Shor's Algorithm

Phichet Phuangrot 6432114821 and Thanakorn Suthamkasem 6430140721

Computer Engineering, Chulalongkorn University

May 18, 2025

Abstract

This report, prepared for the course 2110404 Computational Theory, presents a summary and overview of Shor's algorithm—a quantum algorithm developed for integer factorization. The algorithm demonstrates how quantum computing can solve certain problems exponentially faster than classical approaches, particularly in the context of breaking RSA encryption. This report outlines the mathematical foundations of Shor's algorithm, explains its quantum circuit structure, and discusses its computational significance. The work is presented to Dr. Suthee Ruangwises.

1 Building Blocks of Quantum Computing

Quantum computing is a revolutionary computational paradigm that leverages the principles of quantum mechanics to process information. Unlike classical computers, which use bits (0 or 1), quantum computers use **quantum bits**, or **qubits**, which can represent multiple states simultaneously. This section explores the key components and foundational ideas behind quantum computing.

1.1 Quantum Computing

Quantum computing is an emerging field of computing that harnesses the principles of quantum mechanics to perform calculations. Unlike classical computing which relies on bits that are either 0 or 1, quantum computing utilizes *quantum bits* or *qubits*, which can represent both 0 and 1 simultaneously thanks to the property of *superposition*. This allows quantum computers to process a vast number of possibilities in parallel, offering potentially exponential speedups for specific computational problems such as factoring large integers and searching unsorted databases. Key quantum phenomena enabling this capability include *superposition*, *entanglement*, and *quantum interference*.

1.2 Qubit

A *qubit* (quantum bit) is the fundamental unit of information in quantum computing. While classical bits hold a definite value of either 0 or 1, a qubit can exist in a *superposition* of both states simultaneously. Mathematically, the state of a qubit can be described as:

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

where α and β are complex probability amplitudes that satisfy the normalization condition:

$$|\alpha|^2 + |\beta|^2 = 1.$$

Beyond superposition, qubits exhibit the remarkable property of *entanglement*, where the state of one qubit can become intrinsically linked to the state of another, regardless of the distance separating them. For example, in an entangled pair of qubits, measuring the state of one qubit instantaneously determines the state of the other, a feature that has profound implications for quantum communication and cryptography.

1.3 Superposition

The principle of *superposition* allows a qubit to be in a linear combination of the classical basis states $|0\rangle$ and $|1\rangle$. This can be expressed as:

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

where α and β are complex amplitudes representing the probability amplitudes of the qubit being measured in the corresponding states. The probabilities of measuring the qubit in $|0\rangle$ or $|1\rangle$ are given by $|\alpha|^2$ and $|\beta|^2$, respectively. Since these probabilities must sum to one, the amplitudes obey the normalization condition:

$$|\alpha|^2 + |\beta|^2 = 1.$$

Superposition enables quantum computers to perform computations on many possible inputs simultaneously, providing the foundation for quantum parallelism. This is a critical advantage in algorithms like Shor's factoring algorithm and Grover's search algorithm, where multiple solutions can be explored at once, leading to significant computational speedups compared to classical counterparts.

1.4 Dirac's Notation (Bra-Ket Notation)

Quantum state spaces and the transformations acting on them can be described using vectors and matrices, or in the more compact **bra/ket notation** invented by Dirac.

- **Ket** ($|x\rangle$): Represents a column vector describing a quantum state. For example, the basis states in a two-dimensional space are:
$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}, \quad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$
- **Bra** ($\langle x|$): The conjugate transpose (row vector) of the ket. For example:
$$\langle 0| = \begin{bmatrix}1 & 0\end{bmatrix}, \quad \langle 1| = \begin{bmatrix}0 & 1\end{bmatrix}$$
- **Inner Product** ($\langle x|y\rangle$): Represents the dot product between two states. For instance,
$$\langle 0|0\rangle = 1, \quad \langle 0|1\rangle = 0$$
- **Outer Product** ($|x\rangle\langle y|$): Produces a matrix used to transform quantum states. For example:
$$|0\rangle\langle 1| = \begin{bmatrix}1\\0\end{bmatrix}\begin{bmatrix}0 & 1\end{bmatrix} = \begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}$$
This operator maps:
$$|1\rangle \to |0\rangle\,, \quad |0\rangle \to \begin{bmatrix}0\\0\end{bmatrix}$$

Any linear combination of basis states, such as $a\,|0\rangle + b\,|1\rangle$, can be written as:

$$|\psi\rangle = \begin{bmatrix}a\\b\end{bmatrix}$$

The space of quantum states is modeled by a finite-dimensional complex Hilbert space. All valid quantum states must have unit norm, i.e.:

$$\langle\psi|\psi\rangle = |a|^2 + |b|^2 = 1$$

An important transformation is the **Pauli-X gate**, which swaps $|0\rangle$ and $|1\rangle$. Using bra-ket notation:

$$X = |0\rangle\langle 1| + |1\rangle\langle 0| = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$$

We can also describe its action directly:

$$X : |0\rangle \to |1\rangle\,, \quad |1\rangle \to |0\rangle$$

Bra-ket notation provides a powerful and intuitive way to describe quantum states, their inner relationships, and the operators acting upon them.

1.5 Measurement

Measurement is a fundamental concept in quantum computing, wherein observing a quantum state forces it to collapse into one of the basis states associated with the measuring device. The outcome of a measurement is inherently probabilistic and alters the quantum state.

Single Qubit Measurement: For a single qubit in the state $|\psi\rangle = a|0\rangle + b|1\rangle$ with $|a|^2 + |b|^2 = 1$, measurement in the standard basis $\{|0\rangle, |1\rangle\}$ yields:

- Outcome $|0\rangle$ with probability $|a|^2$ and the state collapses to $|0\rangle$.
- Outcome $|1\rangle$ with probability $|b|^2$ and the state collapses to $|1\rangle$.

Two Qubit Measurement Example: Let a general two-qubit state be:

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$$

with $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$. To measure the **first qubit** in the standard basis, rewrite the state as:

$$|\psi\rangle = |0\rangle \otimes (a|0\rangle + b|1\rangle) + |1\rangle \otimes (c|0\rangle + d|1\rangle)$$

Define:

$$u = \sqrt{|a|^2 + |b|^2}, \quad v = \sqrt{|c|^2 + |d|^2}$$

Then, normalize and express as:

$$|\psi\rangle = u|0\rangle \otimes \left(\frac{a}{u}|0\rangle + \frac{b}{u}|1\rangle\right) + v|1\rangle \otimes \left(\frac{c}{v}|0\rangle + \frac{d}{v}|1\rangle\right)$$

Upon measurement:

- Outcome $|0\rangle$ with probability $|a|^2 + |b|^2 = u^2$, collapsing the state to $|0\rangle \otimes \left(\frac{a}{u}|0\rangle + \frac{b}{u}|1\rangle\right)$
- Outcome $|1\rangle$ with probability $|c|^2 + |d|^2 = v^2$, collapsing the state to $|1\rangle \otimes \left(\frac{c}{v}|0\rangle + \frac{d}{v}|1\rangle\right)$

Multi-Qubit Measurement: For k-qubit measurement on an n-qubit system, there are 2^k possible outcomes. The quantum state is projected into one of 2^k orthogonal subspaces, each corresponding to a measurement result. The measurement device randomly chooses one outcome m_i with probability equal to the sum of squared amplitudes of basis vectors consistent with m_i, and the state collapses accordingly.

Measurement and Entanglement: Measurement provides a useful lens to understand entanglement:

- The state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is **entangled**. Measurement of one qubit instantly affects the state of the other.
- The state $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) = |0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ is **not entangled**. Measurement of one qubit has no influence on the other.

A quantum state is entangled if it cannot be factored into a tensor product of individual qubit states. This entanglement manifests during measurement by introducing dependencies between the outcomes of separate qubits.

1.6 Bloch Sphere

The **Bloch Sphere** is a powerful geometrical representation of the state of a single qubit. Because a qubit state lives in a two-dimensional complex Hilbert space, it can be challenging to visualize its state directly. The Bloch Sphere simplifies this by mapping any pure qubit state to a point on the surface of a unit sphere in three-dimensional real space.

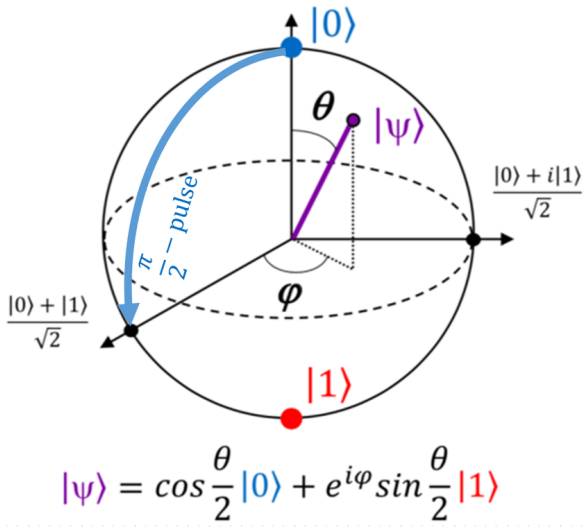

Figure 1: Example illustration of the Bloch Sphere with state $|\psi\rangle$.

Mathematical Representation

A general pure qubit state can be expressed as:

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

where

- $\theta \in [0, \pi]$ is the polar angle,
- $\phi \in [0, 2\pi)$ is the azimuthal angle.

Each pair (θ, ϕ) corresponds to a unique point on the sphere, where:

- The north pole ($\theta = 0$) corresponds to the classical basis state $|0\rangle$.
- The south pole ($\theta = \pi$) corresponds to the classical basis state $|1\rangle$.
- Points along the equator ($\theta = \pi/2$) represent equal superpositions of $|0\rangle$ and $|1\rangle$ with different relative phases ϕ.

Intuition and Significance

- The Bloch Sphere captures the essential aspects of a qubit's quantum state, such as **superposition** and **phase**.
- The angle θ controls the relative probabilities of measuring $|0\rangle$ or $|1\rangle$.
- The angle ϕ encodes the relative phase between these basis states, a uniquely quantum property with no classical analogue.
- This visualization helps understand how quantum gates rotate qubit states on the sphere — for example, the Pauli-X gate flips the qubit state from north to south pole (like a classical NOT), while other gates correspond to rotations about various axes.

2 How Quantum Computers Work

Quantum computers process information using qubits, which follow the laws of quantum mechanics. Unlike classical bits, qubits can exist in superposition and be entangled, enabling parallel computation.

2.1 Quantum Gates

Quantum gates are the building blocks of quantum circuits. Unlike classical logic gates, quantum gates must be **reversible**, meaning their transformations can be undone. This reversibility stems from the requirement that all quantum operations be described by **unitary transformations** — linear transformations that preserve inner products in the complex vector space of quantum states.

Formally, a matrix U is **unitary** if $UU^\dagger = I$, where $U^\dagger$ is the conjugate transpose of U. Unitary transformations preserve orthogonality and probability amplitudes, and correspond to rotations in complex space. This makes quantum gates fundamentally different from classical gates, which may lose information (e.g., the AND gate is not reversible).

Below are examples of single-qubit quantum gates with their matrix representations:

- Identity gate I:
$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
- Pauli-X gate (bit-flip):
$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
- Pauli-Y gate (bit and phase flip):
$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$
- Pauli-Z gate (phase-flip):
$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$
- Hadamard gate H:
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$
This gate is fundamental for creating superpositions:
$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle, \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$
The Hadamard gate transforms classical basis states into equal-probability superpositions, playing a key role in many quantum algorithms such as Grover's and Shor's algorithms.

For multi-qubit systems, gates such as the **Controlled-NOT (CNOT)** gate are used. CNOT operates on two qubits and flips the target qubit if the control qubit is $|1\rangle$:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Graphically, this gate is often represented as a circuit with a control dot and a target $\oplus$ symbol connected by a vertical line.

The Role of Complex Numbers One might ask: why do quantum states need complex numbers if measured probabilities are real and positive?

The answer lies in **quantum interference**. While measurement outcomes depend on the squared magnitudes (which are real and positive), the *evolution* of quantum states via gates is governed by complex amplitudes. Consider:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Both states produce the same measurement probabilities in the computational basis, but they behave differently under certain operations. For example:

$$H\left|+\right\rangle = \left|0\right\rangle, \quad H\left|-\right\rangle = \left|1\right\rangle$$

This behavior shows how interference between positive and negative amplitudes, enabled by complex numbers, allows quantum computers to distinguish and manipulate information in ways that classical systems cannot.

2.2 Quantum Circuits

Quantum circuits are visual representations of quantum computations. Each horizontal wire represents a qubit, and operations (gates) are applied from left to right.

Components of a Circuit

- **Qubit wires:** Horizontal lines representing the evolution of qubit states.
- **Gates:** Symbols (boxes or circles) indicating unitary operations applied to qubits.
- **Measurements:** Depicted typically as a meter symbol or arrow, indicating the final measurement step.

Quantum vs Classical Circuits

- Classical logic circuits operate with bits and use irreversible gates (e.g., AND, OR).
- Quantum circuits manipulate qubits and use reversible gates (unitary transformations).
- Quantum circuits exploit phenomena such as entanglement and interference, allowing algorithms like Shor's (for factoring) or Grover's (for searching) to outperform classical counterparts.

3 Quantum Algorithms in Action

3.1 Phase Kickback

Phase kickback is a fundamental quantum phenomenon that plays a central role in many quantum algorithms, including Quantum Phase Estimation (QPE), the Quantum Fourier Transform (QFT), and ultimately Shor's algorithm for factoring integers [3]. It describes how phase information from a unitary operation applied to one register can be "kicked back" to a control qubit. This effect allows quantum systems to encode global eigenvalue information into relative phases, which can later be extracted through quantum interference.

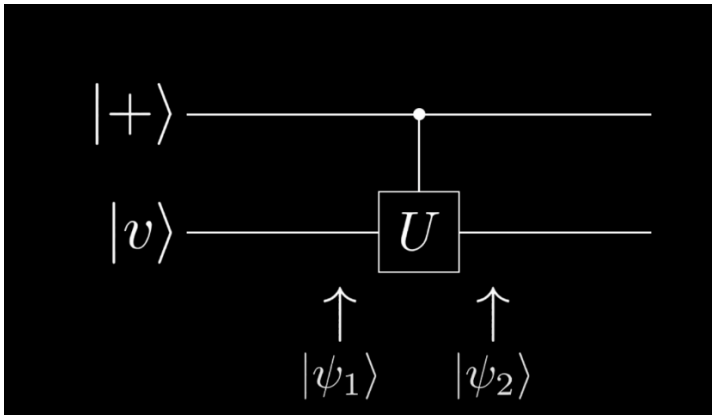

Figure 2: Phase kickback effect: phase information is transferred from the target to the control qubit via a controlled unitary.

Basic Setup

To illustrate the phase kickback effect, consider two quantum systems:

- A control qubit prepared in the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$,
- A target register prepared in an eigenstate $|v\rangle$ of a unitary operator U, such that:

$$U|v\rangle = e^{2\pi i\phi}|v\rangle$$

for some $\phi \in [0, 1)$ [1].

The initial state of the system is:

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

Now apply the controlled-U gate C_U, which applies U to the target register only when the control qubit is in the $|1\rangle$ state:

$$C_U(|0\rangle|v\rangle) = |0\rangle|v\rangle, \quad C_U(|1\rangle|v\rangle) = |1\rangle U|v\rangle = e^{2\pi i\phi}|1\rangle|v\rangle$$

So the resulting state becomes:

$$\begin{aligned}|\psi_2\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle|v\rangle + e^{2\pi i\phi}|1\rangle|v\rangle\right) \\ &= \left(\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\phi}|1\rangle\right)\right) \otimes |v\rangle\end{aligned}$$

The target eigenstate $|v\rangle$ remains unchanged, while the control qubit picks up a relative phase based on the eigenvalue of U. The phase $e^{2\pi i\phi}$ has effectively been "kicked back" to the control qubit.

Leading to Quantum Phase Estimation

This mechanism of encoding phase information into a control register becomes a powerful computational tool when extended to multiple control qubits. By applying a sequence of controlled-U^{2^j} gates, each control qubit accumulates a phase proportional to $2^j\phi$, where ϕ is the unknown eigenphase. This setup is the basis of the *Quantum Phase Estimation* algorithm [1], which uses the *Quantum Fourier Transform* to decode these phases and estimate ϕ with exponential precision.

We explore this next in the following section on Quantum Phase Estimation.

3.2 Quantum Phase Estimation

Quantum Phase Estimation (QPE) is a central algorithm in quantum computing used to estimate the eigenvalue (phase) ϕ of a unitary operator U [1] given an eigenstate $|v\rangle$ such that:

$$U|v\rangle = e^{2\pi i\phi}|v\rangle, \quad \text{where } \phi \in [0, 1)$$

The goal of QPE is to extract the binary expansion of ϕ up to m bits of precision [3], i.e.,

$$\phi = 0.j_1j_2j_3\ldots j_m = \sum_{k=1}^{m}\frac{j_k}{2^k}$$

We begin with a system initialized in the state:

$$|\psi_0\rangle = |0\rangle^{\otimes m} \otimes |v\rangle$$

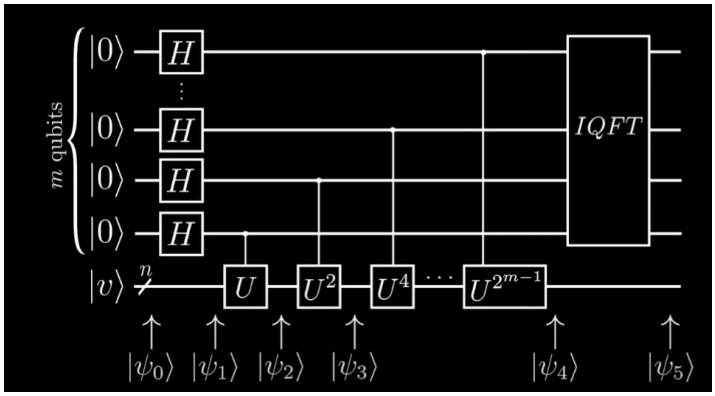

Figure 3: Quantum Phase Estimation Circuit. The top register extracts the binary digits of the phase ϕ.

Step 1: Apply Hadamard Gates to the First Register

We apply Hadamard gates to each of the m control qubits, placing them in an equal superposition:

$$|\psi_1\rangle = \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)^{\otimes m} \otimes |v\rangle$$

Step 2–4: Apply Controlled-U^{2^k} Gates and Use Phase Kickback

Next, we apply a sequence of controlled-U^{2^k} gates (for $k = 0$ to $m-1$) to entangle the phase information with the control register [3] via phase kickback. The state evolves as follows:

$$|\psi_2\rangle = \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)^{\otimes m-1} \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\phi}|1\rangle) \otimes |v\rangle$$
$$|\psi_3\rangle = \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)^{\otimes m-2} \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{4\pi i\phi}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\phi}|1\rangle) \otimes |v\rangle$$
$$|\psi_4\rangle = \bigotimes_{k=0}^{m-1} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 2^k\phi}|1\rangle\right) \otimes |v\rangle$$

Now, we express the phase ϕ in binary:

$$\phi = \sum_{k=1}^{m} \frac{j_k}{2^k}$$

Substituting into the expression, we obtain:

$$|\psi_4\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\frac{j_m}{2}}|1\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(\frac{j_{m-1}}{2}+\frac{j_m}{4})}|1\rangle\right) \otimes \cdots$$
$$\otimes \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(\frac{j_1}{2}+\frac{j_2}{4}+\cdots+\frac{j_m}{2^m})}|1\rangle\right) \otimes |v\rangle$$

Step 5: Apply the Inverse QFT (IQFT)

Finally, we apply the inverse Quantum Fourier Transform (IQFT) to the control register, which extracts the binary digits $j_1 j_2 \dots j_m$ corresponding to the phase ϕ:

$$|\psi_5\rangle = \text{IQFT}\left(|\psi_4\rangle\right) = |j_1 j_2 \dots j_m\rangle \otimes |v\rangle$$

The result is that the control register holds an m-bit approximation of ϕ, and the target register is undisturbed in the eigenstate $|v\rangle$.

Relation to the Quantum Fourier Transform

The final step of QPE—the Inverse Quantum Fourier Transform (IQFT)—is crucial because it transforms the accumulated relative phases in the control register into a computational basis state encoding the binary digits of the phase ϕ. The IQFT effectively "decodes" the frequency information stored via phase kickback into a readable binary representation. This step is only possible due to the mathematical structure provided by the QFT, which connects periodic phase evolution with bitwise representations. Thus, the success of QPE depends fundamentally on the properties of the QFT, making it a core subroutine in quantum algorithms like Shor's algorithm [5].

3.3 Quantum Fourier Transform

The Quantum Fourier Transform (QFT) is the quantum analogue of the classical discrete Fourier transform [3]. It maps quantum basis states into superpositions with amplitudes that encode phase information, and it is essential in many quantum algorithms, particularly Quantum Phase Estimation and Shor's algorithm.

Definition

For an input quantum state $|j\rangle$, where j is an n-bit integer represented as $j = j_0 j_1 \dots j_{n-1}$, the QFT is defined by the unitary transformation [3, 1]:

$$\text{QFT}\,|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j k / 2^n}\,|k\rangle$$

This operation transforms computational basis states into a new basis where phase differences encode frequency information.

Circuit Structure (Based on Figure 4)

Figure 4 illustrates the circuit implementation of the QFT for an n-qubit input $|j\rangle = |j_0 j_1 \dots j_{n-1}\rangle$, where j_0 is the most significant bit. The circuit performs the transformation through a sequence of:

- **Hadamard gates (H)**: Applied to each qubit to create superpositions and enable interference. Each Hadamard gate maps $|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.
- **Controlled phase gates (R_k)**: These gates introduce relative phases based on the values of less significant qubits. For example:

  $$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$$

  Each qubit receives phase contributions from all qubits after it via a series of controlled-R_k gates [3].
- **SWAP gates**: At the end of the circuit, the qubits are reversed using SWAP gates. This step corrects the output ordering, since the QFT produces the result in bit-reversed order.

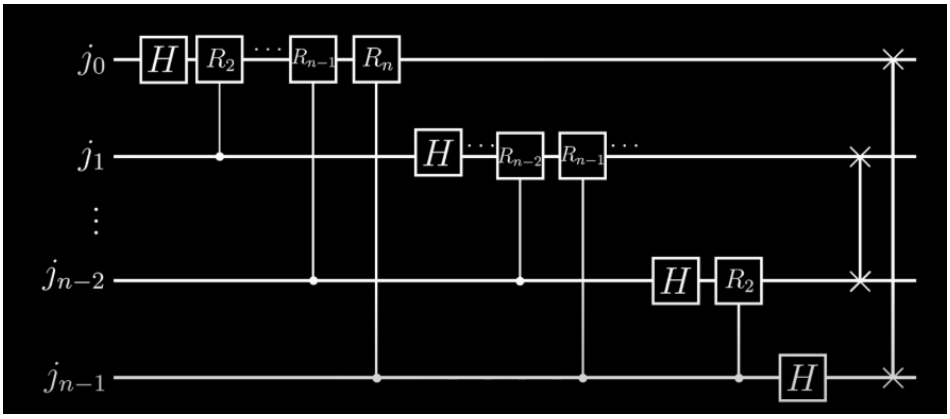

Figure 4: Quantum Fourier Transform circuit. The input state $|j\rangle = |j_0 j_1 \dots j_{n-1}\rangle$ is transformed into a quantum superposition using Hadamard and controlled-R_k gates. SWAP gates at the end reverse the order of qubits.

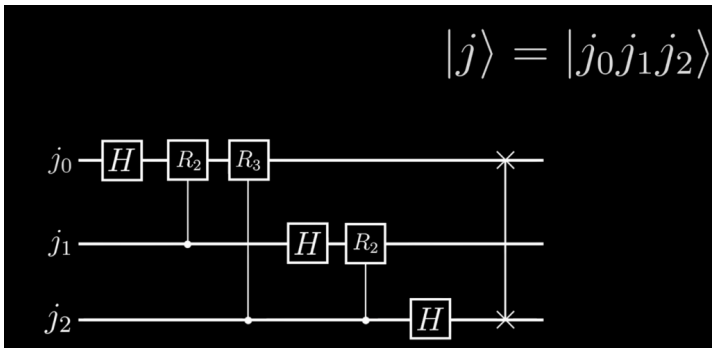

Figure 5: Quantum Fourier Transform on a 3-qubit state

Example: QFT on a 3-Qubit State

We now demonstrate the Quantum Fourier Transform (QFT) on a 3-qubit computational basis input $|j\rangle = |j_0 j_1 j_2\rangle$, where j_0 is the most significant bit. The QFT circuit applies a sequence of Hadamard gates and controlled-R_k phase rotation gates, followed by qubit swaps at the end.

The transformation proceeds as follows:

$$\text{Qubit } |j_0\rangle: \quad \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{j_0}{2}+\frac{j_1}{4}+\frac{j_2}{8}\right)}|1\rangle\right)$$

$$\text{Qubit } |j_1\rangle: \quad \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{j_1}{2}+\frac{j_2}{4}\right)}|1\rangle\right)$$

$$\text{Qubit } |j_2\rangle: \quad \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\frac{j_2}{2}}|1\rangle\right)$$

Finally, the QFT swaps the order of the qubits (e.g., $|j_0 j_1 j_2\rangle \to |j_2 j_1 j_0\rangle$) to complete the transform. This reversal aligns the binary fractional representation of the phase for use in algorithms like Quantum Phase Estimation and Shor's Algorithm.

Connection to Other Algorithms

The QFT plays a central role in both Quantum Phase Estimation (QPE) and Shor's algorithm [3, 5]. In QPE, the QFT is used to extract the phase ϕ from an eigenvalue $e^{2\pi i\phi}$ encoded across the amplitudes of the control register. The inverse QFT (IQFT) is then used to convert this phase-encoded state into

the computational basis to read out ϕ with high probability. In Shor's algorithm, the QFT is used to find the period of a modular function, which is then used to factor large integers.

4 Shor's Algorithm – Breaking Encryption

4.1 RSA Encryption

RSA (Rivest–Shamir–Adleman) encryption [4] is a public-key cryptosystem widely used for secure data transmission. Its security relies on the computational difficulty of factoring large integers, specifically the product of two large primes.

Key Generation

- Choose two large prime numbers p and q.
- Compute the modulus:
$$N = p \cdot q$$
- Compute Euler's totient function:
$$\phi(N) = (p-1)(q-1)$$
- Choose an integer e such that $1 < e < \phi(N)$ and $\gcd(e, \phi(N)) = 1$.
- Compute the modular inverse d such that:
$$d \equiv e^{-1} \mod \phi(N)$$

The public key is (N, e) and the private key is (N, d).

Encryption and Decryption

- To encrypt a message M (with $0 \leq M < N$):
$$C = M^e \mod N$$
- To decrypt:
$$M = C^d \mod N$$

The RSA scheme is secure as long as factoring N is hard. Classical computers can't factor large N efficiently—but Shor's algorithm can [5].

4.2 Number Theory Method

The goal of Shor's algorithm is to factor a composite number $N = p \cdot q$, which breaks the RSA cryptosystem. Instead of brute-force factoring, it reduces the problem to **order finding** [6], which quantum computers can solve efficiently.

Classical Reduction to Period Finding

Let N be a large composite number. Pick a random integer a, such that:

$$1 < a < N \quad \text{and} \quad \gcd(a, N) = 1$$

Define a function:

$$f(x) = a^x \mod N$$

This function is periodic, with period r [6], the smallest integer such that:

$$a^r \equiv 1 \mod N$$

Once we find r, we can factor N (with high probability) using:

$$\text{If } r \text{ is even and } a^{r/2} \not\equiv -1 \mod N, \text{ then:}$$
$$\gcd(a^{r/2} - 1, N) \quad \text{or} \quad \gcd(a^{r/2} + 1, N)$$

gives a non-trivial factor of N.

Example Setup

Suppose $N = 15$ and choose $a = 2$. Then:

$$f(x) = 2^x \mod 15$$

yields the sequence: $2, 4, 8, 1, 2, 4, 8, 1, \ldots$

This function has period $r = 4$, and since:

$$2^{4/2} = 2^2 = 4, \quad \gcd(4 - 1, 15) = 3, \quad \gcd(4 + 1, 15) = 5$$

we have successfully factored 15.

The quantum part of Shor's algorithm is used to find this period r efficiently using Quantum Phase Estimation (QPE), which is powered by the Quantum Fourier Transform (QFT).

4.3 Shor's Algorithm

Shor's algorithm is a quantum algorithm that efficiently factors large composite integers, thereby compromising the widely used RSA cryptographic system. The algorithm splits into two main parts:

- A classical pre- and post-processing step.
- A quantum subroutine for period finding.

The classical part is efficient and well-known. The quantum part provides an exponential speedup over the best-known classical methods, and is the focus of this section.

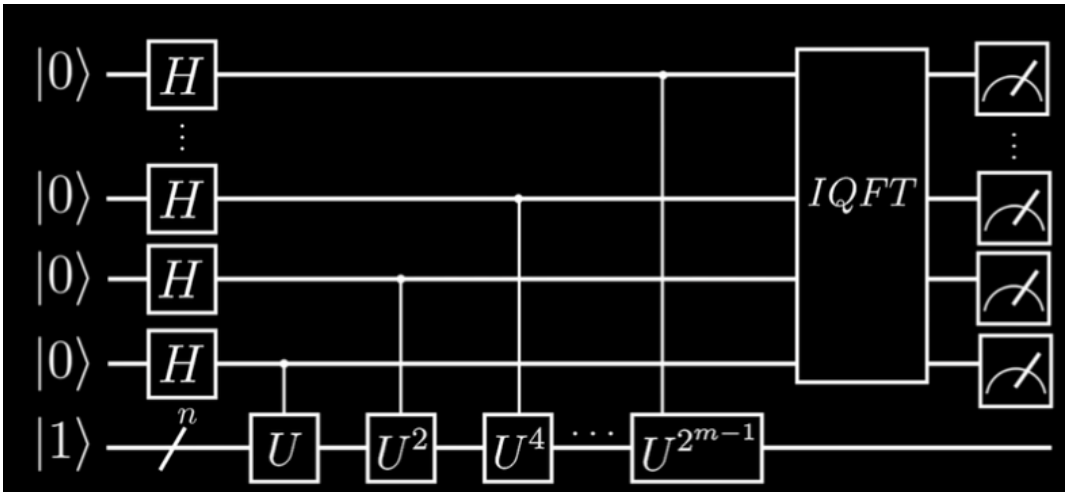

Figure 6: Quantum circuit for period finding in Shor's algorithm

Goal: Period Finding

Given a composite number N, and a random integer a such that $\gcd(a, N) = 1$, we define the function:

$$f(x) = a^x \mod N$$

This function is periodic with some unknown period r. If we can determine r, then with high probability, we can find a factor of N using:

$$\gcd(a^{r/2} \pm 1, N)$$

Quantum Circuit Steps

The quantum part finds this period using the following steps:

1. **Initialize Two Registers:**
   - The first register has n qubits initialized to $|0\rangle^{\otimes n}$.
   - The second register is initialized to $|1\rangle$ (or more generally $|v\rangle$).

   $$|\psi_0\rangle = |0\rangle^{\otimes n} \otimes |1\rangle$$

2. **Superposition:** Apply Hadamard gates $H^{\otimes n}$ to the first register to create a uniform superposition:

   $$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |1\rangle$$

3. **Modular Exponentiation:** Apply the unitary operation U_f which performs modular exponentiation:

   $$U_f : |x\rangle |0\rangle \mapsto |x\rangle |a^x \mod N\rangle$$

   This step entangles the two registers such that:

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

4. **(Optional) Measurement of Second Register:** Optionally, we can measure the second register. If we observe $f(x_0)$, the first register collapses to a superposition of all x such that $f(x) = f(x_0)$, which are all separated by the period r:

   $$|\psi_3\rangle = \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |x_0 + kr\rangle$$

   This is a periodic state in the first register with period r.

5. **Apply Inverse QFT:** Apply the inverse Quantum Fourier Transform (IQFT) to the first register to transform periodicity in the computational basis into peaks in the frequency domain:

   $$|\psi_4\rangle = \text{IQFT}\left(\sum_{k=0}^{m-1} |x_0 + kr\rangle\right)$$

   The result will, with high probability, be a value y such that:

   $$\frac{y}{2^n} \approx \frac{s}{r}$$

   for some integer s. Using the continued fractions algorithm, we can recover r from this. Using continued fraction expansion on $\frac{y}{2^n}$, we can estimate the unknown period r, provided s and r are co-prime and r is not too large.

6. **Measure First Register:** Measure the first register to obtain y, from which we estimate r.

7. **Classical Post-Processing:** Using the estimated r, compute:

   $$\gcd(a^{r/2} \pm 1, N)$$

   With high probability, this yields a non-trivial factor of N.

Summary

The quantum part of Shor's algorithm uses the modular exponentiation and the QFT (via phase kickback and interference) to extract periodicity information. The Inverse QFT is the key that converts hidden periodicity into measurable amplitude peaks. Combined with classical processing, this allows us to factor large numbers exponentially faster than any known classical method.

5 Conclusion

Quantum computing represents a fundamental shift in computational power, capable of solving certain problems exponentially faster than classical computers. One of the most prominent examples is Shor's algorithm for integer factorization, which breaks RSA encryption—a cornerstone of modern cybersecurity.

Classical vs Quantum Complexity

For the integer factorization problem[2, 5]:

- **Classical algorithms:** The best known classical factoring algorithm (the General Number Field Sieve) runs in sub-exponential time:

$$\exp\left((1.9 + o(1))(\log N)^{1/3}(\log\log N)^{2/3}\right)$$

- **Shor's algorithm (quantum):** Runs in polynomial time:

$$O((\log N)^3)$$

This exponential gap illustrates the immense potential of quantum algorithms in cryptography and other fields involving complex structure (e.g., chemistry, optimization, and machine learning).

Implications in Real Life

While full-scale, fault-tolerant quantum computers are still in development, progress in quantum hardware (such as superconducting qubits and trapped ions) has already demonstrated basic versions of quantum algorithms on small scales.

In practical terms, quantum computing could:

- Break widely used cryptographic systems (e.g., RSA, ECC).
- Simulate molecular and quantum systems for drug discovery and materials science.
- Solve optimization problems in logistics, finance, and AI.

Therefore, quantum computing not only poses challenges—such as the need for quantum-safe encryption—but also promises transformative advances across science and industry.

References

[1] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, *Quantum algorithms revisited*, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 454 (1998), pp. 339–354.

[2] A. K. Lenstra and H. W. Lenstra, *The Development of the Number Field Sieve*, vol. 1554 of Lecture Notes in Mathematics, Springer-Verlag, 1993.

[3] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010.

[4] R. L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM, 21 (1978), pp. 120–126.

[5] P. W. Shor, *Algorithms for quantum computation: Discrete logarithms and factoring*, Proceedings 35th Annual Symposium on Foundations of Computer Science, (1994), pp. 124–134.

[6] V. Shoup, *A Computational Introduction to Number Theory and Algebra*, Cambridge University Press, 2009.