de la criptografía RSA y computación cuántica**

Ron Rivest

Adi Shamir

Leonard Adleman



Peter Shor

**Anexo 6: Glosario de términos**

- **Superposición:** Estado cuántico en el que un qubit puede existir simultáneamente como 0 y 1.
- **Entrelazamiento:** Fenómeno cuántico donde dos partículas están correlacionadas independientemente de la distancia.
- **Transformada de Fourier cuántica:** Operación fundamental en muchos algoritmos cuánticos, incluyendo el algoritmo de Shor.

**Anexo 8: Recursos adicionales**

- **Simulador de circuitos cuánticos IBM Quantum Experience:** https://quantum-computing.ibm.com/
- **Artículo original del algoritmo de Shor:** "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer" (1997)

## 9. REFERENCIAS


1. Okonkwo, Joshua & Ozor, Godwin & Okoye, Francis. (2019). Performance Analysis of RSA Algorithm for Audio Data Security in Communication Networks. 8. 48 - 52.

2. Ghaly, Samir & Abdullah, Mahmood. (2021). Design and implementation of a secured SDN system based on hybrid encrypted algorithms. TELKOMNIKA (Telecommunication Computing Electronics and Control). 19. 1118. 10.12928/telkomnika.v19i4.18721.

3. Hussain, Zahid. (2016). Strengths and Weaknesses of Quantum Computing.. International Journal of Scientific and Engineering Research. 7.

4. Chandel, Sonali & Cao, Wenxuan & Sun, Zijing & Yang, Jiayi & Zhang, Bailu & Ni, Tian-Yi. (2020). A Multi-dimensional Adversary Analysis of RSA and ECC in Blockchain Encryption. 10.1007/978-3-030-12385-7_67.

5. Fairview Capital. (2020). Will Quantum Computing Change the World?. Retrieved Juen 30, 2024, from https://medium.com/fairview-capital/will-quantum-computing-change-the-world-c42e409e21e8 ()

6. Wikipedia contributors. (2024). IBM Q System One. In Wikipedia, The Free Encyclopedia. Retrieved 03:31, July 1, 2024, from https://en.wikipedia.org/w/index.php?title=IBM_Q_System_One&oldid=1198204904

7. P.Q. Nguyen y J. Seifert (2000). RSA Key Generation Revisited

8. M.J.J.G. van den Berg y E.F. Deplazes (2012). Factoring RSA Keys with Known Quotient

9. Richard A.Mollin (2002). RSA and Public-Key Cryptography

10. Encinas H. (2005). El Criptosistema RSA

11. M.Shand, J.Vuillemin (2002). Fast implementations of RSA cryptography

12. Lavanya K. Galla; Venkata SreeKrishna Koganti; Nagarjuna Nuthalapati (2002). Implementation of RSA

13. Katz, J., & Lindell, Y. (2020). Introduction to Modern Cryptography.

14. Stallings, W. (2016). Cryptography and Network Security: Principles and Practice.