Lecture 12 – Quantum computers, Shor's algorithm, post-quantum cryptography

TEK4500

10.11.2020

Håkon Jacobsen
hakon.jacobsen@its.uio.no

MIT
Technology
Review

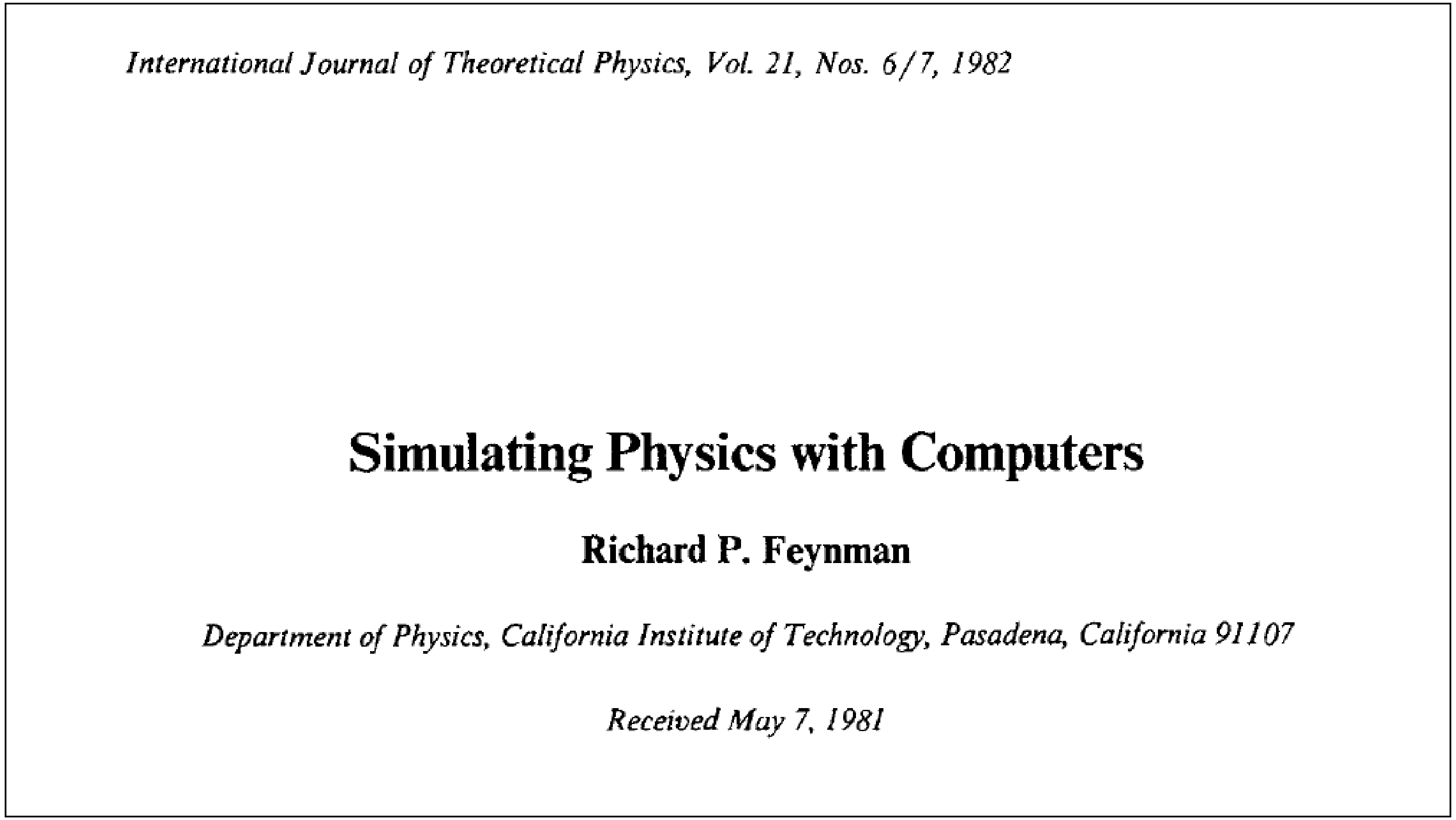
Computing

NSA Says It "Must Act Now" Against the Quantum Computing Threat

The National Security Agency is worried that quantum computers will neutralize our best encryption – but doesn't yet know what to do about that problem.

by Tom Simonite February 3, 2016

Quantum computing – the starting point



International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982

Simulating Physics with Computers

Richard P. Feynman

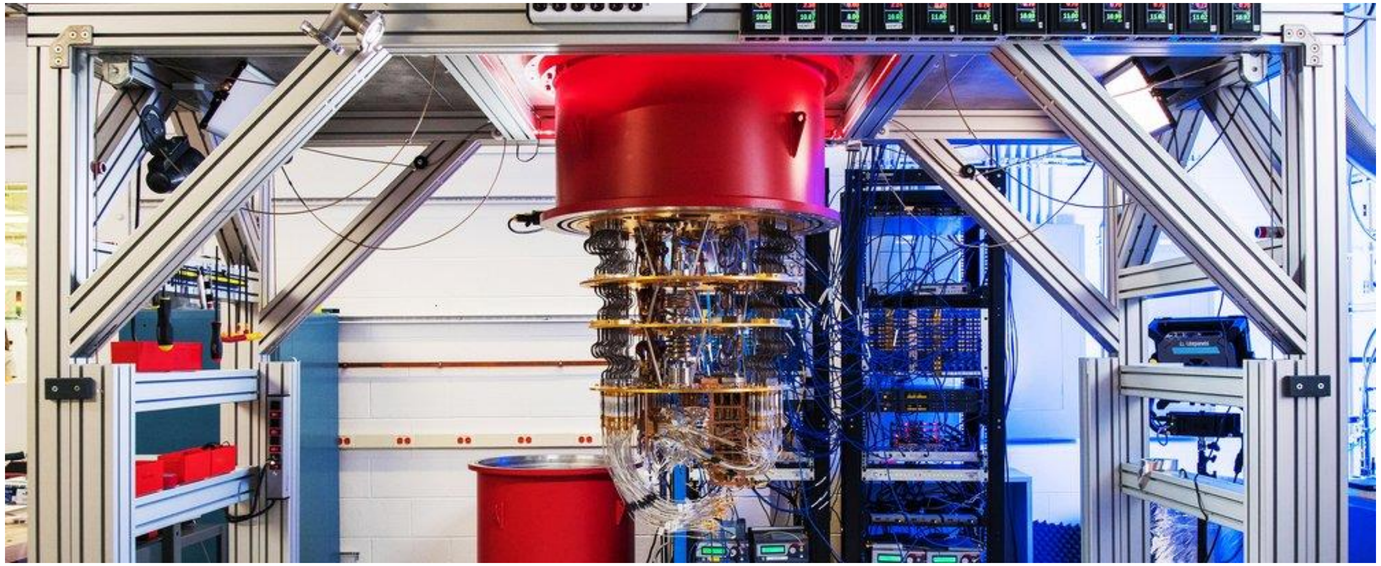
Department of Physics, California Institute of Technology, Pasadena, California 91107

Received May 7, 1981

Elements of (quantum) computing

- Three elements of all computations: data, operations, results

- Quantum computation
    - Data = **qubit**
    - Operation = **quantum gate**
    - Results = **measurements**

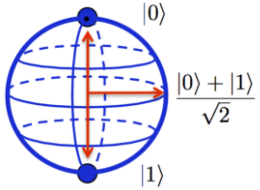


Qubits

- Classical bit: ● 0 ● 1

- Qubit:

Can be in a **superposition** of two basic states $|0\rangle$ and $|1\rangle$



$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad \alpha, \beta \in \boldsymbol{C} \qquad |\alpha|^2 + |\beta|^2 = 1$$

But we can never observe α and β directly!

Must **measure** $|\psi\rangle$ to obtain its value $\Longrightarrow$ state *randomly* collapses to either $|0\rangle$ or $|1\rangle$

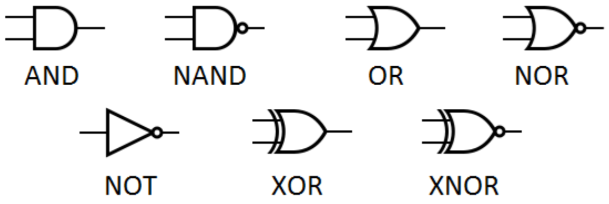
What's the probability of observing $|0\rangle$ or $|1\rangle$?

$$\Pr[\text{measure } |\psi\rangle \Rightarrow |0\rangle] = |\alpha|^2$$

$$\Pr[\text{measure } |\psi\rangle \Rightarrow |1\rangle] = |\beta|^2$$

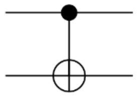
Quantum computation – quantum gates

- Classic bits are transformed using logical gates



- Qubits are transformed using **quantum gates**

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \mapsto |\psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$$

Operator	**Gate(s)**	**Matrix**
Pauli-X (X)		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

(Quantum) NOT-gate (or X gate)

$$|0\rangle \overset{\boldsymbol{X}}{\mapsto} |1\rangle$$

$$|1\rangle \overset{\boldsymbol{X}}{\mapsto} |0\rangle$$

$$\alpha|0\rangle + \beta|1\rangle \overset{\boldsymbol{X}}{\mapsto} \beta|0\rangle + \alpha|1\rangle$$

X gate:

$$\boldsymbol{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

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

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \qquad |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\boldsymbol{X}(|0\rangle) = \boldsymbol{X}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \boldsymbol{X}(|1\rangle) = \boldsymbol{X}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \boldsymbol{X}(|\psi\rangle) = \boldsymbol{X}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \ ?$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

The Z gate

$$|0\rangle \overset{Z}{\mapsto} |0\rangle$$

$$|1\rangle \overset{Z}{\mapsto} -|1\rangle$$

$$\alpha|0\rangle + \beta|1\rangle \overset{Z}{\mapsto} \alpha|0\rangle - \beta|1\rangle$$

Z gate:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

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

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \qquad |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$Z\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad Z\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \qquad Z\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = ?$$

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix}$$

The Hadamard or H gate

$$|0\rangle \overset{\boldsymbol{H}}{\mapsto} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$|1\rangle \overset{\boldsymbol{H}}{\mapsto} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

H gate:

$$\boldsymbol{H} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

$$\Pr[\text{measure } |\psi\rangle = \alpha|0\rangle + \beta|1\rangle \Rightarrow |0\rangle] = |\alpha|^2$$

$$\Pr[\text{measure } |\psi\rangle = \alpha|0\rangle + \beta|1\rangle \Rightarrow |1\rangle] = |\beta|^2$$

$$\Pr[\text{measure } \boldsymbol{H}|0\rangle \Rightarrow |0\rangle] = \left|\frac{1}{\sqrt{2}}\right|^2 = 0.5$$

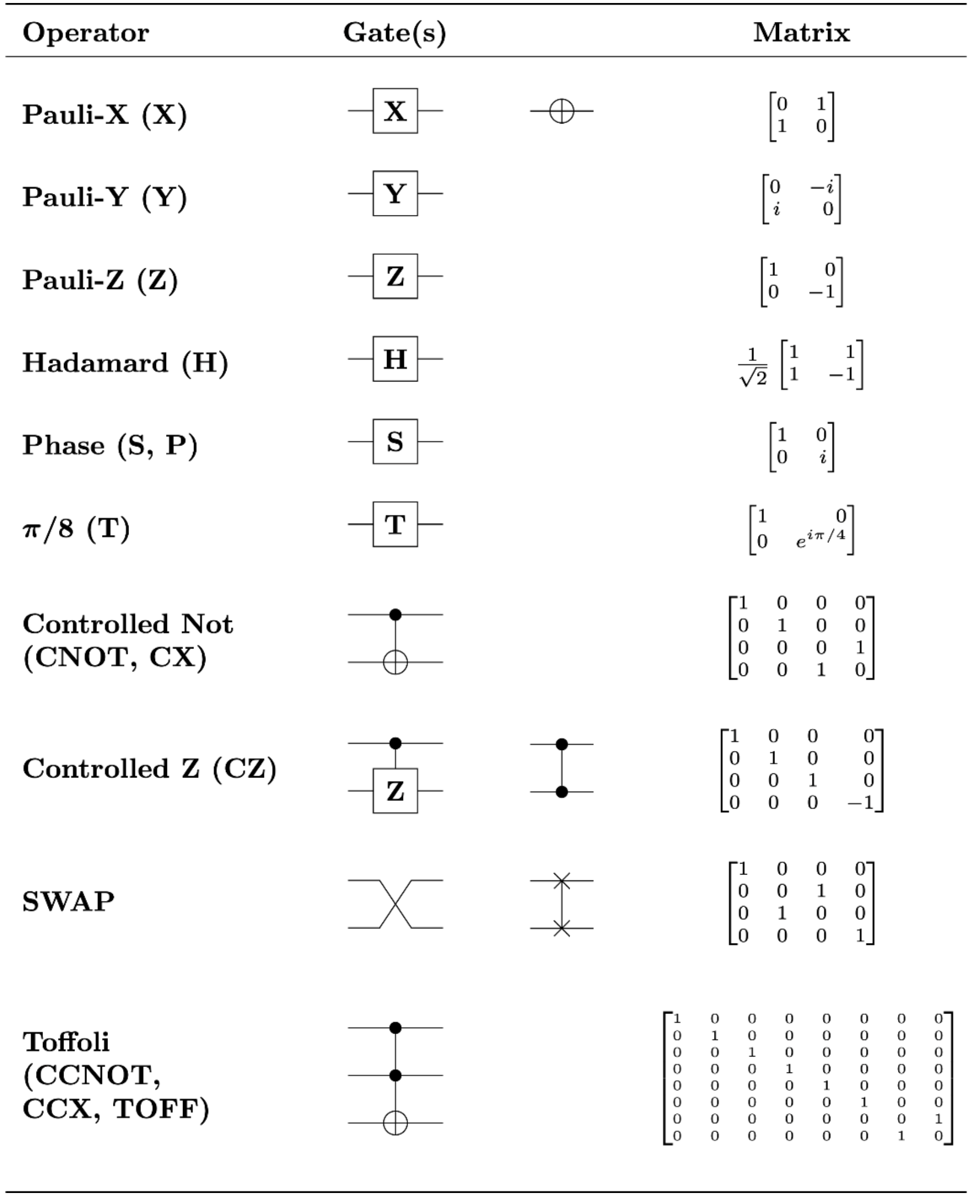
$$\Pr[\text{measure } \boldsymbol{H}|1\rangle \Rightarrow |1\rangle] = \left|\frac{1}{\sqrt{2}}\right|^2 = 0.5$$

The Hadamard gate allows us to create random bits!

$$\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$$

$$\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$$

Many other gates



Operator	Gate(s)		Matrix
Pauli-X (X)	X	$\oplus$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)	Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)	Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)	H		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)	S		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)	T		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)	Z		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP			$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)			$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Quantum gates

- Turns out all quantum gates can be described by matrices
  - In fact, very special matrices: unitary matrices
  - … and *only* unitary matrices! (fact of nature)

$$|0\rangle \overset{X}{\mapsto} |1\rangle \qquad |1\rangle \overset{X}{\mapsto} |0\rangle \qquad \boldsymbol{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- Quantum operations are *linear* and can be combined

$$|0\rangle \overset{Z}{\mapsto} |0\rangle \qquad |1\rangle \overset{Z}{\mapsto} |-1\rangle \qquad \boldsymbol{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$|\psi_0\rangle \overset{\boldsymbol{Z}}{\mapsto} |\psi_1\rangle \overset{\boldsymbol{X}}{\mapsto} |\psi_2\rangle \overset{\boldsymbol{H}}{\mapsto} |\psi_3\rangle \overset{\boldsymbol{Z}}{\mapsto} |\psi_4\rangle$$

$$\boldsymbol{ZHXZ}|\psi_0\rangle = |\psi_4\rangle$$

$$|0\rangle \overset{H}{\mapsto} \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |1\rangle \overset{H}{\mapsto} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \qquad \boldsymbol{H} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

$$\boldsymbol{ZHXZ}|0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$$

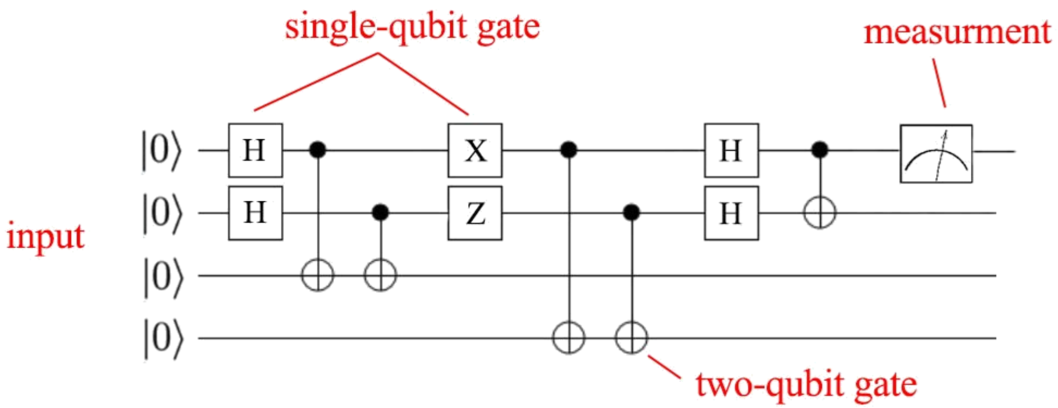
Quantum computers – multiple qubits

- A quantum computer consists of multiple qubits

$$|\psi_0\psi_1\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \qquad \alpha,\beta,\gamma,\delta \in \boldsymbol{C}$$

$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$$

- Can apply quantum gates to a subset of qubits in a multi-qubit state



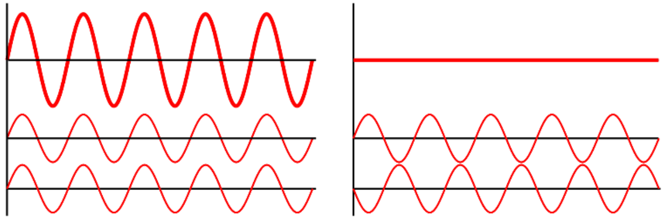
What makes quantum computation special?

- **Warning:** a quantum computer does *not* simply "try out all solutions in parallel"

- The magic comes from allowing complex (or even just negative real) superposition amplitudes

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

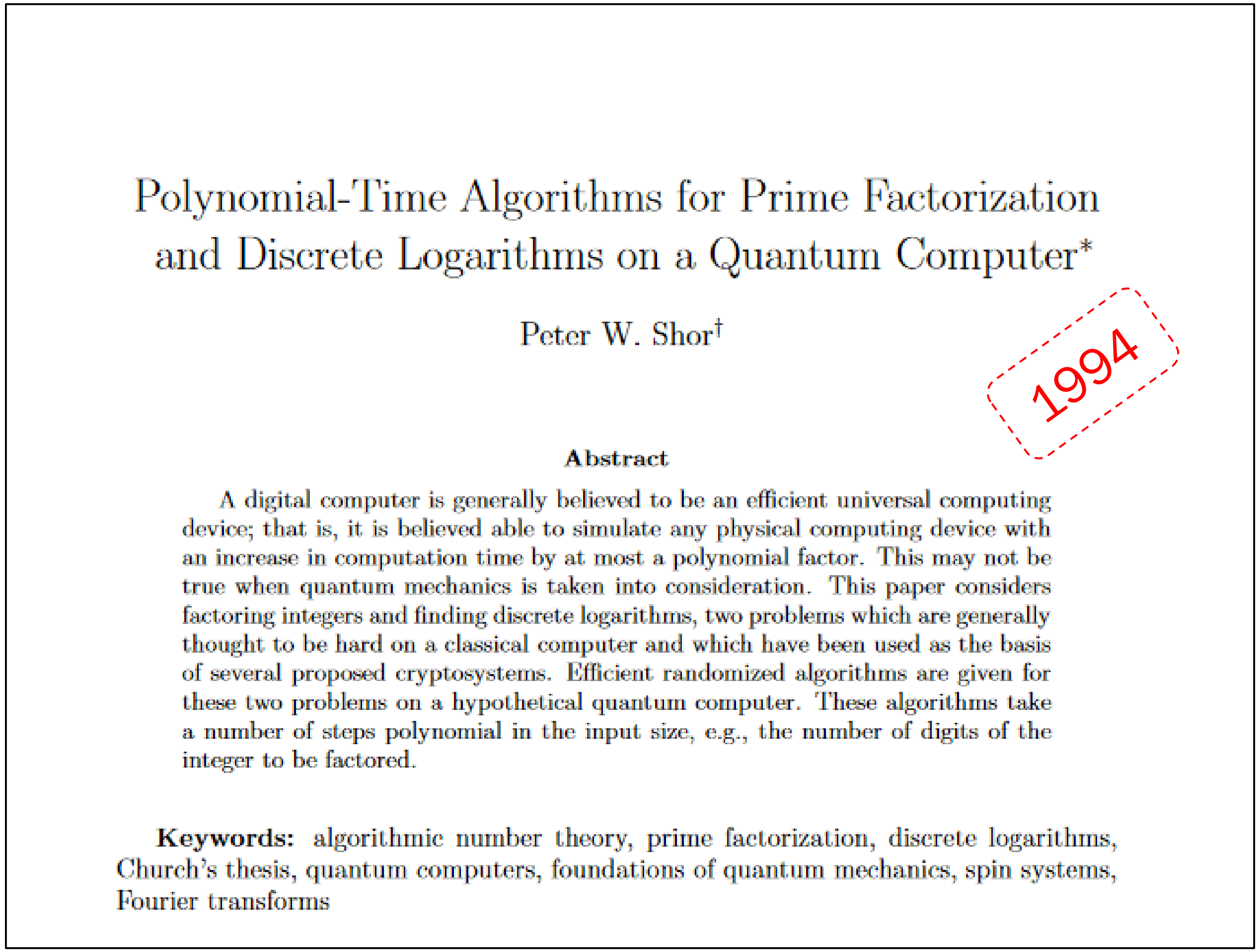
- Can *carefully* choreograph computations so that wrong answers "cancel" out their amplitudes, while correct answers "combine" (**quantum interference**)



  - increases probability of measuring correct result
  - only a few problems allow this choreography; speed up *not* possible for all computations

https://www.smbc-comics.com/comic/the-talk-3

Shor's algorithm



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

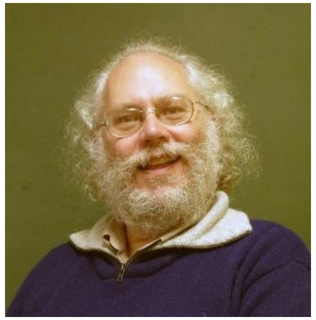
Peter W. Shor†

Abstract

A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

Keywords: algorithmic number theory, prime factorization, discrete logarithms, Church's thesis, quantum computers, foundations of quantum mechanics, spin systems, Fourier transforms

1994



Factoring to order-finding

$$2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, \ldots$$

$$2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1, 2 \ldots \qquad \mathrm{mod}\ 15$$

$$2, 4, 8, 16, 11, 1, 2, 4, 8, 16, 11, 1, 2 \ldots \qquad \mathrm{mod}\ 21$$

these sequences are periodic

Factoring to order-finding

$N = pq$

$|\langle a \rangle| = r$

$$\underbrace{a^1, a^2, a^3, \ldots, \overset{=1}{a^r}}, a^1, a^2 \ldots \quad \pmod N$$

order of a = the smallest positive r such that $a^r = 1 \pmod N$

Euler's theorem: for all $a \in \boldsymbol{Z}_N^*$

$$a^{\phi(N)} = a^{(p-1)(q-1)} = 1 \pmod N$$

Fact: r must divide $(p-1)(q-1)$

Proof:

- $(p-1)(q-1) = sr + t \qquad\qquad 0 \le t < r$
- $a^{(p-1)(q-1)} = a^{sr+t} = a^{sr} a^t = (a^r)^s a^t = 1 \cdot a^t = 1 \bmod N \implies t = 0$ (since r is the *smallest*)
- $(p-1)(q-1) = sr$ Q.E.D.

Conclusion: learn $r \implies$ we learn a factor of $(p-1)(q-1)$

repeat with a different $a \implies$ learn another factor of $(p-1)(q-1)$ (with high prob.)

eventually we learn full $(p-1)(q-1) \implies$ can find p and q (Problem set 10)

Factoring to order-finding

$N = pq$

$$\underbrace{a^1, a^2, a^3, \ldots, \overset{=1}{a^r}}, a^1, a^2 \ldots \quad (\mathrm{mod}\, N)$$

order of a = the smallest positive r such that $a^r = 1 \ (\mathrm{mod}\, N)$

how likely is this for random $a \in \boldsymbol{Z}_N^*$? **Answer:** very! (prob. ≥ 0.5**)**

Suppose: $\overbrace{r \text{ is } \textit{even} \text{ and } a^{r/2} \neq \pm 1}$

$$\boxed{x^2 - 1 = (x+1)(x-1)}$$

Then: $a^r - 1 = \left(a^{r/2}\right)^2 - 1 = \left(a^{r/2} + 1\right)\left(a^{r/2} - 1\right) = 0 \mod N \implies N$ divides $\left(a^{r/2} + 1\right)\left(a^{r/2} - 1\right)$

Then: N does *not* divide $\left(a^{r/2} + 1\right)$ or $\left(a^{r/2} - 1\right) \Longrightarrow p$ divides $\left(a^{r/2} + 1\right)$ and q divides $\left(a^{r/2} - 1\right)$

…or vice versa

Then: $\gcd\left(a^{r/2} + 1, N\right) = p$

…and $\gcd\left(a^{r/2} - 1, N\right) = q$

Shor's algorithm

This is where the quantum magic happens!

Shor's algorithm

Input: $N = pq$
Output: p (or q)

```
while true do
    $a \overset{\$}{\leftarrow} \boldsymbol{Z}_N^*$
    $r \leftarrow \mathrm{Order}_N(a)$          // how to find?
    if $r$ is even then
        $x \leftarrow a^{r/2} + 1 \pmod N$
        $p \leftarrow \gcd(x, N)$
        if $p \geq 2$ then
            return $p$
```

Shor’s algorithm

- To factor N: find order of a in $\boldsymbol{Z}_N^*$

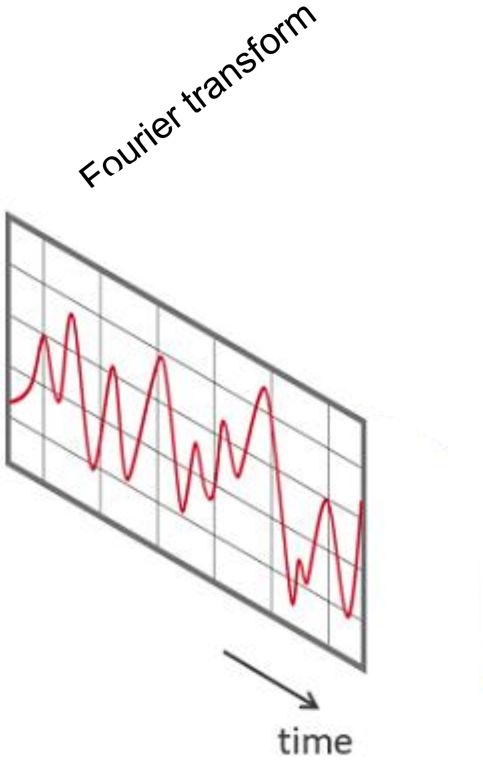
- Problem: r can be very large
  - Classical solutions take exponential time

- **Note:** the function $f(i) = a^i \bmod N$ is *periodic*:

$$f(i + kr) = a^{i+kr} = a^i \bmod N = f(i)$$

  - finding signal frequencies ⟺ finding signal period

- Key ingredient of Shor's algorithm:
  quantum Fourier transform (QFT)



Consequences of Shor’s algorithm

- Quantum order-finding algorithm can be implemented in $\mathcal{O}(k^3)$ quantum gate steps ($k = \log N$)
  - (quantum) polynomial time (BQP)

- Factoring is solvable in quantum polynomial time
  - Totally breaks RSA

- Modified Shor can also solve discrete logarithm problem
  - Totally breaks discrete log-based crypto
  - Including elliptic curve cryptography ☹

- Public-key crypto is dead…

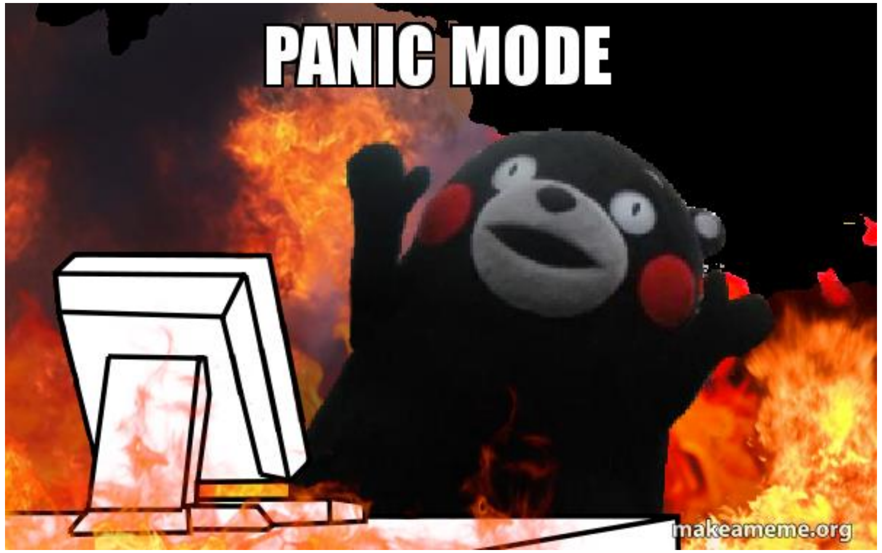
Shor's algorithm

Input: $N = pq$
Output: p (or q)

```
while true do
    $a \overset{\$}{\leftarrow} \boldsymbol{Z}_N^*$
    $r \leftarrow \mathrm{Order}_N(a)$            // QFT++
    if $r$ is even then
        $x \leftarrow a^{r/2} + 1 \pmod N$
        $p \leftarrow \gcd(x, N)$
        if $p \geq 2$ then
            return $p$
```

The quantum menace

- How far away is a quantum computer?
  - Nobody knows



- Building a large-scale quantum computer is a huge engineering challenge
  - *very* susceptible to noise (decoherence)
  - requires quantum error correction (is it even possible?)
  - many *physical* qubits needed to simulate a single *logical* qubit
    - ≈ 1000 physical qubits needed for 1 logical qubit
    - ≈ 1000 logical qubits needed for Shor's algorithm
    - largest (known) quantum computers:

≈ 65 physical qubits (IBM; 2020)	(no error correction)
≈ 53 physical qubits (Google; 2019)	(no error correction; demonstrated *quantum supremacy*)

Dealing with quantum computers

- Symmetric cryptography
  - Grover's algorithm: solves $\mathcal{O}(2^n)$ problems in $\mathcal{O}\left(2^{n/2}\right)$ quantum steps
  - Solution: double key-lengths $(128 \rightarrow 256)$

- Quantum cryptography
  - Use quantum mechanics to build cryptography

- Post-quantum cryptography
  - Classical algorithms believed to withstand quantum attacks

Post-quantum cryptography

- Public-key cryptography based on problems other than factoring and discrete logarithms

- **Top candidates:**

  - Lattice-based cryptography

  - Code-based cryptography

  - Multivariate cryptography

  - Hash-based cryptography

  - Isogeny-based cryptography

The NIST post-quantum competition

- Public competition to standardize post-quantum schemes
  - Public-key encryption
  - Digital signatures

- Started in 2017
  - Round 1: 69 submissions
  - Round 2: 26 candidates selected
  - Round 3: 15 candidates selected (current)

- Winner(s) expected in about a year

Algorithm (public-key encryption)	**Problem**
Classic McEliece	Code-based
CRYSTALS-KYBER	Lattice-based
NTRU	Lattice-based
SABER	Lattice-based
BIKE	Code-based
FrodoKEM	Lattice-based
HQC	Code-based
NTRU Prime	Lattice-based
SIKE	Isogeny-based

Algorithm (digital signatures)	**Problem**
CRYSTALS-DILITHIUM	Lattice-based
FALCON	Lattice-based
Rainbow	Multivariate-based
GeMSS	Multivariate-based
Picnic	ZKP
SPHINCS+	Hash-based

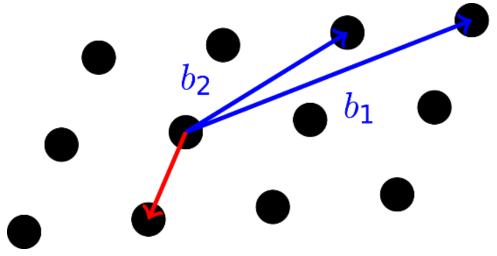
Lattice-based cryptography

- Very versatile computational problems
  - Public-key encryption
  - Digital signatures
  - Hash functions
  - Fully homomorphic encryption
  - Key exchange

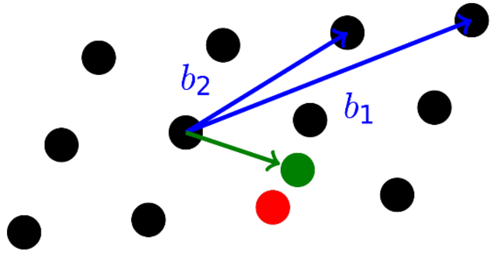
- Leads to efficient and compact schemes

- Based on hardness of problems in algebraic number theory
  - Believed to be hard also for quantum computers

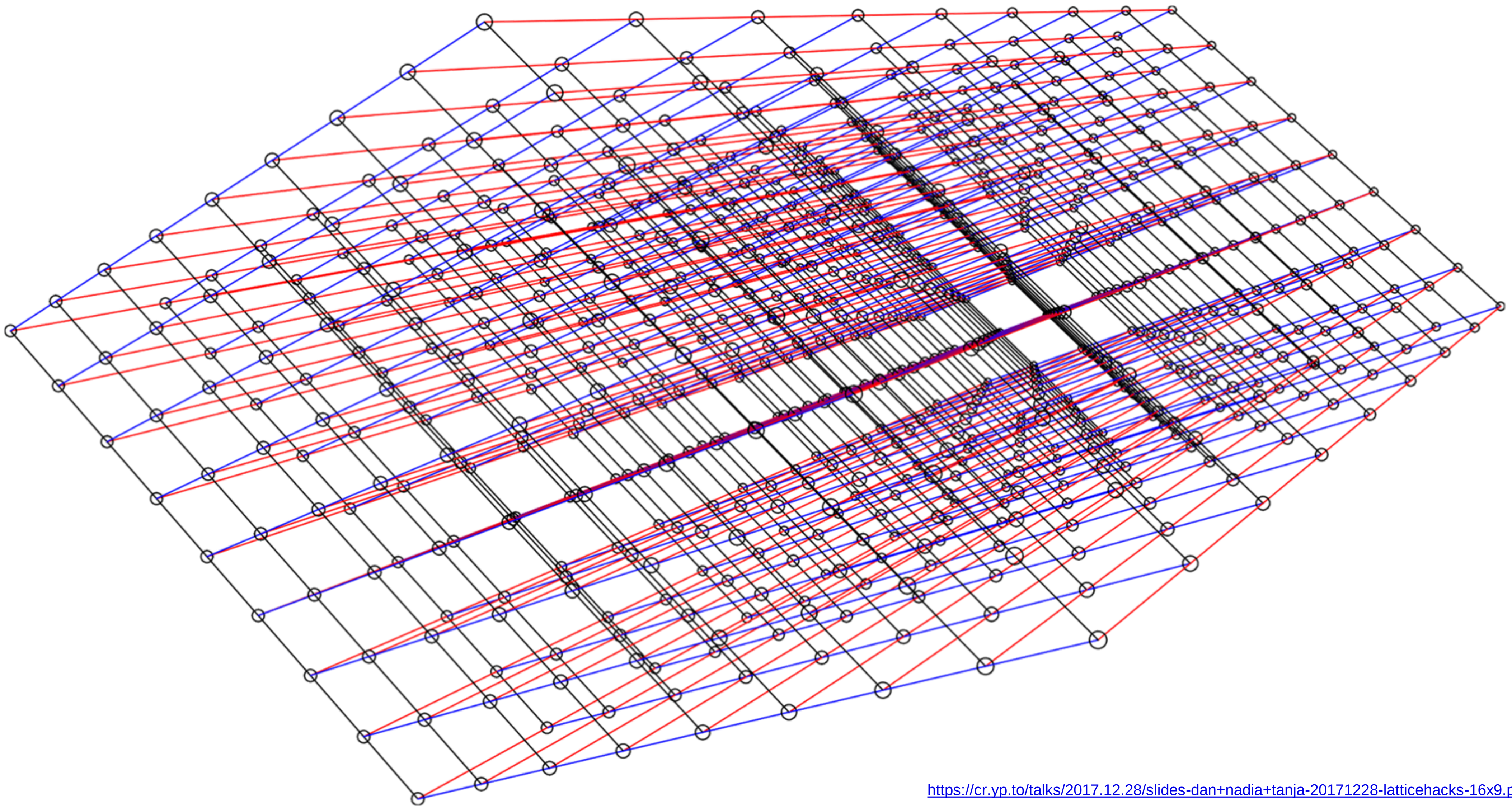
Shortest vector problem



Closest vector problem



Lattice-based cryptography



https://cr.yp.to/talks/2017.12.28/slides-dan+nadia+tanja-20171228-latticehacks-16x9.pdf

Learn more about post-quantum cryptography?

- Want to learn more about post-quantum cryptography?

- Sign up for TEK5550 - Advanced Topics in Cryptology next spring!

End of course

Next week

- Summary lecture

- Nothing planned; tell me want you want me to repeat/explain further

- **Exam**
  - Digital home exam
  - Wednesday November 25
  - 4 hours (possibly +0.5)
  - Format: single PDF file made available on Inspera and Canvas (similar to midterm)
  - Answers are typed directly into Inspera (no PDF upload); will create forms that mirrors problems in exam PDF
  - **NO collaboration** is allowed
  - Students may be picked out for conversations to prove ownership of answer