Shor's algorithm

The magic of the Quantum Fourier transform

J Avron

March 2, 2022

What classical computers cant do
Factoring

- Factoring: $35 = \underbrace{5 \times 7}_{primes}$
- Try $35/2 = ?, 35/3 = ? \ldots$
- # trials: $\sqrt{N}$
- Best known: $O\left(e^{n^{1/3}\cdots}\right), \; n = \log N$

$\#$ with 230 digits

2000 years on 2.2 GHz processor

RSA cryptosystem

It's not a bug, it's a feature

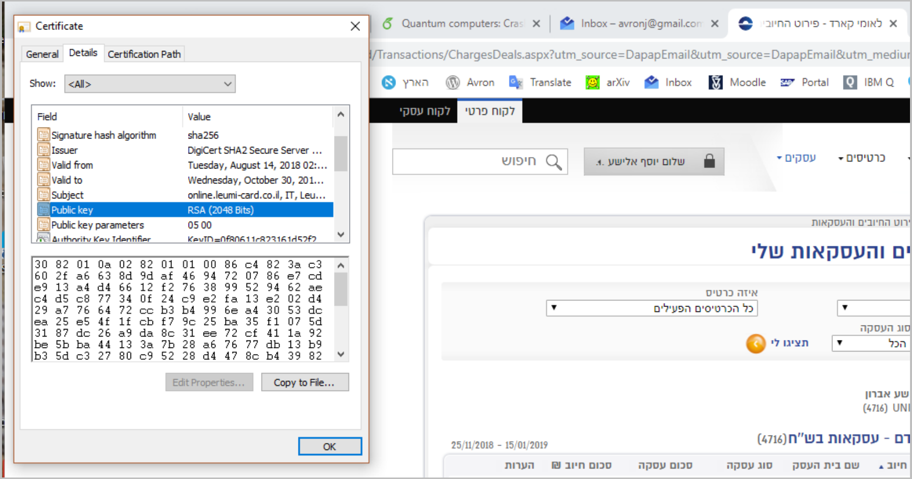
- $\underbrace{N}_{public} = \underbrace{p \times q}_{secret}$
- $Cipher = f(Message, N)$
- $Message = g(Cipher, p, q)$

RSA security

- f, g are known functions
- $Cipher = (Message)^e \; Mod \; N, \;\; Message = (Cipher)^d \; Mod \; N$
- $e \times d = Mod(p-1)(q-1)$, e=public, d=private
- Security rests on the presumed difficulty of factoring

Everybody uses RSA

All the time



The quantum threat

Shor algorithm

- Peter Shor 1994
- Fast factoring
- Time $= O((\#digits)^2)$
- Needs a quantum computer

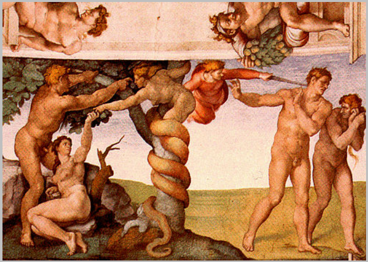
Quantum computer

Allows for fast factoring

The potential disaster/benefits

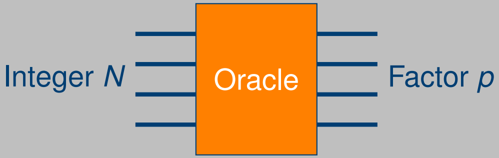
If a fast factoring algorithm is found

Bad	Good
The bastards read your email	You read the mail of the bastard
Internet insecure	Dark-net is insecure
Financial transaction insecure	Money laundering more difficult
State records exposed	State records exposed
...	...



Factoring Oracle

Weak and unreliable is good enough



$$\text{Oracle}(N) = \begin{cases} \text{Error} & \text{Probability} = 1/2 \\ 1,\text{N} & \text{Porbability} = 3/10 \\ 42 & \text{Porbability} = 1/5 \\ p & \text{Probability} = 1/10 \end{cases}$$

Verify answer on a classical computer

- If incorrect, query again
- 10 trials will give p w.h.p.

Math Preliminaries

Facts from number theory

- $a^k \mod N$: A periodic function of k, assuming $gcd(a, N) = 1$
- Example: $a = 2, N = 15$ the period=4

k	1	2	3	4	5	...	15
2^k *Mod* 15	2	4	8	16=1	2	...	8

- Euler-Fermat: $a^{(p-1)(q-1)} = 1 \mod N, \quad gcd(a, N) = 1$

Factoring reduces to finding the period of $a^k \mod N$

- $pq = N$
- $(p-1)(q-1) = \text{Integer} \times \text{period}$
- Period gives information on the private key

More math preliminaries

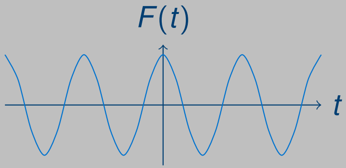
Fourier transform and its Discrete cousin

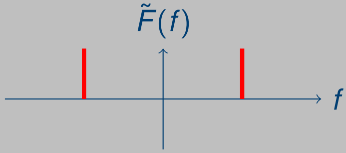
- $$\tilde{F}(f) = \frac{1}{\sqrt{2\pi}} \int e^{ift} F(t) dt$$
- $$\widetilde{e^{i\omega t}} \Longrightarrow \delta(f - \omega)$$

Discrete Fourier: $\underbrace{\omega = e^{2\pi i/L}}_{\text{root of unity}}$

$$\tilde{F}(m) = \sum_{k=1}^{L} \mathcal{F}_{mk} F(k), \quad \mathcal{F}_{km} = \frac{\omega^{km}}{\sqrt{L}}$$

$$\mathcal{F}_{L=2} = H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

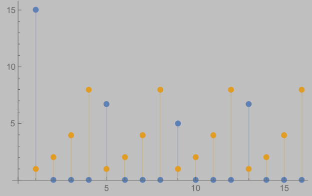




Periodic functions

Fourier transform is sparse

$$\tilde{F}(m) = \frac{1}{\sqrt{L}} \sum_{k=1}^{L} \omega^{km} F(k)$$



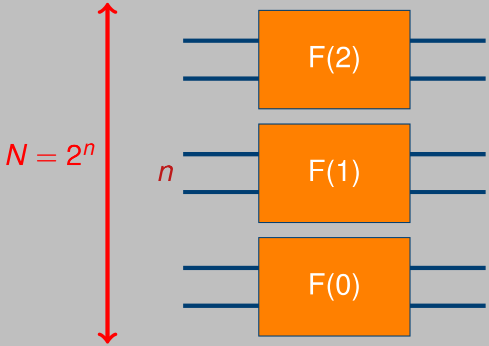
$$F(k + period) = F(k) \iff \tilde{F}(m) = \underbrace{\omega^{m\,\text{period}}}_{?=1} \tilde{F}(m)$$

- $\tilde{F}(m) \neq 0 \implies m \times \text{period} = (\text{Integer}) \times L$
- $period = (integer)L/m$

k	0	1	2	3	4	5	...
2^k *Mod* 15	1	2	4	8	16=1	2	...
Fourier	X	0	0	0	X	...	0

Functions contain exponential amount of information

How many bits to store a function with $N = 2^n$ arguments?



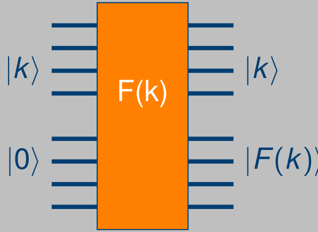
Storing $\{F\}$ needs $O(N \log N)$ bits

- n bits for each argument k
- N possible values for k

$\{F\}$ can be stored in $2n$ qubits

The superposition advantage

- n bits encode one k
- n bits encode $F(k)$
- n qubits for 2^n bits in superposition
- $(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \cdots \otimes (|0\rangle + |1\rangle)$
- $2n$ qubits encode $\{k, F(k)\}$

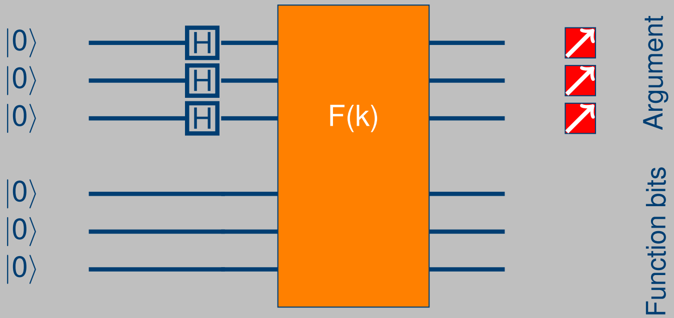


Parallel processing

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} |0\rangle \xrightarrow{\text{Function gate}} \frac{|0\rangle |F(0)\rangle + |1\rangle |F(1)\rangle}{\sqrt{2}}$$

No free-lunch principle

The massive superposition is only in the belly of the beast



Measurement reveals

- one, random, entry k and the corresponding $F(k)$

Shor algorithm

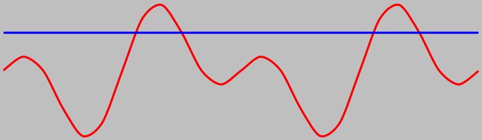
Quantum Fourier: Exponential improvement on FFT

- Under the hood: massive superposition

$$\underbrace{|0\ldots0\rangle}_{\textit{argument}}\ \underbrace{|a^0\rangle}_{\textit{function}} + \cdots + |1\ldots1\rangle\,|a^{L-1}\rangle$$

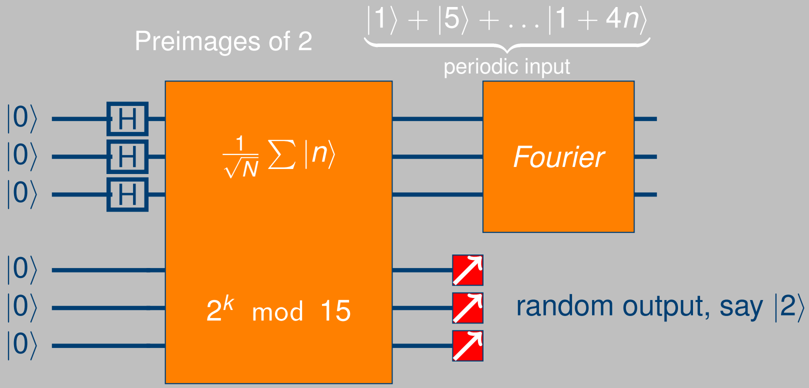
- Measure function register $|a^k\rangle$
- Get: Random outcome, e.g. $|a^k\rangle = |2\rangle$
- Argument register: superposition of pre-images of $|2\rangle$

$$\underbrace{\Big(|1\rangle + |1+4\rangle + |1+2\times4\rangle + |1+3\times4\rangle\Big)}_{\text{periodic sequence}} \otimes |2\rangle\,, \quad 2^{1+4n} = 2 \mod$$



If you look twice the cat is dead

Don't query the argument: Interfere



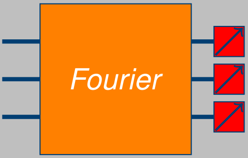
2^k *Mod* 15	1	2	4	8	1	2	...
k,m		1				5	...
Fourier	1	0	0	0	i	...	0

You also need to be lucky

You may not get enough information on the period

- Bad luck: Measure $|0\rangle$
- Learn nothing:
  $0 \times \textit{period} = \text{integer} \times L$

$\sum |1 + 4n\rangle$



2^k *Mod* 15	1	2	4	8	1	2	...
m	0	1	2	3	4	5	...
$\|Fourier\|^2$	1	0	0	0	1	...	0

Moral: Information in basis states exposed in one shot

Information in amplitudes is inaccessible in one shot

Fourier= Interference

- Computational States: Revealed in single shot
- Amplitudes: Revealed in statistics

Amplitudes: The roulette of the quantum casino