Number of Qubits in Quantum Factoring

SAC 2025

Pierre-Alain Fouque

Université de Rennes

Contents

Cryptography: Hard Computational problems (I)

In 1978, Rivest, Shamir, and Adleman described the RSA cryptosystem whose security is related to the untractability of factoring

Factorization Problem
Given an integer $N = pq$, where p and q are two primes. Recover p ?

Cryptography: Hard Computational problems (I)

In 1978, Rivest, Shamir, and Adleman described the RSA cryptosystem whose security is related to the untractability of factoring

Factorization Problem
Given an integer $N = pq$, where p and q are two primes. Recover p ?

Classical algorithm:

- Number Field Sieve (NFS). Complexity: $2^{\tilde{O}(n^{1/3})}$ (constants matter...) where n is the size of N: $n = \log_2(N)$
- Record: 250-digits (830 bits): 2700 computer years
- $\approx 2^{128}$ for a 2048-bit modulus

Cryptography: Hard Computational problems (II)

Discrete Logarithm
Let p a prime and q a prime divisor of $p-1$, and g a generator of the q-order subgroup of $(\mathbb{Z}/p\mathbb{Z})^*$. Given $y = g^x \bmod p$, recover x ?

Cryptography: Hard Computational problems (II)

Discrete Logarithm
Let p a prime and q a prime divisor of $p-1$, and g a generator of the q-order subgroup of $(\mathbb{Z}/p\mathbb{Z})^*$. Given $y = g^x \bmod p$, recover x ?

Example: $g = 2$ **in** $(\mathbb{Z}/11\mathbb{Z})^*$

Cryptography: Hard Computational problems (II)

Discrete Logarithm
Let p a prime and q a prime divisor of $p-1$, and g a generator of the q-order subgroup of $(\mathbb{Z}/p\mathbb{Z})^*$. Given $y = g^x \bmod p$, recover x ?

Example: $g = 2$ **in** $(\mathbb{Z}/11\mathbb{Z})^*$

- $2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 5 \bmod 11, 2^5 = 10 \bmod 11, 2^6 = 9 \bmod 11, 2^7 = 7 \bmod 11, 2^8 = 3 \bmod 11, 2^9 = 6 \bmod 11, 2^{10} = 1 \bmod 11...$

Cryptography: Hard Computational problems (II)

Discrete Logarithm
Let p a prime and q a prime divisor of $p-1$, and g a generator of the q-order subgroup of $(\mathbb{Z}/p\mathbb{Z})^*$. Given $y = g^x \bmod p$, recover x ?

Example: $g = 2$ **in** $(\mathbb{Z}/11\mathbb{Z})^*$

- $2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 5 \bmod 11, 2^5 = 10 \bmod 11, 2^6 = 9 \bmod 11, 2^7 = 7 \bmod 11, 2^8 = 3 \bmod 11, 2^9 = 6 \bmod 11, 2^{10} = 1 \bmod 11$...
- What is the subgroup generated by 4 ? generated by 10 ?

Cryptography: Hard Computational problems (II)

Discrete Logarithm
Let p a prime and q a prime divisor of $p-1$, and g a generator of the q-order subgroup of $(\mathbb{Z}/p\mathbb{Z})^*$. Given $y = g^x \bmod p$, recover x ?

Example: $g = 2$ **in** $(\mathbb{Z}/11\mathbb{Z})^*$

- $2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 5 \bmod 11, 2^5 = 10 \bmod 11, 2^6 = 9 \bmod 11, 2^7 = 7 \bmod 11, 2^8 = 3 \bmod 11, 2^9 = 6 \bmod 11, 2^{10} = 1 \bmod 11$...
- What is the subgroup generated by 4 ? generated by 10 ?
- As $(\mathbb{Z}/p\mathbb{Z})^*$ is cyclic, for all $d | p-1$, there is a subgroup of order d

Cryptography: Hard Computational problems (II)

Discrete Logarithm
Let p a prime and q a prime divisor of $p-1$, and g a generator of the q-order subgroup of $(\mathbb{Z}/p\mathbb{Z})^*$. Given $y = g^x \bmod p$, recover x ?

Example: $g = 2$ **in** $(\mathbb{Z}/11\mathbb{Z})^*$

- $2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 5 \bmod 11, 2^5 = 10 \bmod 11, 2^6 = 9 \bmod 11, 2^7 = 7 \bmod 11, 2^8 = 3 \bmod 11, 2^9 = 6 \bmod 11, 2^{10} = 1 \bmod 11 ...$
- What is the subgroup generated by 4 ? generated by 10 ?
- As $(\mathbb{Z}/p\mathbb{Z})^*$ is cyclic, for all $d | p-1$, there is a subgroup of order d

Complexity and Security level

- Classical algorithms: Pollard $\sqrt{q}$ and NFS: $2^{\tilde{O}((\log_2 p)^{1/3})}$

Cryptography: Hard Computational problems (II)

Discrete Logarithm
Let p a prime and q a prime divisor of $p-1$, and g a generator of the q-order subgroup of $(\mathbb{Z}/p\mathbb{Z})^*$. Given $y = g^x \bmod p$, recover x ?

Example: $g = 2$ **in** $(\mathbb{Z}/11\mathbb{Z})^*$

- $2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 5 \bmod 11, 2^5 = 10 \bmod 11, 2^6 = 9 \bmod 11, 2^7 = 7 \bmod 11, 2^8 = 3 \bmod 11, 2^9 = 6 \bmod 11, 2^{10} = 1 \bmod 11$...
- What is the subgroup generated by 4 ? generated by 10 ?
- As $(\mathbb{Z}/p\mathbb{Z})^*$ is cyclic, for all $d | p-1$, there is a subgroup of order d

Complexity and Security level

- Classical algorithms: Pollard $\sqrt{q}$ and NFS: $2^{\tilde{O}((\log_2 p)^{1/3})}$
- p a 2048-bit prime and q a 256-bit prime

Shor's quantum factorisation algorithm (1996)

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

Breakthrough

- Polynomial-time algorithm $O(n^2)$ gates and $O(n)$ qubits

Shor's quantum factorisation algorithm (1996)

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

Breakthrough

- Polynomial-time algorithm $O(n^2)$ gates and $O(n)$ qubits
- If we were able to built a noise-free quantum algorithm, we will be able to break all communications...

Shor's quantum factorisation algorithm (1996)

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

Breakthrough

- Polynomial-time algorithm $O(n^2)$ gates and $O(n)$ qubits
- If we were able to built a noise-free quantum algorithm, we will be able to break all communications...
- Post-Quantum Cryptography: classical algorithms where hard problems are conjectured to resist quantum computers ...

Shor's quantum factorisation algorithm (1996)

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

Breakthrough

- Polynomial-time algorithm $O(n^2)$ gates and $O(n)$ qubits
- If we were able to built a noise-free quantum algorithm, we will be able to break all communications...
- Post-Quantum Cryptography: classical algorithms where hard problems are conjectured to resist quantum computers ...
- E.g.: hard lattice problems, coding problems, ...

Shor's quantum factorisation algorithm (1996)

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

Breakthrough

- Polynomial-time algorithm $O(n^2)$ gates and $O(n)$ qubits
- If we were able to built a noise-free quantum algorithm, we will be able to break all communications...
- Post-Quantum Cryptography: classical algorithms where hard problems are conjectured to resist quantum computers ...
- E.g.: hard lattice problems, coding problems, ...
- Standards are available since 2024 and the transition to PQC begins

Basic Quantum Information and Computation

1-qubit

1. 2 base state qubits $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
2. a quantum state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$, superposition of base qubits = linear combination, with $\alpha, \beta \in \mathbb{C}$
3. Eg.: $|\psi\rangle = (3 + 4i) |0\rangle + (2 - 8i) |1\rangle$, where $i^2 = -1$

1-qubit

1. 2 base state qubits $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
2. a quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$, superposition of base qubits = linear combination, with $\alpha, \beta \in \mathbb{C}$
3. Eg.: $|\psi\rangle = (3+4i)|0\rangle + (2-8i)|1\rangle$, where $i^2 = -1$
4. Norm: $|\alpha|^2 + |\beta|^2 = 1$: $|\psi\rangle = \frac{3+4i}{\sqrt{93}}|0\rangle + \frac{2-8i}{\sqrt{93}}|1\rangle$
5. If we measure $|\psi\rangle$, 0 with proba. $|\alpha|^2$ and 1 with proba. $|\beta|^2$
6. $|\phi\rangle$ and $|\psi\rangle$ are equivalent if there exists $z \in \mathbb{C}$ s.t. $|\phi\rangle = z|\psi\rangle$. Such qubits cannot be distinguished by measures.

Quantum Gates

1. Gate X/NOT: $|0\rangle \mapsto |1\rangle \quad |1\rangle \mapsto |0\rangle$
2. By linearity, $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, $\mathsf{X}\,|\psi\rangle = \beta\,|0\rangle + \alpha\,|1\rangle$
3. Matrix version: $M_{\mathsf{X}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ Since

   $M_{\mathsf{X}}\,|0\rangle = M_{\mathsf{X}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$ and

   $M_{\mathsf{X}}\,|1\rangle = M_{\mathsf{X}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$

Quantum Hadamard Gates

A very important gate

1. Gate H: $|0\rangle \mapsto \frac{|0\rangle+|1\rangle}{\sqrt{2}} \quad |1\rangle \mapsto \frac{|0\rangle-|1\rangle}{\sqrt{2}}$
2. By linearity, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, $\mathsf{H}|\psi\rangle = \alpha\mathsf{H}|0\rangle + \beta\mathsf{H}|1\rangle$
   $\mathsf{H}|\psi\rangle = \frac{\alpha}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{\beta}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{\alpha+\beta}{\sqrt{2}}|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}|1\rangle$

Quantum Hadamard Gates

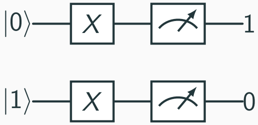
A very important gate

1. Gate H: $|0\rangle \mapsto \frac{|0\rangle+|1\rangle}{\sqrt{2}} \quad |1\rangle \mapsto \frac{|0\rangle-|1\rangle}{\sqrt{2}}$
2. By linearity, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, $\mathsf{H}|\psi\rangle = \alpha\mathsf{H}|0\rangle + \beta\mathsf{H}|1\rangle$
   $\mathsf{H}|\psi\rangle = \frac{\alpha}{\sqrt{2}}(|0\rangle+|1\rangle) + \frac{\beta}{\sqrt{2}}(|0\rangle-|1\rangle) = \frac{\alpha+\beta}{\sqrt{2}}|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}|1\rangle$
3. Matrix version: $M_{\mathsf{H}} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

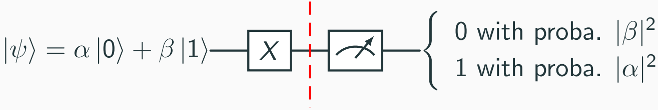
$$M_{\mathsf{H}}|0\rangle = M_{\mathsf{H}}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{|0\rangle+|1\rangle}{\sqrt{2}}.$$

   Similarly for $M_{\mathsf{H}}|1\rangle$.
4. Eg., if $|\psi\rangle = i|0\rangle + (2+i)|1\rangle$, compute $M_{\mathsf{H}}|\psi\rangle$?

Some Quantum Circuits

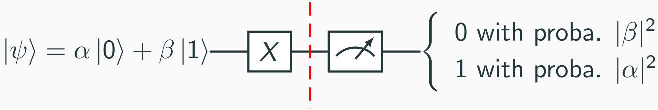


$$X|\psi\rangle = \beta|0\rangle + \alpha|1\rangle$$

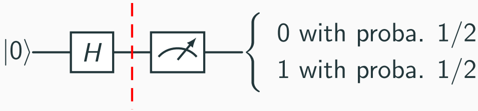


Some Quantum Circuits

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



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



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

Gates X, Y, and Z of Pauli

$$\begin{cases} |0\rangle \mapsto |1\rangle \\ |1\rangle \mapsto |0\rangle \end{cases} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\begin{cases} |0\rangle \mapsto i\,|1\rangle \\ |1\rangle \mapsto -i\,|0\rangle \end{cases} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$\begin{cases} |0\rangle \mapsto |0\rangle \\ |1\rangle \mapsto -\,|1\rangle \end{cases} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

2-qubits $\Rightarrow$ 4 possibilities

2-qubit

- $|\psi\rangle = \alpha\,|0.0\rangle + \beta\,|0.1\rangle + \gamma\,|1.0\rangle + \delta\,|1.1\rangle$, with $\alpha, \beta, \gamma, \delta \in \mathbb{C}$
- $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$
- $\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$ and $|0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |0.0\rangle$

Vectors

$$|0.0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix},\ |0.1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},\ |1.0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix},\ |1.1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},\ |\psi\rangle = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix}$$

2-qubits

Vectors

$$|0.0\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix},\ |0.1\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix},\ |1.0\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix},\ |1.1\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix},\ |\psi\rangle = \begin{pmatrix}\alpha\\\beta\\\gamma\\\delta\end{pmatrix}$$

Tensor product: not commutative product

- $|0.0\rangle = |0\rangle\,.\,|0\rangle = |0\rangle \otimes |0\rangle = \begin{matrix}|0\rangle\\\otimes\\|0\rangle\end{matrix}$
- $u = \begin{pmatrix}x_1\\x_2\\\vdots\\x_n\end{pmatrix},\ v = \begin{pmatrix}y_1\\y_2\\\vdots\\y_m\end{pmatrix},\ u \otimes v = \begin{pmatrix}x_1y_1\\\vdots\\x_1y_m\\x_2y_1\\\vdots\\x_ny_m\end{pmatrix}$

Tensor product

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$$, compute $\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, and all base vectors

Properties

- $(\lambda u) \otimes v = \lambda(u \otimes v) = u \otimes (\lambda v), \quad \text{for } \lambda \in \mathbb{C}$
- $(u_1 + u_2) \otimes v = u_1 \otimes v + u_2 \otimes v$
- $u \otimes (v_1 + v_2) = u \otimes v_1 + u \otimes v_2$

Operations on qubits

- Addition of qubits: $|\phi\rangle = (1+3i)\,|0\rangle + 2i\,|1\rangle$ and $|\psi\rangle = 3\,|0\rangle + (1-i)\,|1\rangle$,

$$|\phi\rangle + |\psi\rangle = (4+3i)\,|0\rangle + (1+i)\,|1\rangle$$

  For 2 2-qubits: $(|1.0\rangle + |0.1\rangle) + (|1.0\rangle - |0.1\rangle) = 2\,|1.0\rangle$
- Multiplication of 2 1-qubit is a 2-qubit: $|\phi\rangle \cdot |\psi\rangle$

$$((1+3i)\,|0\rangle + 2i\,|1\rangle) \otimes (3\,|0\rangle + (1-i)\,|1\rangle)$$
$$(1+3i)\cdot 3 \cdot |0\rangle\,|0\rangle + (1+3i)\cdot(1-i)\,|0\rangle\,|1\rangle + 6i \cdot |1\rangle\,|0\rangle + \ldots$$
$$(3+9i)\,|0.0\rangle + (4+2i)\,|0.1\rangle + 6i\,|1.0\rangle + (2+2i)\,|1.1\rangle$$

CNOT Gate: controlled gate with 2-qubit

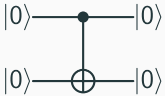
If ... then ... else ...

- $|0.0\rangle \mapsto |0.0\rangle, |0.1\rangle \mapsto |0.1\rangle, |1.0\rangle \mapsto |1.1\rangle, |1.1\rangle \mapsto |1.0\rangle$
- If $|0.0\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, |0.1\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, |1.0\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, |1.1\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$,

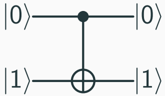
  $M = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$, the upper left submatrix is the identity performed on the first line, the bottom right submatrix is the inversion operation performed on the second line

Gate CNOT with 2-qubits

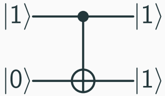
$|0.0\rangle \mapsto |0.0\rangle$



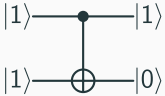
$|0.1\rangle \mapsto |0.1\rangle$



$|1.0\rangle \mapsto |1.1\rangle$



$|1.1\rangle \mapsto |1.0\rangle$



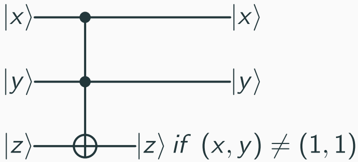
n-qubits

- $|\psi\rangle = \alpha_0 |0.0..0\rangle + \alpha_1 |0.0..0.1\rangle + \ldots + \alpha_{2^n-1} |1.1..1\rangle$
- $|\psi\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix} \in \mathbb{C}^{2^n}$
- $\| \, |\psi\rangle \, \| = \sqrt{|\alpha_0|^2 + |\alpha_1|^2 + \ldots + |\alpha_{2^n-1}|^2}$
- Measure: 0.0...0 with proba. $|\alpha_0|^2$, 0.0...0.1 with proba. $|\alpha_1|^2$, ... 1.1....1 with proba. $|\alpha_{2^n-1}|^2$

n-qubits

- $|\psi\rangle = \alpha_0 |0.0..0\rangle + \alpha_1 |0.0..0.1\rangle + \ldots + \alpha_{2^n-1} |1.1..1\rangle$
- $|\psi\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix} \in \mathbb{C}^{2^n}$
- $\| \, |\psi\rangle \, \| = \sqrt{|\alpha_0|^2 + |\alpha_1|^2 + \ldots + |\alpha_{2^n-1}|^2}$
- Measure: 0.0...0 with proba. $|\alpha_0|^2$, 0.0...0.1 with proba. $|\alpha_1|^2$, ... 1.1....1 with proba. $|\alpha_{2^n-1}|^2$

3-qubit Toffoli Gate (CCNOT)



Quantum Circuit

where A is a unitary $A^* A = I_n$

Theorem
Every n-qubit quantum gate can be realized with a circuit using only CNOT and 1-qubit gates

Quantum Circuit

where A is a unitary $A^*A = I_n$

Theorem
Every n-qubit quantum gate can be realized with a circuit using only CNOT and 1-qubit gates

Theorem (Solovay-Kitaev)
There is an infinite number of 1-qubit gates, and every such gate can be approximated with only H, T, and CNOT gates

The T gate: $|0\rangle \mapsto |0\rangle$ and $|1\rangle \mapsto e^{i\pi/4}\,|1\rangle$: $T = e^{i\pi/8}\begin{pmatrix} e^{-\pi/8} & 0 \\ 0 & e^{\pi/8} \end{pmatrix}$

Quantum Circuit

$|\psi\rangle \xrightarrow{/^n} \boxed{A} \xrightarrow{/^n} A|\psi\rangle$ where A is a unitary $A^*A = I_n$

Theorem
Every n-qubit quantum gate can be realized with a circuit using only CNOT and 1-qubit gates

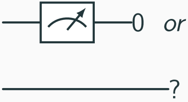
Theorem (Solovay-Kitaev)
There is an infinite number of 1-qubit gates, and every such gate can be approximated with only H, T, and CNOT gates

Theorem: Toffoli (CCNOT) is a universal gate

- Toffoli gate is invertible: $(|a.b.c\rangle \mapsto |a.b.c \oplus (ab)\rangle)$:
  $T\,|a.b.1\rangle = |a.b.NAND(a,b)\rangle$
- Any classical circuit using N gates in the set AND, OR, NOT (universal gates for classical circuits) can be computed using $O(N)$ Toffoli gates

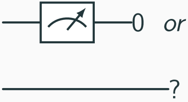
Basic Circuits: Deutsch-Jozsa and Simon algorithms

Partial Measurement of a 2-qubit

- $|\psi\rangle = \alpha\,|0.0\rangle + \beta\,|0.1\rangle + \gamma\,|1.0\rangle + \delta\,|1.1\rangle$, $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$
- $|\psi\rangle$ 

- Let $|\psi\rangle = \frac{\sqrt{2}}{2}\,|0.0\rangle + \frac{1}{2}\,|0.1\rangle + \frac{1}{2}\,|1.1\rangle$. If one measures the first qubit as 1, what is the second qubit ?

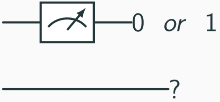
Partial Measurement of a 2-qubit

- $|\psi\rangle = \alpha\,|0.0\rangle + \beta\,|0.1\rangle + \gamma\,|1.0\rangle + \delta\,|1.1\rangle$, $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$
- $|\psi\rangle$ 

- Let $|\psi\rangle = \frac{\sqrt{2}}{2}\,|0.0\rangle + \frac{1}{2}\,|0.1\rangle + \frac{1}{2}\,|1.1\rangle$. If one measures the first qubit as 1, what is the second qubit ?
- the second is $|1\rangle$, but what if we observe $|0\rangle$?

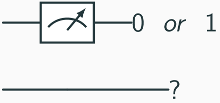
Partial Measurement of a 2-qubit

- $|\psi\rangle = \alpha\,|0.0\rangle + \beta\,|0.1\rangle + \gamma\,|1.0\rangle + \delta\,|1.1\rangle$, $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$
- $|\psi\rangle$



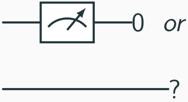
- Let $|\psi\rangle = \frac{\sqrt{2}}{2}\,|0.0\rangle + \frac{1}{2}\,|0.1\rangle + \frac{1}{2}\,|1.1\rangle$. If one measures the first qubit as 1, what is the second qubit ?
- the second is $|1\rangle$, but what if we observe $|0\rangle$?
- $|\psi\rangle = \frac{|0\rangle}{2} \cdot (\sqrt{2}\,|0\rangle + |1\rangle) + \frac{1}{2}\,|1\rangle\,|1\rangle$, the 2^{nd} is $\sqrt{\frac{2}{3}}\,|0\rangle + \frac{1}{\sqrt{3}}\,|1\rangle$

Partial Measurement of a 2-qubit

- $|\psi\rangle = \alpha\,|0.0\rangle + \beta\,|0.1\rangle + \gamma\,|1.0\rangle + \delta\,|1.1\rangle$, $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$
- $|\psi\rangle$ 

- Let $|\psi\rangle = \frac{\sqrt{2}}{2}\,|0.0\rangle + \frac{1}{2}\,|0.1\rangle + \frac{1}{2}\,|1.1\rangle$. If one measures the first qubit as 1, what is the second qubit ?
- the second is $|1\rangle$, but what if we observe $|0\rangle$?
- $|\psi\rangle = \frac{|0\rangle}{2} \cdot (\sqrt{2}\,|0\rangle + |1\rangle) + \frac{1}{2}\,|1\rangle\,|1\rangle$, the 2^{nd} is $\sqrt{\frac{2}{3}}\,|0\rangle + \frac{1}{\sqrt{3}}\,|1\rangle$
- More generally, $|\psi\rangle = |0\rangle \cdot (\alpha\,|0\rangle + \beta\,|1\rangle) + |1\rangle \cdot (\gamma\,|0\rangle + \delta\,|1\rangle)$, and if one measures $|0\rangle$ for the first qubit, the second $\frac{\alpha}{\sqrt{|\alpha|^2+|\beta|^2}}\,|0\rangle + \frac{\beta}{\sqrt{|\alpha|^2+|\beta|^2}}\,|1\rangle$

Partial Measurement of a 2-qubit

- $|\psi\rangle = \alpha\,|0.0\rangle + \beta\,|0.1\rangle + \gamma\,|1.0\rangle + \delta\,|1.1\rangle$, $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$
- $|\psi\rangle$ 

- Let $|\psi\rangle = \frac{\sqrt{2}}{2}\,|0.0\rangle + \frac{1}{2}\,|0.1\rangle + \frac{1}{2}\,|1.1\rangle$. If one measures the first qubit as 1, what is the second qubit ?
- the second is $|1\rangle$, but what if we observe $|0\rangle$?
- $|\psi\rangle = \frac{|0\rangle}{2} \cdot (\sqrt{2}\,|0\rangle + |1\rangle) + \frac{1}{2}\,|1\rangle\,|1\rangle$, the 2$^{\text{nd}}$ is $\sqrt{\frac{2}{3}}\,|0\rangle + \frac{1}{\sqrt{3}}\,|1\rangle$
- Exo: If $|\psi\rangle = \frac{1}{5}(2\,|0.0.0\rangle - |0.0.1\rangle + 3\,|0.1.0\rangle + |0.1.1\rangle - 2\,|1.0.0\rangle + 2\,|1.0.1\rangle + \sqrt{2}\,|1.1.1\rangle)$, and we measure 0.0, what is the last qubit ?

Quantum oracle gate

Oracle

- Let $f : E \longrightarrow \mathbb{Z}/2\mathbb{Z}$ be a function
- $(\mathbb{Z}/2\mathbb{Z}, +) = (\{0, 1\}, \oplus)$
- $F : E \times \mathbb{Z}/2\mathbb{Z} \longrightarrow E \times \mathbb{Z}/2\mathbb{Z}, \quad (x, y) \longmapsto (x, y \oplus f(x))$, is a bijection

Quantum oracle gate

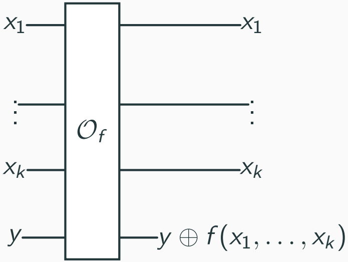
Oracle

- Let $f : E \longrightarrow \mathbb{Z}/2\mathbb{Z}$ be a function
- $(\mathbb{Z}/2\mathbb{Z}, +) = (\{0, 1\}, \oplus)$
- $F : E \times \mathbb{Z}/2\mathbb{Z} \longrightarrow E \times \mathbb{Z}/2\mathbb{Z}, \quad (x, y) \longmapsto (x, y \oplus f(x))$, is a bijection
- Proof: $F^{-1} = F$, $F(F(x, y)) = F(x, y \oplus f(x)) = (x, y)$

Quantum oracle gate

Oracle

- Let $f : E \longrightarrow \mathbb{Z}/2\mathbb{Z}$ be a function
- $(\mathbb{Z}/2\mathbb{Z}, +) = (\{0,1\}, \oplus)$
- $F : E \times \mathbb{Z}/2\mathbb{Z} \longrightarrow E \times \mathbb{Z}/2\mathbb{Z}, \quad (x,y) \longmapsto (x, y \oplus f(x))$, is a bijection
- Proof: $F^{-1} = F$, $F(F(x,y)) = F(x, y \oplus f(x)) = (x,y)$
- Deutsch-Jozsa Oracle $f : (\mathbb{Z}/2\mathbb{Z})^k \longrightarrow \mathbb{Z}/2\mathbb{Z}$:



Deutsch-Jozsa problem

Goal

- Let $f : \{0,1\} \longrightarrow \{0,1\}$.
- There are 4 such functions: two are constant and two are balanced (0 and 1 are taken the same number of times)
  $f_0 = \begin{cases} 0 \mapsto 0 \\ 1 \mapsto 0 \end{cases}$ $f_1 = \begin{cases} 0 \mapsto 1 \\ 1 \mapsto 1 \end{cases}$ $f_2 = \begin{cases} 0 \mapsto 0 \\ 1 \mapsto 1 \end{cases}$ $f_3 = \begin{cases} 0 \mapsto 1 \\ 1 \mapsto 0 \end{cases}$
- **Decide** if f is constant or balanced ?

Deutsch-Jozsa problem

Goal

- Let $f : \{0,1\} \longrightarrow \{0,1\}$.
- There are 4 such functions: two are constant and two are balanced (0 and 1 are taken the same number of times)
$$f_0 = \begin{cases} 0 \mapsto 0 \\ 1 \mapsto 0 \end{cases} \quad f_1 = \begin{cases} 0 \mapsto 1 \\ 1 \mapsto 1 \end{cases} \quad f_2 = \begin{cases} 0 \mapsto 0 \\ 1 \mapsto 1 \end{cases} \quad f_3 = \begin{cases} 0 \mapsto 1 \\ 1 \mapsto 0 \end{cases}$$
- **Decide** if f is constant or balanced ?
- Classically, ask 2 queries ($f(0)$ and $f(1)$), quantumly 1 query !

Deutsch-Jozsa problem

Goal

- Let $f : \{0,1\} \longrightarrow \{0,1\}$.
- There are 4 such functions: two are constant and two are balanced (0 and 1 are taken the same number of times)

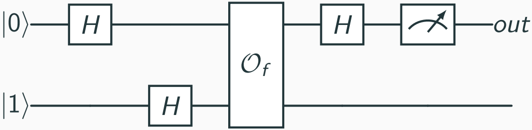
$$f_0 = \begin{cases} 0 \mapsto 0 \\ 1 \mapsto 0 \end{cases} \quad f_1 = \begin{cases} 0 \mapsto 1 \\ 1 \mapsto 1 \end{cases} \quad f_2 = \begin{cases} 0 \mapsto 0 \\ 1 \mapsto 1 \end{cases} \quad f_3 = \begin{cases} 0 \mapsto 1 \\ 1 \mapsto 0 \end{cases}$$

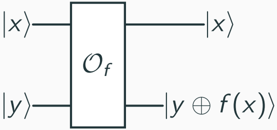
- **Decide** if f is constant or balanced ?
- Classically, ask 2 queries ($f(0)$ and $f(1)$), quantumly 1 query !

Exponential gap: Let $f : \{0,1\}^n \longrightarrow \{0,1\}$ and we have the promise f is either balanced or constant.

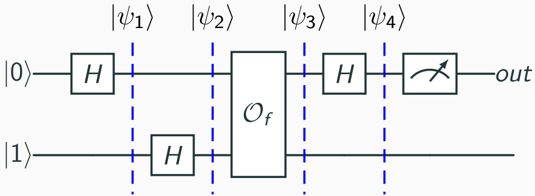
Classically, one need at most $2^{n-1} + 1$ queries, while only 1 quantumly !

Deutsch-Jozsa Quantum Circuit ($n = 1$)

- 

- with 

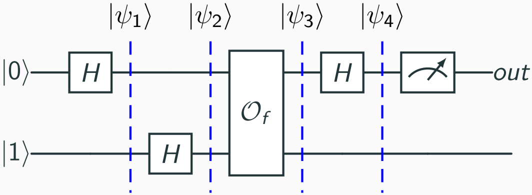
Deutsch-Jozsa Quantum Circuit ($n = 1$)

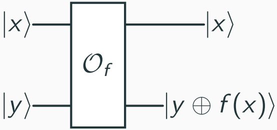
- 

$|\psi_1\rangle = (|0\rangle + |1\rangle).\,|1\rangle,$

$|\psi_2\rangle = (|0\rangle + |1\rangle).(|0\rangle - |1\rangle) = 0.0 - 0.1 + 1.0 - 1.1$

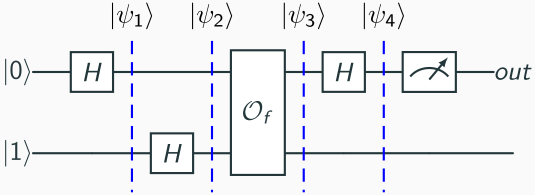
Deutsch-Jozsa Quantum Circuit ($n = 1$)

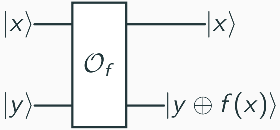




- $|\psi_2\rangle = 0.0 - 0.1 + 1.0 - 1.1,$

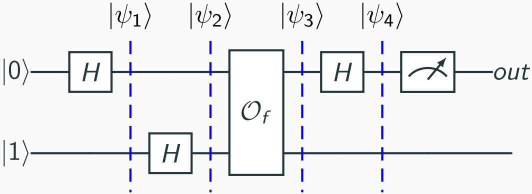
Deutsch-Jozsa Quantum Circuit ($n = 1$)

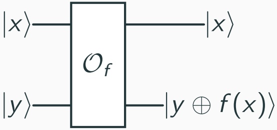




- $|\psi_2\rangle = 0.0 - 0.1 + 1.0 - 1.1,$
- $|\psi_3\rangle = \underbrace{0.(0 \oplus f(0)) - 0.(1 \oplus f(0))}_{A} + \underbrace{1.(0 \oplus f(1)) - 1.(1 \oplus f(1))}_{B}$
- $A = \begin{cases} 0.0 - 0.1 \text{ if } f(0) = 0 \\ -(0.0 - 0.1) \text{ if } f(0) = 1 \end{cases}$ so $A = (-1)^{f(0)}(0.0 - 0.1)$

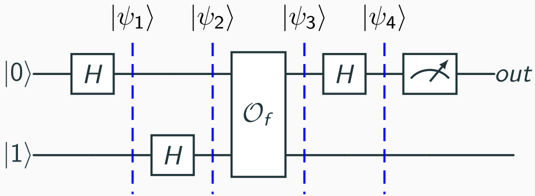
Deutsch-Jozsa Quantum Circuit ($n = 1$)





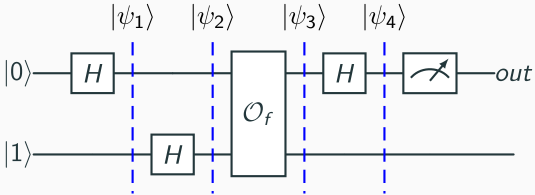
- $|\psi_2\rangle = 0.0 - 0.1 + 1.0 - 1.1,$
- $|\psi_3\rangle = \underbrace{0.(0 \oplus f(0)) - 0.(1 \oplus f(0))}_{A} + \underbrace{1.(0 \oplus f(1)) - 1.(1 \oplus f(1))}_{B}$
- $A = (-1)^{f(0)}(0.0 - 0.1)$ and $B = (-1)^{f(1)}(1.0 - 1.1)$
- $|\psi_3\rangle = (-1)^{f(0)}(0.0 - 0.1) + (-1)^{f(1)}(1.0 - 1.1)$

Deutsch-Jozsa Quantum Circuit ($n = 1$)

- 

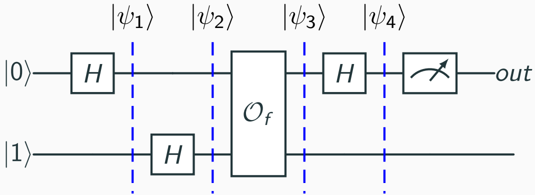
- $|\psi_3\rangle = (-1)^{f(0)}(0.0 - 0.1) + (-1)^{f(1)}(1.0 - 1.1)$
- $|\psi_4\rangle = (-1)^{f(0)}((0+1).0 - (0+1).1) + (-1)^{f(1)}((0-1).0 - (0-1).1)$
- $|\psi_4\rangle = (-1)^{f(0)}(0.0 - 0.1 + 1.0 - 1.1) + (-1)^{f(1)}(0.0 - 0.1 - 1.0 + 1.1)$

Deutsch-Jozsa Quantum Circuit ($n = 1$)

- 

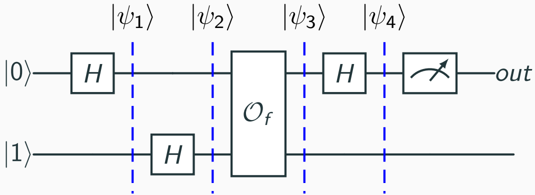
- $|\psi_4\rangle = (-1)^{f(0)}(0.0 - 0.1 + 1.0 - 1.1) + (-1)^{f(1)}(0.0 - 0.1 - 1.0 + 1.1)$

Deutsch-Jozsa Quantum Circuit ($n = 1$)

- 

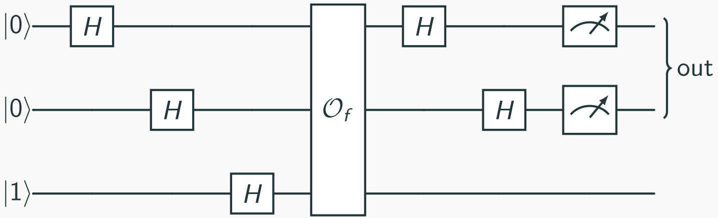
- $|\psi_4\rangle = (-1)^{f(0)}(0.0 - 0.1 + 1.0 - 1.1) + (-1)^{f(1)}(0.0 - 0.1 - 1.0 + 1.1)$
- $|\psi_4\rangle = ((-1)^{f(0)} + (-1)^{f(1)})0.0 + (-(-1)^{f(0)} - (-1)^{f(1)})0.1 + ((-1)^{f(0)} - (-1)^{f(1)})1.0 + (-(-1)^{f(0)} + (-1)^{f(1)})1.1$

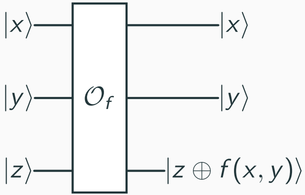
Deutsch-Jozsa Quantum Circuit ($n = 1$)

- 

- $|\psi_4\rangle = (-1)^{f(0)}(0.0 - 0.1 + 1.0 - 1.1) + (-1)^{f(1)}(0.0 - 0.1 - 1.0 + 1.1)$
- $|\psi_4\rangle = ((-1)^{f(0)} + (-1)^{f(1)})0.0 + (-(-1)^{f(0)} - (-1)^{f(1)})0.1 + ((-1)^{f(0)} - (-1)^{f(1)})1.0 + (-(-1)^{f(0)} + (-1)^{f(1)})1.1$
- If f is constant, $(-1)^{f(0)} + (-1)^{f(1)} = \pm 2$ and $(-1)^{f(0)} - (-1)^{f(1)} = 0$ and $(-1)^{f(0)} - (-1)^{f(1)} = 0$, so $|\psi_4\rangle = 0.0 - 0.1$ the measure of the first qubit 0 in both cases
- If f is balanced, check that the first bit is 1

Deutsch-Jozsa Circuit for $n = 2$





- Check that if f is constant, the final state before the measurement is $\pm |0.0\rangle \left| \frac{1}{\sqrt{2}}(0-1) \right\rangle$, and the 2 first bits are 0.0
- if f is balanced, the final state does not contain qubits starting with 0.0, so no measurement of these qubits will give 0.0.

Simon algorithm

Problem
Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ a 2-to-1 function so that there exists $c \in \{0,1\}^n$ such that

$$f(x) = f(x \oplus c), \text{ where } \oplus \text{ is bitwise exclusive or}$$

Simon algorithm

Problem
Let $f : \{0,1\}^n \to \{0,1\}^n$ a 2-to-1 function so that there exists $c \in \{0,1\}^n$ such that

$$f(x) = f(x \oplus c), \text{ where } \oplus \text{ is bitwise exclusive or}$$

Example

$$\begin{array}{ll} \text{f}(000) = 101 & \text{f}(100) = 011 \\ \text{f}(001) = 010 & \text{f}(101) = 100 \\ \text{f}(010) = 011 & \text{f}(110) = 101 \\ \text{f}(011) = 100 & \text{f}(111) = 010 \end{array}$$

What is c ?

Simon algorithm

Problem
Let $f : \{0,1\}^n \to \{0,1\}^n$ a 2-to-1 function so that there exists $c \in \{0,1\}^n$ such that

$$f(x) = f(x \oplus c), \text{ where } \oplus \text{ is bitwise exclusive or}$$

Example

$$\begin{array}{ll} \text{f}(000) = 101 & \text{f}(100) = 011 \\ \text{f}(001) = 010 & \text{f}(101) = 100 \\ \text{f}(010) = 011 & \text{f}(110) = 101 \\ \text{f}(011) = 100 & \text{f}(111) = 010 \end{array}$$

What is c ? $c = 110$

Simon algorithm

Problem
Let $f : \{0,1\}^n \to \{0,1\}^n$ a 2-to-1 function so that there exists $c \in \{0,1\}^n$ such that

$$f(x) = f(x \oplus c), \text{ where } \oplus \text{ is bitwise exclusive or}$$

Example

$$\begin{array}{ll} \text{f}(000) = 101 & \text{f}(100) = 011 \\ \text{f}(001) = 010 & \text{f}(101) = 100 \\ \text{f}(010) = 011 & \text{f}(110) = 101 \\ \text{f}(011) = 100 & \text{f}(111) = 010 \end{array}$$

What is c ? $c = 110$

Classical algorithms

Simon algorithm

Problem

Let $f : \{0,1\}^n \to \{0,1\}^n$ a 2-to-1 function so that there exists $c \in \{0,1\}^n$ such that

$$f(x) = f(x \oplus c), \text{ where } \oplus \text{ is bitwise exclusive or}$$

Example

$$\begin{array}{ll} \text{f}(000) = 101 & \text{f}(100) = 011 \\ \text{f}(001) = 010 & \text{f}(101) = 100 \\ \text{f}(010) = 011 & \text{f}(110) = 101 \\ \text{f}(011) = 100 & \text{f}(111) = 010 \end{array}$$

What is c ? $c = 110$

Classical algorithms

- Compute $f(x)$ until a collision $f(x_1) = f(x_2)$ ... and then $c = x_1 \oplus x_2$

Simon algorithm

Problem
Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ a 2-to-1 function so that there exists $c \in \{0,1\}^n$ such that

$$f(x) = f(x \oplus c), \text{ where } \oplus \text{ is bitwise exclusive or}$$

Example

$$\begin{array}{ll} \text{f}(000) = 101 & \text{f}(100) = 011 \\ \text{f}(001) = 010 & \text{f}(101) = 100 \\ \text{f}(010) = 011 & \text{f}(110) = 101 \\ \text{f}(011) = 100 & \text{f}(111) = 010 \end{array}$$

What is c ? $c = 110$

Classical algorithms

- Compute $f(x)$ until a collision $f(x_1) = f(x_2)$ ... and then $c = x_1 \oplus x_2$
- Another solution: since $f(000) \neq f(001)$, $c \neq 001$, ...

Simon Quantum Algorithm

Hadamard Transform

- $H^{\otimes n} \left|\underline{j}\right\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} (-1)^{j \cdot k} \left|\underline{k}\right\rangle$
- $H^{\otimes n} \left|\underline{0}\right\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} \left|\underline{k}\right\rangle$

Simon Quantum Algorithm

Hadamard Transform

- $H^{\otimes n}\left|\underline{j}\right\rangle = \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1}(-1)^{j\cdot k}\left|\underline{k}\right\rangle$
- $H^{\otimes n}\left|\underline{0}\right\rangle = \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1}\left|\underline{k}\right\rangle$

Simon's algorithm

Start with $2n$ qubits:	$\left|\underline{0}\right\rangle\left|\underline{0}\right\rangle$
Apply $H^{\otimes n}$	$\sum_x \left|\underline{x}\right\rangle\left|\underline{0}\right\rangle$
Apply O_f	$\sum_x \left|\underline{x}\right\rangle\left|\underline{f(x)}\right\rangle$
Measure the second register	$\left|\underline{x_0}\right\rangle + \left|\underline{x_0 + s}\right\rangle$
Apply $H^{\otimes n}$	$\sum_y((-1)^{x_0\cdot y} + (-1)^{(x_0\oplus s)\cdot y})\left|\underline{y}\right\rangle$
	$= \sum_y(-1)^{x_0\cdot y}\cdot(1+(-1)^{s\cdot y})\left|\underline{y}\right\rangle$

Measure y such that $1+(-1)^{s\cdot y} \neq 0$ iff $s\cdot y = 0$

Simon Quantum Algorithm

Hadamard Transform

- $H^{\otimes n} \left|\underline{j}\right\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} (-1)^{j \cdot k} \left|\underline{k}\right\rangle$
- $H^{\otimes n} \left|\underline{0}\right\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} \left|\underline{k}\right\rangle$

Simon's algorithm

Start with $2n$ qubits:	$\left|\underline{0}\right\rangle \left|\underline{0}\right\rangle$
Apply $H^{\otimes n}$	$\sum_x \left|\underline{x}\right\rangle \left|\underline{0}\right\rangle$
Apply O_f	$\sum_x \left|\underline{x}\right\rangle \left|\underline{f(x)}\right\rangle$
Measure the second register	$\left|\underline{x_0}\right\rangle + \left|\underline{x_0 + s}\right\rangle$
Apply $H^{\otimes n}$	$\sum_y ((-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus s) \cdot y}) \left|\underline{y}\right\rangle$
	$= \sum_y (-1)^{x_0 \cdot y} \cdot (1 + (-1)^{s \cdot y}) \left|\underline{y}\right\rangle$

Measure y such that $1 + (-1)^{s \cdot y} \neq 0$ iff $s \cdot y = 0$

Post-processing

- With $n - 1$ values $y_1, \ldots, y_{n-1}$ independent vectors, we obtain a linear system to recover s

Shor Algorithm

Arithmetic

- $\mathbb{Z}/N\mathbb{Z}$ is not an integral domain: $N = 15$, $5 \times 3 = 0 \bmod 15$
- $(\mathbb{Z}/N\mathbb{Z})^*$ multiplicative group of invertible elements, not cyclic !
- order of a: smallest positive integer r s.t. $a^r = 1 \bmod N$
- $r|\varphi(N)$ Lagrange Theorem in the group $(\mathbb{Z}/N\mathbb{Z})^*$
- r is the smallest period of the function $f : k \mapsto a^k \bmod N$

Arithmetic

- $(\mathbb{Z}/N\mathbb{Z})^*$ multiplicative group of invertible elements, not cyclic !
- order of a: smallest positive integer r s.t. $a^r = 1 \bmod N$
- $r|\varphi(N)$ Lagrange Theorem in the group $(\mathbb{Z}/N\mathbb{Z})^*$
- r is the smallest period of the function $f : k \mapsto a^k \bmod N$

Assumptions

1. Assumption 1: $\mathrm{ord}(a) = r$ is even with proba. $1/2$
2. Fact: $(a^{r/2} - 1)(a^{r/2} + 1) = 0 \bmod N$

Arithmetic

- $(\mathbb{Z}/N\mathbb{Z})^*$ multiplicative group of invertible elements, not cyclic !
- order of a: smallest positive integer r s.t. $a^r = 1 \bmod N$
- $r|\varphi(N)$ Lagrange Theorem in the group $(\mathbb{Z}/N\mathbb{Z})^*$
- r is the smallest period of the function $f : k \mapsto a^k \bmod N$

Assumptions

1. Assumption 1: $\mathrm{ord}(a) = r$ is even with proba. $1/2$
2. Fact: $(a^{r/2} - 1)(a^{r/2} + 1) = 0 \bmod N$
3. Assumption 2: $a^{r/2} + 1$ is not divisible by N for many a's (CRT)
4. Under Assumption 1 and 2: $d = \gcd(a^{r/2} - 1, N)$ and $d' = \gcd(a^{r/2} + 1, N)$ are non-trivial factors of N

Arithmetic

- $(\mathbb{Z}/N\mathbb{Z})^*$ multiplicative group of invertible elements, not cyclic !
- order of a: smallest positive integer r s.t. $a^r = 1 \bmod N$
- $r|\varphi(N)$ Lagrange Theorem in the group $(\mathbb{Z}/N\mathbb{Z})^*$
- r is the smallest period of the function $f : k \mapsto a^k \bmod N$

Assumptions

1. Assumption 1: $\mathrm{ord}(a) = r$ is even with proba. $1/2$
2. Fact: $(a^{r/2} - 1)(a^{r/2} + 1) = 0 \bmod N$
3. Assumption 2: $a^{r/2} + 1$ is not divisible by N for many a's (CRT)
4. Under Assumption 1 and 2: $d = \gcd(a^{r/2} - 1, N)$ and $d' = \gcd(a^{r/2} + 1, N)$ are non-trivial factors of N

a=2	$(a, N) = 1$	$r = 4, 2^4 = 16 = 1 \bmod 15$	$(2^{4/2} - 1, 15) = 3$
a=3	no		
a=11	$(a, N) = 1$	$r = 2, 11^2 = 121 = 1 \bmod 15$	$(11^{2/2} - 1, 15) = 5$

Order and Oracle

- order of a: smallest positive integer r s.t. $a^r = 1 \bmod N$
- $r|\varphi(N)$ Lagrange Theorem in the group $(\mathbb{Z}/N\mathbb{Z})^*$
- r is the smallest period of the function $f : k \mapsto a^k \bmod N$
- Oracle $F : (k, 0) \mapsto (k, a^k \bmod N)$
- E.g. $N = 15$ and $a = 2$, $r = 4$

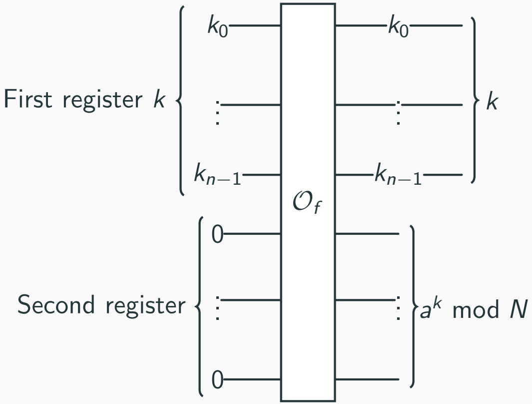
Order and Oracle

- order of a: smallest positive integer r s.t. $a^r = 1 \bmod N$
- $r|\varphi(N)$ Lagrange Theorem in the group $(\mathbb{Z}/N\mathbb{Z})^*$
- r is the smallest period of the function $f : k \mapsto a^k \bmod N$
- Oracle $F : (k, 0) \mapsto (k, a^k \bmod N)$
- E.g. $N = 15$ and $a = 2$, $r = 4$

$$
\begin{array}{llll}
(0,0) \overset{F}{\mapsto} (0,1) & (4,0) \overset{F}{\mapsto} (4,1) & (8,0) \overset{F}{\mapsto} (8,1) & (12,0) \overset{F}{\mapsto} (12,1) \\
(1,0) \overset{F}{\mapsto} (1,2) & (5,0) \overset{F}{\mapsto} (5,2) & (9,0) \overset{F}{\mapsto} (9,2) & (13,0) \overset{F}{\mapsto} (13,2) \\
(2,0) \overset{F}{\mapsto} (2,4) & (6,0) \overset{F}{\mapsto} (6,4) & (10,0) \overset{F}{\mapsto} (10,4) & (14,0) \overset{F}{\mapsto} (14,4) \\
(3,0) \overset{F}{\mapsto} (3,8) & (7,0) \overset{F}{\mapsto} (7,8) & (11,0) \overset{F}{\mapsto} (11,8) & (15,0) \overset{F}{\mapsto} (15,8)
\end{array}
$$

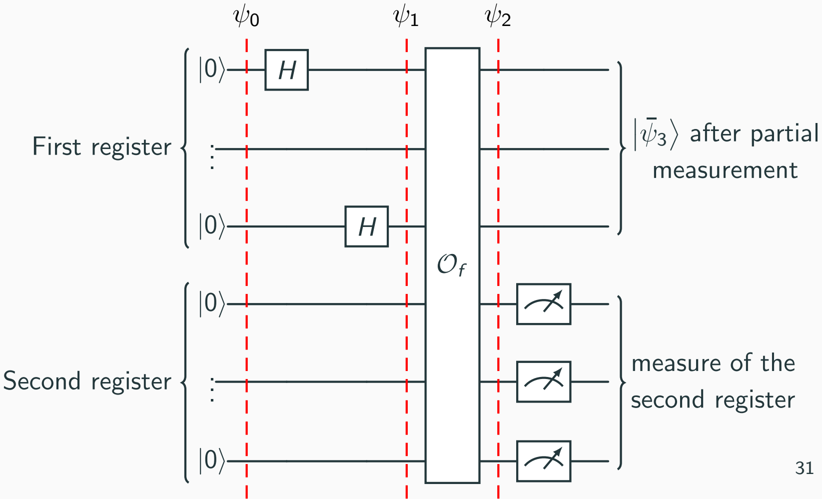
Oracle Circuit $2^n \geq N$

The oracle is composed of 2 registers: the first receives the integer k in binary with n bits, and the second, 0 on n bits. We write $|\underline{k}\rangle$ the register containing k written in binary. For instance, $|\underline{0}\rangle = |0.\dots 0\rangle$ with n bits. The initial state is $|\underline{k}\rangle \otimes |\underline{0}\rangle$.

- 

Starting the Circuit $2^n \geq N$

- Initialization: $|\psi_0\rangle = |\underline{0}\rangle \otimes |\underline{0}\rangle$.
- Hadamard: $|\psi_1\rangle = H^{\otimes n}(|\underline{0}\rangle) \otimes |\underline{0}\rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |\underline{k}\rangle\right) \otimes |\underline{0}\rangle$
- Oracle: $|\psi_2\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} |\underline{k}\rangle \otimes \left|\underline{a^k}\right\rangle$



Using the period to rewrite $|\psi_2\rangle$

- Assumption 3: $\mathrm{ord}(a) = r | 2^n$. This assumption is not true, and can be removed (see later)
- Under Assumption 3: $k = \alpha r + \beta$ with $0 \leq \beta < r$ and $0 \leq \alpha < 2^n/r$,

$$|\psi_2\rangle = \sum_{k=0}^{2^n-1} |\underline{k}\rangle \otimes |\underline{a^k}\rangle = \sum_{\beta=0}^{r-1} \left(\sum_{\alpha=0}^{2^n/r-1} |\underline{\alpha r + \beta}\rangle \right) \otimes |a^\beta\rangle$$

Using the period to rewrite $|\psi_2\rangle$

- Assumption 3: $\mathrm{ord}(a) = r | 2^n$. This assumption is not true, and can be removed (see later)
- Under Assumption 3: $k = \alpha r + \beta$ with $0 \leq \beta < r$ and $0 \leq \alpha < 2^n/r$,

$$|\psi_2\rangle = \sum_{k=0}^{2^n-1} |\underline{k}\rangle \otimes |\underline{a^k}\rangle = \sum_{\beta=0}^{r-1} \left(\sum_{\alpha=0}^{2^n/r-1} |\underline{\alpha r + \beta}\rangle \right) \otimes |a^{\beta}\rangle$$

- If we measure the second register, we get for a fixed β_0,

$$|\psi_3\rangle = \sum_{\alpha=0}^{2^n/r-1} |\alpha r + \beta_0\rangle \otimes |a^{\beta_0}\rangle$$

Using the period to rewrite $|\psi_2\rangle$

- Assumption 3: $\mathrm{ord}(a) = r | 2^n$. This assumption is not true, and can be removed (see later)
- Under Assumption 3: $k = \alpha r + \beta$ with $0 \leq \beta < r$ and $0 \leq \alpha < 2^n/r$,

$$|\psi_2\rangle = \sum_{k=0}^{2^n-1} |\underline{k}\rangle \otimes |\underline{a^k}\rangle = \sum_{\beta=0}^{r-1} \left(\sum_{\alpha=0}^{2^n/r-1} |\underline{\alpha r + \beta}\rangle \right) \otimes |a^\beta\rangle$$

- If we measure the second register, we get for a fixed β_0,

$$|\psi_3\rangle = \sum_{\alpha=0}^{2^n/r-1} |\alpha r + \beta_0\rangle \otimes |a^{\beta_0}\rangle$$

- Assume we measure the first register, $|\alpha_0 r + \beta_0\rangle$ for fixed α_0 and β_0
- If we redo the computation, we will not the same β_0,
- We cannot do many measures of the first register ...

Example $N = 15$, $a = 2$

- $|\psi_0\rangle = |\underline{0}\rangle \otimes |\underline{0}\rangle$
- Hadamard Transform: $|\psi_1\rangle = (|\underline{0}\rangle + |\underline{1}\rangle + \ldots + |\underline{15}\rangle) \otimes |\underline{0}\rangle$
- Oracle: $|\psi_2\rangle = |\underline{0}\rangle . \left|\underline{a^0}\right\rangle + |\underline{1}\rangle . \left|\underline{a^1}\right\rangle + \ldots + |\underline{15}\rangle . \left|\underline{a^{15}}\right\rangle$

Example $N = 15$, $a = 2$

- $|\psi_0\rangle = |\underline{0}\rangle \otimes |\underline{0}\rangle$
- Hadamard Transform: $|\psi_1\rangle = (|\underline{0}\rangle + |\underline{1}\rangle + \ldots + |\underline{15}\rangle) \otimes |\underline{0}\rangle$
- Oracle: $|\psi_2\rangle = |\underline{0}\rangle . \left|\underline{a^0}\right\rangle + |\underline{1}\rangle . \left|\underline{a^1}\right\rangle + \ldots + |\underline{15}\rangle . \left|\underline{a^{15}}\right\rangle$
- Since $r = 4 | 2^4 = 16$, the values form a rectangular table

$$\begin{aligned}
|\psi_2\rangle = &\left(|\underline{0}\rangle + |\underline{4}\rangle + |\underline{8}\rangle + |\underline{12}\rangle \right) . |\underline{1}\rangle + \\
&\left(|\underline{1}\rangle + |\underline{5}\rangle + |\underline{9}\rangle + |\underline{13}\rangle \right) . |\underline{2}\rangle + \\
&\left(|\underline{2}\rangle + |\underline{6}\rangle + |\underline{10}\rangle + |\underline{14}\rangle \right) . |\underline{4}\rangle + \\
&\left(|\underline{3}\rangle + |\underline{7}\rangle + |\underline{11}\rangle + |\underline{15}\rangle \right) . |\underline{8}\rangle
\end{aligned}$$

- If we measure the second register, $|\underline{4}\rangle$, the first register is

$$\left|\widetilde{\psi_3}\right\rangle = |\underline{2}\rangle + |\underline{6}\rangle + |\underline{10}\rangle + |\underline{14}\rangle$$

- They are separated by the period $r = 4$, but how can we recover r ?

Discrete Fourier Transform

Complex numbers

- $$1 + z + \ldots + z^{n-1} = \begin{cases} n & \text{if } z = 1 \\ \frac{1-z^n}{1-z} & \text{otherwise.} \end{cases}$$
- Crucial Lemma: $n > 0, j \in \mathbb{Z}$,
$$\frac{1}{n}\sum_{k=0}^{n-1} e^{2i\pi\frac{kj}{n}} = \begin{cases} 1 & \text{if } \frac{j}{n} \text{ is an integer} \\ 0 & \text{otherwise.} \end{cases}$$

Discrete Fourier Transform

Complex numbers

- $$1 + z + \ldots + z^{n-1} = \begin{cases} n & \text{if } z = 1 \\ \frac{1-z^n}{1-z} & \text{otherwise.} \end{cases}$$
- Crucial Lemma: $n > 0, j \in \mathbb{Z}$,
$$\frac{1}{n} \sum_{k=0}^{n-1} e^{2i\pi \frac{kj}{n}} = \begin{cases} 1 & \text{if } \frac{j}{n} \text{ is an integer} \\ 0 & \text{otherwise.} \end{cases}$$

Discrete Fourier Transform and Inverse

$$\widehat{F} \left|\underline{k}\right\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{2i\pi \frac{kj}{2^n}} \left|\underline{j}\right\rangle \text{ and } \widehat{F}^{-1} \left|\underline{k}\right\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{-2i\pi \frac{kj}{2^n}} \left|\underline{j}\right\rangle$$

Discrete Fourier Transform

Complex numbers

- $$1 + z + \ldots + z^{n-1} = \begin{cases} n & \text{if } z = 1 \\ \frac{1-z^n}{1-z} & \text{otherwise.} \end{cases}$$
- Crucial Lemma: $n > 0, j \in \mathbb{Z}$,

$$\frac{1}{n}\sum_{k=0}^{n-1} e^{2i\pi\frac{kj}{n}} = \begin{cases} 1 & \text{if } \frac{j}{n} \text{ is an integer} \\ 0 & \text{otherwise.} \end{cases}$$

Discrete Fourier Transform and Inverse

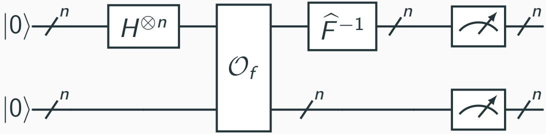
$$\widehat{F}\left|\underline{k}\right\rangle = \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1} e^{2i\pi\frac{kj}{2^n}}\left|\underline{j}\right\rangle \text{ and } \widehat{F}^{-1}\left|\underline{k}\right\rangle = \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1} e^{-2i\pi\frac{kj}{2^n}}\left|\underline{j}\right\rangle$$

The Discrete Fourier Transform is Linear and Unitary

$$\text{If } \left|\psi\right\rangle = \sum_{k=0}^{2^n-1}\alpha_k\left|\underline{k}\right\rangle, \text{ then } \widehat{F}\left|\psi\right\rangle = \sum_{k=0}^{2^n-1}\alpha_k\widehat{F}\left|\underline{k}\right\rangle$$

Shor Circuit

- Initialization: $|\psi_0\rangle = |\underline{0}\rangle \otimes |\underline{0}\rangle$.
- Hadamard: $|\psi_1\rangle = H^{\otimes n}(|\underline{0}\rangle) \otimes |\underline{0}\rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |\underline{k}\rangle\right) \otimes |\underline{0}\rangle$
- Oracle: $|\psi_2\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} |\underline{k}\rangle \otimes \left|\underline{a^k}\right\rangle$



- Measure of the first register: $\left|\frac{2^n \ell}{r}\right\rangle$
- Allows (often) to get r (or a factor of r)

Computation

- After measuring the second register $\left|\bar{\psi}_3\right\rangle = \sum_{\alpha=0}^{2^n/r-1} \left|\underline{\alpha r + \beta_0}\right\rangle$

Computation

- After measuring the second register $\left|\bar{\psi}_3\right\rangle = \sum_{\alpha=0}^{2^n/r-1} \left|\underline{\alpha r + \beta_0}\right\rangle$
- Action of $\widehat{F}^{-1}$:

$$\left|\bar{\psi}_4\right\rangle = \widehat{F}^{-1} \left|\hat{\psi}_3\right\rangle = \sum_{\alpha=0}^{2^n/r-1} \widehat{F}^{-1} \left|\underline{\alpha r + \beta_0}\right\rangle$$

$$= \sum_{\alpha} \sum_{j=0}^{2^n-1} e^{-\frac{2i\pi(\alpha r+\beta_0)j}{2^n}} \left|\underline{j}\right\rangle = \sum_{j} \overbrace{\left(\sum_{\alpha} e^{-2i\pi \frac{\alpha j}{2^n/r}} \right)}^{0 \text{ or } 1} e^{-2i\pi \frac{\beta_0 j}{2^n}} \left|\underline{j}\right\rangle$$

$$= \sum_{j \text{ with } j/(2^n/r) \text{ integer}} e^{-2i\pi \frac{\beta_0 j}{2^n}} \left|j\right\rangle = \sum_{\ell=0}^{r-1} e^{-2i\pi \beta_0 \frac{\ell}{r}} \left|\frac{2^n \ell}{r}\right\rangle$$

Computation

- After measuring the second register $\left|\bar{\psi}_3\right\rangle = \sum_{\alpha=0}^{2^n/r-1} \left|\underline{\alpha r + \beta_0}\right\rangle$
- Action of $\widehat{F}^{-1}$:

$$\left|\bar{\psi}_4\right\rangle = \widehat{F}^{-1}\left|\hat{\psi}_3\right\rangle = \sum_{\alpha=0}^{2^n/r-1} \widehat{F}^{-1}\left|\underline{\alpha r + \beta_0}\right\rangle$$

$$= \sum_{\alpha}\sum_{j=0}^{2^n-1} e^{-\frac{2i\pi(\alpha r+\beta_0)j}{2^n}}\left|\underline{j}\right\rangle = \sum_{j} \overbrace{\left(\sum_{\alpha} e^{-2i\pi\frac{\alpha j}{2^n/r}}\right)}^{0 \text{ or } 1} e^{-2i\pi\frac{\beta_0 j}{2^n}}\left|\underline{j}\right\rangle$$

$$= \sum_{j \text{ with } j/(2^n/r) \text{ integer}} e^{-2i\pi\frac{\beta_0 j}{2^n}}\left|j\right\rangle = \sum_{\ell=0}^{r-1} e^{-2i\pi\beta_0\frac{\ell}{r}}\left|\frac{2^n\ell}{r}\right\rangle$$

- Measure the first register: $\left|\frac{2^n\ell}{r}\right\rangle$, for $\ell \in \{0, 1, \dots, r-1\}$
- We get $m = \frac{2^n\ell}{r}$ for one of the states $\left|\frac{2^n\ell}{r}\right\rangle$

Measure the first register

$m = \frac{2^n \ell}{r}$ **integer with** n **known and** ℓ **unknown**

- Divide m by 2^n to obtain the rational $x = \frac{m}{2^n} = \frac{\ell}{r}$
- If $x \in \mathbb{Z}$, we get no information on r, and we redo the quantum circuit
- If $\gcd(\ell, r) = 1$, then $\frac{\ell}{r}$ is irreducible and we get r.
- If $\gcd(\ell, r) \neq 1$, then $x = \frac{m}{2^n} = \frac{\ell'}{r'} = \frac{\ell}{r}$ and we get r' a factor of r. We redo the computation with $a' = a^{r'}$ which is of period r/r'.

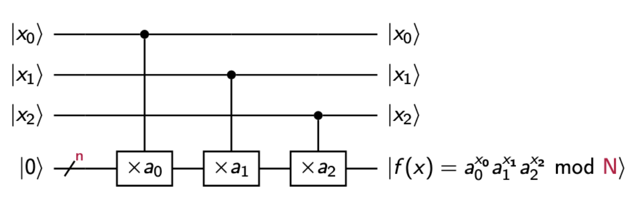
Implementation of the oracle

Reduce exponentiation to controlled multi-product modulo N:

$$f(x) = a^x = \prod_i \left(a^{2^i}\right)^{x_i} = \prod_i \left(a_i\right)^{x_i} \bmod N, \text{ where } a_i = a^{2^i} \bmod N$$

The constants a_i are precomputed:

- Asymptotic best: $O(n \times (n \log n))$ operations
- Typical: $O(n \times (n^2))$ operations



Shor for any even order

Up to now..

- If $r|2^n$, measuring $\left|\frac{2^n\ell}{r}\right\rangle$ gives an integer $m = \frac{2^n\ell}{r}$ and $x = \frac{m}{2^n} = \frac{\ell}{r}$ which allows to recover r or a factor
- As $r|2^n$, m is a multiple of $\frac{2^n}{r}$ and x is a multiple of $\frac{1}{r}$

Shor for any even order

Up to now..

- If $r|2^n$, measuring $\left|\frac{2^n\ell}{r}\right\rangle$ gives an integer $m = \frac{2^n\ell}{r}$ and $x = \frac{m}{2^n} = \frac{\ell}{r}$ which allows to recover r or a factor
- As $r|2^n$, m is a multiple of $\frac{2^n}{r}$ and x is a multiple of $\frac{1}{r}$

Now...

- If $r \nmid 2^n$, the measurement gives an integer m which is close to $\frac{2^n\ell}{r}$, but $\frac{2^n\ell}{r}$ is not any more an integer ...
- The rational $x = \frac{m}{2^n}$ is close to a multiple of $\frac{1}{r}$ but not an exact multiple...

Continued Fractions

Definition

- $a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\dots + \frac{1}{a_n}}}}$, noted $[a_0, a_1, \dots, a_n]$
- E.g., $[5, 2, 1, 4] = 5 + \frac{1}{2 + \frac{1}{1 + \frac{1}{4}}} = 5.3571428\dots$
- $[5] = 5, [5, 2] = \frac{11}{2} = 5.5, [5, 2, 1] = \frac{16}{3} = 5.33\dots$

Continued Fractions

Definition

- $a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\dots + \frac{1}{a_n}}}}$, noted $[a_0, a_1, \dots, a_n]$
- E.g., $[5, 2, 1, 4] = 5 + \frac{1}{2 + \frac{1}{1 + \frac{1}{4}}} = 5.3571428\dots$
- $[5] = 5, [5, 2] = \frac{11}{2} = 5.5, [5, 2, 1] = \frac{16}{3} = 5.33\dots$

Good Approximation by continued fractions

- $\pi = 3.14159\dots \approx \frac{314}{100}$ (denominator is large)
- $\frac{314}{100} = 3 + \frac{14}{100} = 3 + \frac{1}{\frac{100}{14}} = 3 + \frac{1}{7 + \frac{2}{14}} = 3 + \frac{1}{7 + \frac{1}{7}} = [3, 7, 7]$
- $[3, 7] = 3 + \frac{1}{7} = \frac{22}{7} = 3.1428$
- $[3, 7, 15, 1] = \frac{355}{113} = 3.14159292\dots$ (same order with 6 exact values instead of 2)

Example Shor with $N = 21$

- $N = 21$, $a = 2$, $2^n = 512 = 2^9$
- Circuit outputs $|427\rangle$, so $x = \frac{427}{512}$
- $\frac{427}{512} \approx \frac{4}{5}$ so order 5 ??
- $\frac{427}{512} = [0, 1, 5, 42, 2]$ and $[0, 1] = 1, [0, 1, 5] = \frac{5}{6}, [0, 1, 5, 42] = \frac{211}{253}$
- We keep the best fraction whose denominator is $\leq N$ and it gives r or a fraction of r

Example Shor with $N = 21$

- $N = 21$, $a = 2$, $2^n = 512 = 2^9$
- Circuit outputs $|427\rangle$, so $x = \frac{427}{512}$
- $\frac{427}{512} \approx \frac{4}{5}$ so order 5 ??
- $\frac{427}{512} = [0, 1, 5, 42, 2]$ and $[0, 1] = 1, [0, 1, 5] = \frac{5}{6}, [0, 1, 5, 42] = \frac{211}{253}$
- We keep the best fraction whose denominator is $\leq N$ and it gives r or a fraction of r

Shor algorithm with arbitrary order

- $N = 21$, $a = 2$, $2^n = 512 = 2^9 \geq N^2$
- $|\psi_0\rangle = |\underline{0}\rangle \otimes |\underline{0}\rangle$
- $|\psi_1\rangle = \sum_{k=0}^{r-1} |\underline{k}\rangle \otimes |\underline{0}\rangle$
- $|\psi_2\rangle = \sum_{k=0}^{r-1} |\underline{k}\rangle \otimes |\underline{a^k \bmod N}\rangle$
- $r = 6$ and $\frac{2^n \ell}{r} \notin \mathbb{Z}$

Example

The first two lines have 86 terms and 85 in the others

- The state $|\psi_2\rangle$ is not rectangular:

$$
\begin{aligned}
|\psi_2\rangle = &\frac{1}{\sqrt{512}}(|\underline{0}\rangle + |\underline{6}\rangle + \ldots + |\underline{504}\rangle + |\underline{510}\rangle)\,|\underline{1}\rangle \\
&+ \frac{1}{\sqrt{512}}(|\underline{1}\rangle + |\underline{7}\rangle + \ldots + |\underline{505}\rangle + |\underline{511}\rangle)\,|\underline{2}\rangle \\
&+ \frac{1}{\sqrt{512}}(|\underline{2}\rangle + |\underline{8}\rangle + \ldots + |\underline{506}\rangle)\,|\underline{4}\rangle \\
&+ \ldots \\
&+ \frac{1}{\sqrt{512}}(|\underline{5}\rangle + |\underline{11}\rangle + \ldots + |\underline{509}\rangle)\,|\underline{11}\rangle
\end{aligned}
$$

Example

The first two lines have 86 terms and 85 in the others

- The state $|\psi_2\rangle$ is not rectangular:

$$
\begin{aligned}
|\psi_2\rangle = &\frac{1}{\sqrt{512}}\left(|\underline{0}\rangle + |\underline{6}\rangle + \ldots + |\underline{504}\rangle + |\underline{510}\rangle\right)|\underline{1}\rangle \\
&+ \frac{1}{\sqrt{512}}\left(|\underline{1}\rangle + |\underline{7}\rangle + \ldots + |\underline{505}\rangle + |\underline{511}\rangle\right)|\underline{2}\rangle \\
&+ \frac{1}{\sqrt{512}}\left(|\underline{2}\rangle + |\underline{8}\rangle + \ldots + |\underline{506}\rangle\right)|\underline{4}\rangle \\
&+ \ldots \\
&+ \frac{1}{\sqrt{512}}\left(|\underline{5}\rangle + |\underline{11}\rangle + \ldots + |\underline{509}\rangle\right)|\underline{11}\rangle
\end{aligned}
$$

- measure the second register $|2\rangle$: $|\psi_3\rangle = |\underline{1}\rangle + |\underline{7}\rangle + \ldots + |\underline{511}\rangle$
- $|\psi_4\rangle = \hat{F}^{-1}|\psi_3\rangle = \sum_{\alpha=0}^{85} \hat{F}^{-1}|\underline{6\alpha+1}\rangle$
- $|\psi_4\rangle = \sum_{j=0}^{511}\left(\sum_{\alpha=0}^{85} e^{-2i\pi\frac{6\alpha j}{512}}\right)e^{-2i\pi\frac{j}{512}}|\underline{j}\rangle$

Example Shor with arbitrary order

$|\psi_4\rangle = \frac{1}{\sqrt{512}} \sum_{j=0}^{511} \left(\frac{1}{\sqrt{86}} \sum_{\alpha=0}^{85} e^{-2i\pi \frac{6\alpha j}{512}} \right) e^{-2i\pi \frac{j}{512}} \left| \underline{j} \right\rangle$

Now, $\Sigma(j) = \frac{1}{\sqrt{86}} \sum_{\alpha=0}^{85} e^{-2i\pi \frac{6\alpha j}{512}}$ does not take only 0 /1 values.

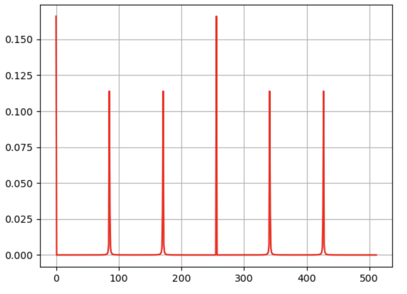
Example Shor with arbitrary order

$$|\psi_4\rangle = \frac{1}{\sqrt{512}} \sum_{j=0}^{511} \left(\frac{1}{\sqrt{86}} \sum_{\alpha=0}^{85} e^{-2i\pi \frac{6\alpha j}{512}} \right) e^{-2i\pi \frac{j}{512}} \left|\underline{j}\right\rangle$$

Now, $\Sigma(j) = \frac{1}{\sqrt{86}} \sum_{\alpha=0}^{85} e^{-2i\pi \frac{6\alpha j}{512}}$ does not take only 0 /1 values.

If we measure the first register, we get $|j\rangle$ with probability $|\Sigma(j)|^2$.

The proba. are ≈ 0, except when $j \approx \frac{2^n \ell}{r}$: for $\ell = 5$, $\frac{512 \times 5}{6} = 426.66$.



j	p_j
422	0.00062...
423	0.00099...
424	0.00186...
425	0.00469...
426	0.02888...
427	**0.11389...**
428	0.00702...
429	0.00226...
430	0.00109...
431	0.00063...

Hardy-Wright Theorem

Theorem
Let $x \in \mathbb{R}$ and a rational $\frac{p}{q}$ such that

$$\left|x - \frac{p}{q}\right| < \frac{1}{2q^2}.$$

Then, $\frac{p}{q}$ is obtained as one of the continued fractions of x.

Hardy-Wright Theorem

Theorem
Let $x \in \mathbb{R}$ and a rational $\frac{p}{q}$ such that

$$\left|x - \frac{p}{q}\right| < \frac{1}{2q^2}.$$

Then, $\frac{p}{q}$ is obtained as one of the continued fractions of x.

Let m the closest integer to $\frac{2^n \ell}{r}$. So, $|m - \frac{2^n \ell}{r}| < \frac{1}{2}$.

If $x = \frac{m}{2^n}$, we get $|x - \frac{\ell}{r}| < \frac{1}{2^{n+1}}$.

As we set $2^n \geq N^2 \geq r^2$, $|x - \frac{\ell}{r}| < \frac{1}{2r^2}$.

Using Theorem, we obtain $\frac{\ell}{r}$ as one of the continued fractions of x.

Between 1996 and 2020

Generalization

- HSP (Hidden Subgroup Problem): Let G a group and H a subgroup. The function f is constant on each coset of H, find H

Between 1996 and 2020

Generalization

- HSP (Hidden Subgroup Problem): Let G a group and H a subgroup. The function f is constant on each coset of H, find H
- Shor and Simon algorithms: special case of HSP

Between 1996 and 2020

Generalization

- HSP (Hidden Subgroup Problem): Let G a group and H a subgroup. The function f is constant on each coset of H, find H
- Shor and Simon algorithms: special case of HSP
- Kitaev: any Abelian Group G

Between 1996 and 2020

Generalization

- HSP (Hidden Subgroup Problem): Let G a group and H a subgroup. The function f is constant on each coset of H, find H
- Shor and Simon algorithms: special case of HSP
- Kitaev: any Abelian Group G
- Non-abelian: Kuperberg subexponential algo. for Dihedral HSP

Between 1996 and 2020

Generalization

- HSP (Hidden Subgroup Problem): Let G a group and H a subgroup. The function f is constant on each coset of H, find H
- Shor and Simon algorithms: special case of HSP
- Kitaev: any Abelian Group G
- Non-abelian: Kuperberg subexponential algo. for Dihedral HSP
- LWE (learning with errors problems) can be reduced to (stronger version) Dihedral HSP (with errors)

Generalization

- HSP (Hidden Subgroup Problem): Let G a group and H a subgroup. The function f is constant on each coset of H, find H
- Shor and Simon algorithms: special case of HSP
- Kitaev: any Abelian Group G
- Non-abelian: Kuperberg subexponential algo. for Dihedral HSP
- LWE (learning with errors problems) can be reduced to (stronger version) Dihedral HSP (with errors)

How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits

Craig Gidney[1] and Martin Ekerå[2]

New recent results on factorization

- Shor algorithm: $3n$ qubits and $O(n^2)$ gates

New recent results on factorization

- Shor algorithm: $3n$ qubits and $O(n^2)$ gates
- Regev algorithm: $O(n^{3/2})$ qubits and $O(n^{3/2})$ gates, runs $n^{1/2}$

New recent results on factorization

- Shor algorithm: $3n$ qubits and $O(n^2)$ gates
- Regev algorithm: $O(n^{3/2})$ qubits and $O(n^{3/2})$ gates, runs $n^{1/2}$
- Pilatte removes mathematical assumptions in Regev algorithm

New recent results on factorization

- Shor algorithm: $3n$ qubits and $O(n^2)$ gates
- Regev algorithm: $O(n^{3/2})$ qubits and $O(n^{3/2})$ gates, runs $n^{1/2}$
- Pilatte removes mathematical assumptions in Regev algorithm
- Ragavan-Vaikuntanathan (C'24): $10n$ qubits and $O(n^{3/2})$ gates, runs $n^{1/2}$

New recent results on factorization

- Shor algorithm: $3n$ qubits and $O(n^2)$ gates
- Regev algorithm: $O(n^{3/2})$ qubits and $O(n^{3/2})$ gates, runs $n^{1/2}$
- Pilatte removes mathematical assumptions in Regev algorithm
- Ragavan-Vaikuntanathan (C'24): $10n$ qubits and $O(n^{3/2})$ gates, runs $n^{1/2}$
- $n/2 + o(n)$ qubits and $O(n^2)$ gates, runs constants [CFS25]

Reducing the number of qubits

New algorithm[1]

- Factoring RSA moduli using $n/2 + o(n)$ qubits and $O(n^3)$ gates
- For RSA-2048: ≤ 1700 qubits and $\leq 60 \times 2^{36}$ Toffoli gates (60 runs)
- Based on a completely classical arithmetic circuit

[1] CFS, CRYPTO 2025, "Reducing the Number of Qubits in Quantum Factoring"

New algorithm[1]

- Factoring RSA moduli using $n/2 + o(n)$ qubits and $O(n^3)$ gates
- For RSA-2048: ≤ 1700 qubits and $\leq 60 \times 2^{36}$ Toffoli gates (60 runs)
- Based on a completely classical arithmetic circuit
- Gidney reduces: qubits down to 1399 logical qubits by computing the MSB rather than the LSB, 2^{32} Toffoli gates as previous counting and 9.2 runs, and update estimates at the physical level

Gidney latest result

How to factor 2048 bit RSA integers with less than a million noisy qubits

Craig Gidney

Google Quantum AI, Santa Barbara, California 93117, USA

June 9, 2025

[1] CFS, CRYPTO 2025, “Reducing the Number of Qubits in Quantum Factoring”

Discrete logarithm and RSA special case

Find d s.t. $a = g^d$:

[2]Ekerå, Håstad, "Quantum algorithms for computing short discrete logarithms and factoring RSA integers, PQCrypto 2017"

Discrete logarithm and RSA special case

Find d s.t. $a = g^d$: $f(x, y) := g^x a^{-y} = g^{x-dy} \bmod N$

- Also a hidden period problem: $f(x + d, y + 1) = f(x, y)$
- Also reduces to controlled multi-product

[2] Ekerå, Håstad, "Quantum algorithms for computing short discrete logarithms and factoring RSA integers, PQCrypto 2017"

Discrete logarithm and RSA special case

Find d s.t. $a = g^d$: $f(x, y) := g^x a^{-y} = g^{x-dy} \bmod N$

- Also a hidden period problem: $f(x + d, y + 1) = f(x, y)$
- Also reduces to controlled multi-product

Ekerå & Håstad method[2]:

- Reduce RSA factorisation ($N = pq$) to small DLOG of size $n/2$: if we recover $p + q$, we can factor N
- Use an input register of size $n/2 + (n/2)/s$ for some s
- $\approx s + 1$ measurements to find d via an efficient lattice-based post-processing. Typically $s = O(\log n)$.

Space is reduced to: $n/2 +$ **workspace**

[2] Ekerå, Håstad, "Quantum algorithms for computing short discrete logarithms and factoring RSA integers, PQCrypto 2017"

Variant Shor's algorithm

Ideas

- Once $p + q$ is known, using $N = pq$, recover p is easy

[3] "Quantum period-finding is compression robust"

Variant Shor's algorithm

Ideas

- Once $p + q$ is known, using $N = pq$, recover p is easy
- $G = \langle g \rangle$ a cyclic subgroup of $(\mathbb{Z}/N\mathbb{Z})^*$ of order $> (p + q - 2)/2$
- Compute $x = g^{(N-1)/2} = g^{(p+q-2)/2} \bmod N$ since $(N - \varphi(N) - 1)/2 = (p + q - 2)/2$ as $\varphi(N) = N - p - q + 1$
- Compute short discrete logarithm $d = (p + q - 2)/2$ from g and x

[3] "Quantum period-finding is compression robust"

Variant Shor's algorithm

Ideas

- Once $p+q$ is known, using $N = pq$, recover p is easy
- $G = \langle g \rangle$ a cyclic subgroup of $(\mathbb{Z}/N\mathbb{Z})^*$ of order $> (p+q-2)/2$
- Compute $x = g^{(N-1)/2} = g^{(p+q-2)/2} \bmod N$ since $(N - \varphi(N) - 1)/2 = (p+q-2)/2$ as $\varphi(N) = N - p - q + 1$
- Compute short discrete logarithm $d = (p+q-2)/2$ from g and x
- Get many pairs (j,k) s.t. k is the ℓ most significant bits of $dj \bmod 2^m$: Hidden Number Problem (HNP)

[3] "Quantum period-finding is compression robust"

Variant Shor's algorithm

Ideas

- Once $p + q$ is known, using $N = pq$, recover p is easy
- $G = \langle g \rangle$ a cyclic subgroup of $(\mathbb{Z}/N\mathbb{Z})^*$ of order $> (p + q - 2)/2$
- Compute $x = g^{(N-1)/2} = g^{(p+q-2)/2} \bmod N$ since $(N - \varphi(N) - 1)/2 = (p + q - 2)/2$ as $\varphi(N) = N - p - q + 1$
- Compute short discrete logarithm $d = (p + q - 2)/2$ from g and x
- Get many pairs (j, k) s.t. k is the ℓ most significant bits of $dj \bmod 2^m$: Hidden Number Problem (HNP)
- May, Schlieper[3]: we can replace f by $h \circ f$ where h is a universal hash function is still periodic

[3] "Quantum period-finding is compression robust"

Variant Shor's algorithm

Ideas

- Once $p + q$ is known, using $N = pq$, recover p is easy
- $G = \langle g \rangle$ a cyclic subgroup of $(\mathbb{Z}/N\mathbb{Z})^*$ of order $> (p + q - 2)/2$
- Compute $x = g^{(N-1)/2} = g^{(p+q-2)/2} \bmod N$ since $(N - \varphi(N) - 1)/2 = (p + q - 2)/2$ as $\varphi(N) = N - p - q + 1$
- Compute short discrete logarithm $d = (p + q - 2)/2$ from g and x
- Get many pairs (j, k) s.t. k is the ℓ most significant bits of $dj \bmod 2^m$: Hidden Number Problem (HNP)
- May, Schlieper[3]: we can replace f by $h \circ f$ where h is a universal hash function is still periodic
- How to compute some bits of $a^k \bmod N \bmod 2^r$ with $o(\log n)$ extra space using RNS

[3] "Quantum period-finding is compression robust"

Conclusion

- To break RSA-2048, 1400 logical qubits are needed

Conclusion

- To break RSA-2048, 1400 logical qubits are needed
- For DLP and small discrete log or Schnorr-like mechanisms, 300 logical qubits are needed (safe prime p of 1024 bits)

Conclusion

- To break RSA-2048, 1400 logical qubits are needed
- For DLP and small discrete log or Schnorr-like mechanisms, 300 logical qubits are needed (safe prime p of 1024 bits)
- For ECDLP, 2124 qubits for 256-bit [HJNRS20], ... and seems to be more complicate than factoring

Conclusion

- To break RSA-2048, 1400 logical qubits are needed
- For DLP and small discrete log or Schnorr-like mechanisms, 300 logical qubits are needed (safe prime p of 1024 bits)
- For ECDLP, 2124 qubits for 256-bit [HJNRS20], ... and seems to be more complicate than factoring
- but stay tune, many new results are coming