Shor's algorithm for factorization:

Breaking RSA on a quantum computer

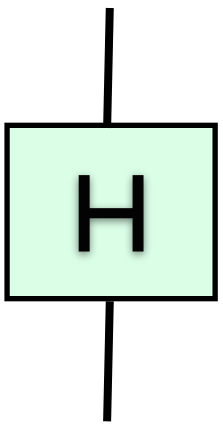
Operations on qubits

- Unitary evolution
  - reversible!
  - no destruction of info allowed
  - impossible to directly implement AND, OR, XOR, ...

- What kind of gates are possible? Generic construction:
  - #inputs = #outputs
  - some inputs unmodified
  - the unmodified inputs “control” the operation

Single-qubit gates

Hadamard gate

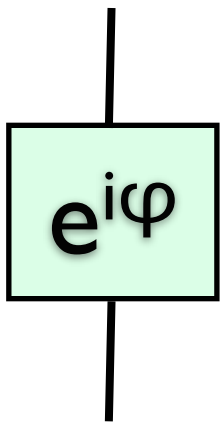


$$\frac{|0\rangle + |1\rangle}{\sqrt{2}}\langle 0| + \frac{|0\rangle - |1\rangle}{\sqrt{2}}\langle 1|$$

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$|0\rangle \mapsto |0\rangle$$
$$|1\rangle \mapsto e^{i\varphi}|1\rangle$$

Phase gate



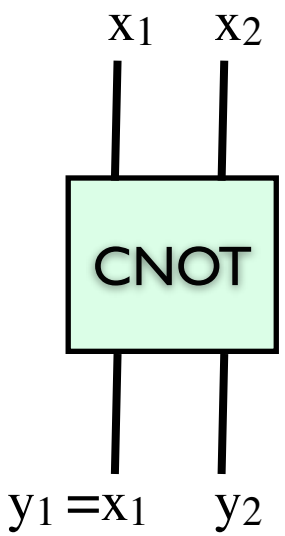
$$|0\rangle\langle 0| + e^{i\varphi}|1\rangle\langle 1|$$

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}$$

Two-qubit gates

Controlled NOT

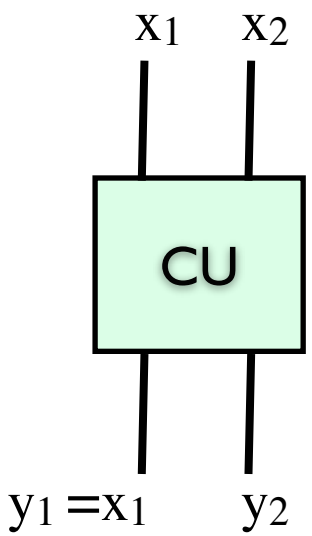
x_1	x_2	y_1	y_2
0	*0*	*0*	*0*
0	*1*	*0*	*1*
1	*0*	*1*	*1*
1	*1*	*1*	*0*



$$|00\rangle\langle 00| + |01\rangle\langle 01| + |11\rangle\langle 10| + |10\rangle\langle 11|$$

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

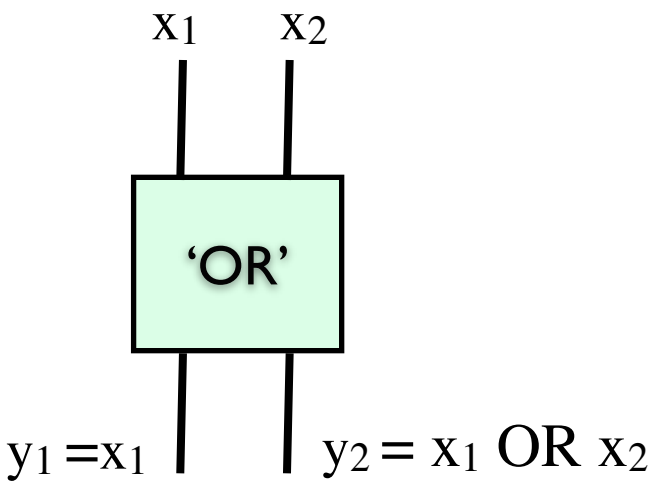
Controlled U



$$|00\rangle\langle 00| + |01\rangle\langle 01| + \Big(|1\rangle \otimes U|0\rangle\Big)\langle 10| + \Big(|1\rangle \otimes U|1\rangle\Big)\langle 11|$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix}$$

Question time



This is an OR operation with two-qubit output

- Is it reversible?
- Construct the truth table,

  the operator in Dirac notation

  and the 4x4 matrix
- Is the operator unitary?
- Same questions for XOR

Short review of RSA keys

- Private key
  - prime factors p and q
  - decryption exponent D
- Public key
  - modulus N = pq
  - encryption exponent E, with $ED=1 \bmod \varphi(N)$

$$\varphi(N) = (p-1)(q-1)$$

Difficult to derive private from public key

- Factoring is hard
- Finding φ(N) from N is equally hard
- You need φ(N) to derive D from E

Shor's algorithm (1996)

Factorization

- State of the art without quantum computing
  - general number field sieve
  - #operations: $\exp(1.9\, n^{1/3}\, [\log n]^{2/3})$ $\quad n = \log N = \#\text{bits}$

- Shor's algorithm
  - classical part & quantum part
  - quantum algorithm for finding ***period*** of function
  - #operations: $O(n^2\, [\log n][\log \log n])$

Factorization from period-finding

Proposition:

Given:
- $a \in Z^*_N$,
- the order r of a
- r is even
- $a^{r/2}+1 \not\equiv 0 \bmod N$

factors of N follow!

$$a^r - 1 \equiv 0 \bmod N$$

$$(a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \bmod N$$

cannot be 0 (def. of order r)

given: not 0

And yet $(a^{r/2} - 1)(a^{r/2} + 1) = kN$

Ergo: $(a^{r/2} - 1) \bmod N$ contains factor p

and $(a^{r/2} + 1) \bmod N$ contains factor q.

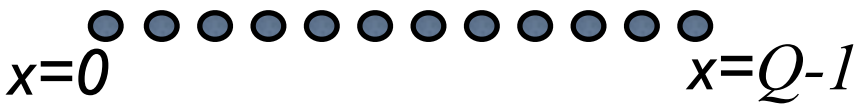
(or vice versa)

Classical part of Shor's algorithm

1. Draw a random integer $a < N$.
2. If $\gcd(a, N) \neq 1$ then you are amazingly lucky; you can find a nontrivial factor of N.
3. Use the quantum subroutine to find the period of a. (Denoted as r.)
4. If r is odd go back to step 1.
5. If $a^{r/2} + 1 \equiv 0 \bmod N$ go back to step 1.
6. $\gcd(a^{r/2} - 1, N)$ and $\gcd(a^{r/2} + 1, N)$ are factors of N.

Quantum part: period finding

The main tool: Discrete Fourier Transform (DFT)



function f on finite discrete space

$$\omega = \exp(i\frac{2\pi}{Q}) \qquad \omega^Q = 1$$

$$x, k \in \{0, \cdots, Q-1\}$$

$$f(x) = \frac{1}{Q}\sum_{k=0}^{Q-1} \tilde{f}(k)\omega^{kx}$$

$$\tilde{f}(k) = \sum_{x=0}^{Q-1} f(x)\omega^{-kx}$$

DFT is a basis transformation

Vector v expressed in the standard basis

$$|v\rangle = \sum_x |\underline{x}\rangle v_x \qquad v_x \stackrel{\text{def}}{=} \langle \underline{x}|v\rangle \qquad \sum_x |\underline{x}\rangle\langle\underline{x}| = \mathbf{1}$$

$$|v\rangle = \mathbf{1}|v\rangle = \left(\sum_{x=0}^{Q-1} |\underline{x}\rangle\langle\underline{x}|\right)|v\rangle = \sum_{x=0}^{Q-1} |\underline{x}\rangle\langle\underline{x}|v\rangle = \sum_{x=0}^{Q-1} |\underline{x}\rangle v_x$$

Vector v expressed in the Fourier basis

basis vectors $|\tilde{k}\rangle$, for $k \in \{0, \ldots, Q-1\}$

$$|\tilde{k}\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |\underline{x}\rangle \omega^{kx}$$

$$\boxed{|v\rangle = \sum_{k=0}^{Q-1} |\tilde{k}\rangle\langle\tilde{k}|v\rangle = \sum_{k=0}^{Q-1} |\tilde{k}\rangle \tilde{v}_k}$$

$$\tilde{v}_k \stackrel{\text{def}}{=} \langle\tilde{k}|v\rangle$$

$$\langle\tilde{k}'|\tilde{k}\rangle = \delta_{kk'}$$

Discrete Fourier Transform, Dirac notation

$$|\tilde{k}\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |\underline{x}\rangle \omega^{kx} \qquad \tilde{v}_k \stackrel{\text{def}}{=} \langle \tilde{k}|v\rangle$$

$$\boxed{U_{kx} \stackrel{\text{def}}{=} \langle \tilde{k}|\underline{x}\rangle = \frac{1}{\sqrt{Q}} \omega^{-kx}}$$

same as ordinary DFT

$$\tilde{v}_k = \langle \tilde{k}|v\rangle = \langle \tilde{k}| \left(\sum_x |\underline{x}\rangle\langle \underline{x}| \right) |v\rangle = \sum_x \langle \tilde{k}|\underline{x}\rangle\langle \underline{x}|v\rangle = \sum_x U_{kx} v_x$$

$$v_x = \langle \underline{x}|v\rangle = \langle \underline{x}| \left(\sum_k |\tilde{k}\rangle\langle \tilde{k}| \right) |v\rangle = \sum_k \langle \underline{x}|\tilde{k}\rangle\langle \tilde{k}|v\rangle = \sum_k (U^{-1})_{xk} \tilde{v}_k$$

Fourier example #1

Wave $v_x = e^{i\frac{2\pi}{R}x}/\sqrt{Q}$ $\qquad R \ll Q$

$$\tilde{v}_k = \frac{1}{Q}\sum_{x=0}^{Q-1} \omega^{-kx} e^{ix\frac{2\pi}{R}} = \frac{1}{Q}\sum_{x=0}^{Q-1} \omega^{x(\frac{Q}{R}-k)}$$

If Q/R is an integer, then the result is $\tilde{v}_k = \delta_{k,Q/R}$.

If Q/R is not integer

$$\tilde{v}_k = \frac{1}{Q}\cdot\frac{1-\omega^{(Q/R-k)Q}}{1-\omega^{Q/R-k}} = \frac{1}{Q}\cdot\frac{1-e^{i2\pi Q/R}}{1-\omega^{Q/R-k}} \propto \frac{1}{1-\omega^{Q/R-k}}$$

Has a sharp peak around *Q/R*

Fourier example #2

$v_x = g(x)$ where g is some arbitrary function that has period R

$R \ll Q$

Q/R not integer

$g(y+jR) = g(y) \qquad x = y + jR \qquad \sum_{x=0}^{Q-1}(\cdots) = \sum_{y=0}^{R-1}\sum_{j=0}^{t-1}(\cdots)$

$t \approx Q/R$

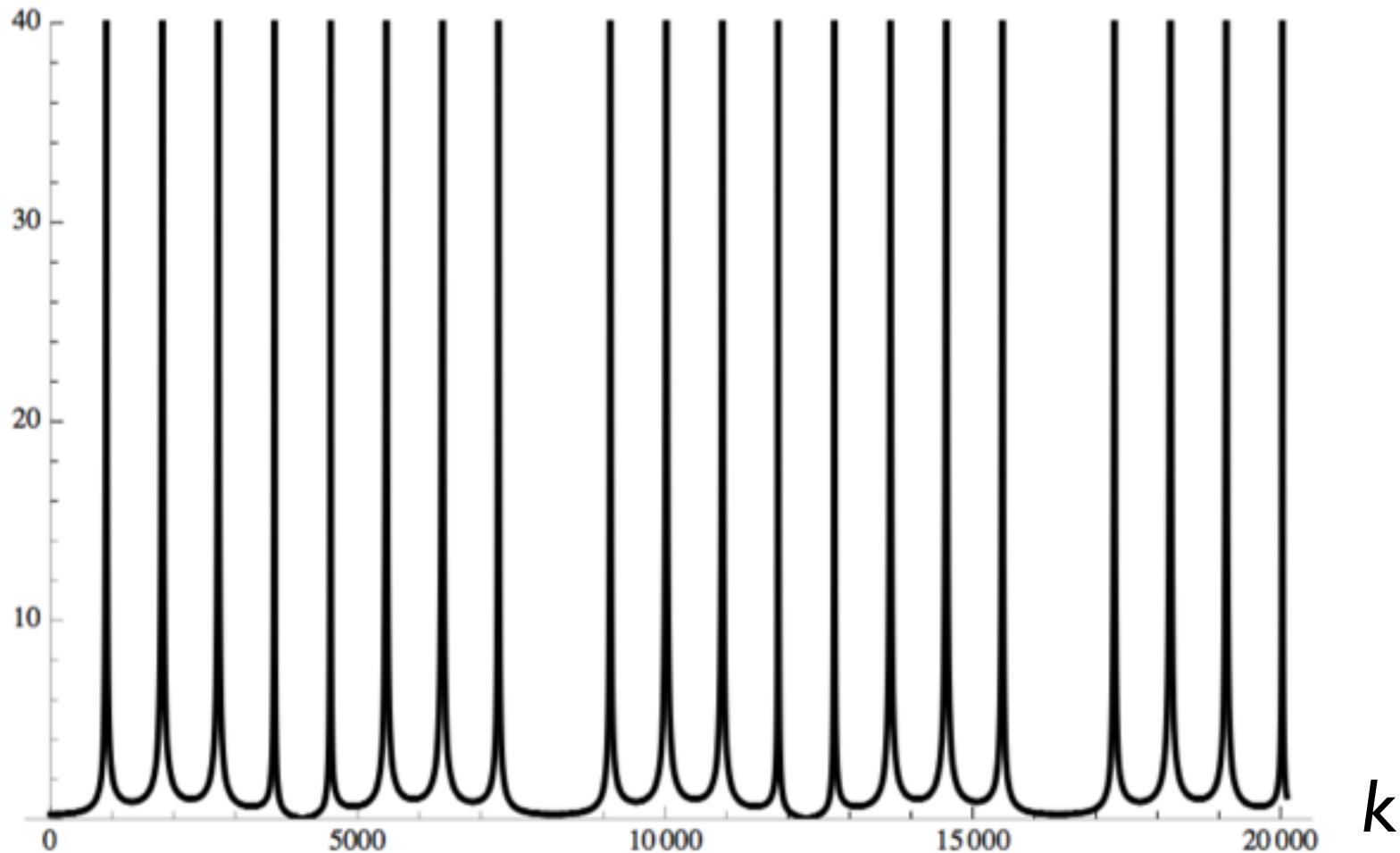
$$\tilde{v}_k = \frac{1}{\sqrt{Q}}\sum_{y=0}^{R-1}\sum_{j=0}^{t-1} g(y)\omega^{-k(y+jR)} = \frac{1}{\sqrt{Q}}\sum_{y=0}^{R-1} g(y)\omega^{-ky}\sum_{j=0}^{t-1}(\omega^{-kR})^j$$

$$kR \bmod Q \neq 0: \qquad \sum_{j=0}^{t-1}(\omega^{-kR})^j = \frac{1-(\omega^{-kR})^t}{1-\omega^{-kR}}$$

$kR \bmod Q = 0$: $\sum_{j=0}^{t-1}(\omega^{-kR})^j = t$ *very large number!*

The Fourier transform $\tilde{v}_k$ is nonzero mostly in the direct vicinity of $k \approx Q/R$ and integer multiples of Q/R (so-called harmonics).

$$\left| \frac{1-(\omega^{-kR})^t}{1-\omega^{-kR}} \right|$$



Q/R=910

Quantum part of Shor's algorithm

$Q = 2^q$ such that $N^2 < Q < 2N^2$.

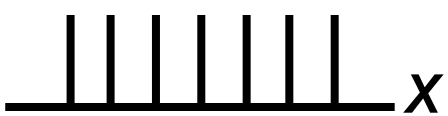
Two registers of size q

Initial state $|\Psi_0\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |\underline{x}\rangle|\underline{0}\rangle$

Apply the function $f(x) := a^x \bmod N$ $|\Psi_1\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |\underline{x}\rangle|\underline{f(x)}\rangle$

Measure 2nd register in standard basis

$$|\Psi_2\rangle = \frac{1}{\sqrt{t}} \sum_{j=0}^{t-1} |\underline{z + jr}\rangle \otimes |\underline{f(z)}\rangle \qquad t \approx Q/r$$

1st register state:  has period *r*

Measure first register in Fourier basis

How to measure in the Fourier basis?

Apply this basis-change operation

$$V_{\mathrm{DFT}} = \sum_k |\underline{k}\rangle\langle\tilde{k}|$$

and then measure in the standard basis.

(Coefficient $\tilde{v}_k$ is moved from k-basis vector to standard basis vector.)

What is the bottleneck in Shor's algorithm?

Implementation of $f(x) = a^x \bmod N$

Repeated squarings, using efficient multiplication

Quantum algorithms

Knowm algorithms

http://math.nist.gov/quantum/zoo/

- "Hidden subgroup"
  - period finding (Shor)
  - breaks RSA
  - breaks discrete logs
- Grover
  - find a special entry in a list of size N
  - runtime $O(\sqrt{N})$ instead of O(N)
- Decoding
  - only specific types of code
- HHL: linear system of equations, with constraint
- ....

"Post-quantum" cryptography

Crypto that is secure even when attacked by quantum computers.

Asymmetric

- Avoid factoring and discrete logs
- Difficult problems from lattices / coding / ...
- Hash-based

Symmetric

- Much easier
- Double key length (because of Grover)
- ...

Quantum cryptography

- Quantum Key Distribution
- Quantum Oblivious Transfer
- Quantum Key Recycling
- Unclonable Encryption
- Revocable Commitment
- Quantum public keys
- Quantum Readout of PUFs
- ....

Much easier than Quantum Computation!