

Quantum Computing and Post-quantum Cryptography

PART 1

André Schrottenloher

Inria Rennes
Team CAPSULE

Inria

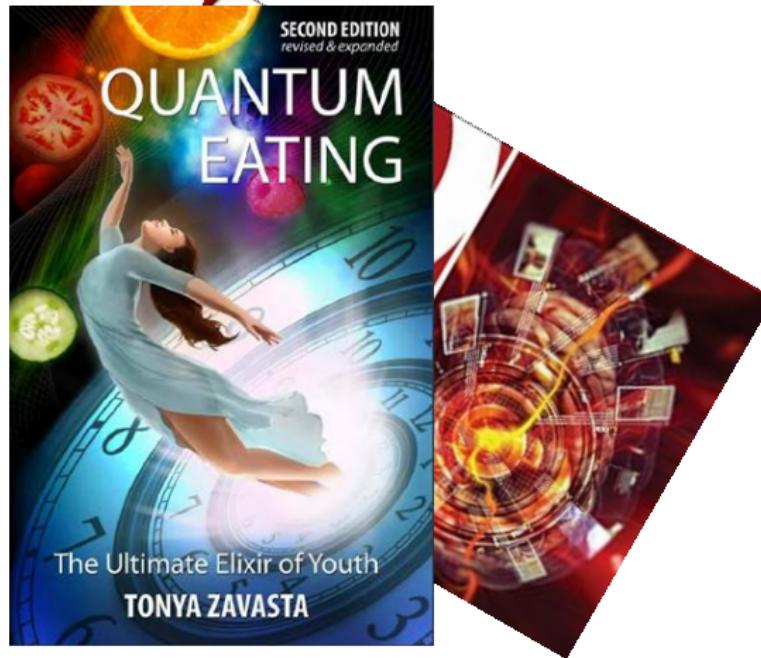


The quantum revolution



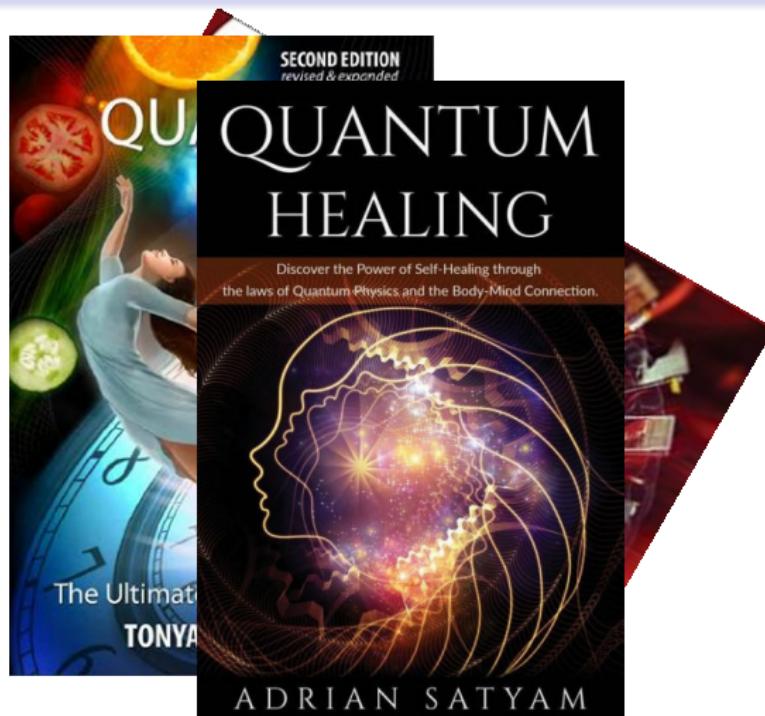
Images stolen from Xavier Bonnetain.

The quantum revolution



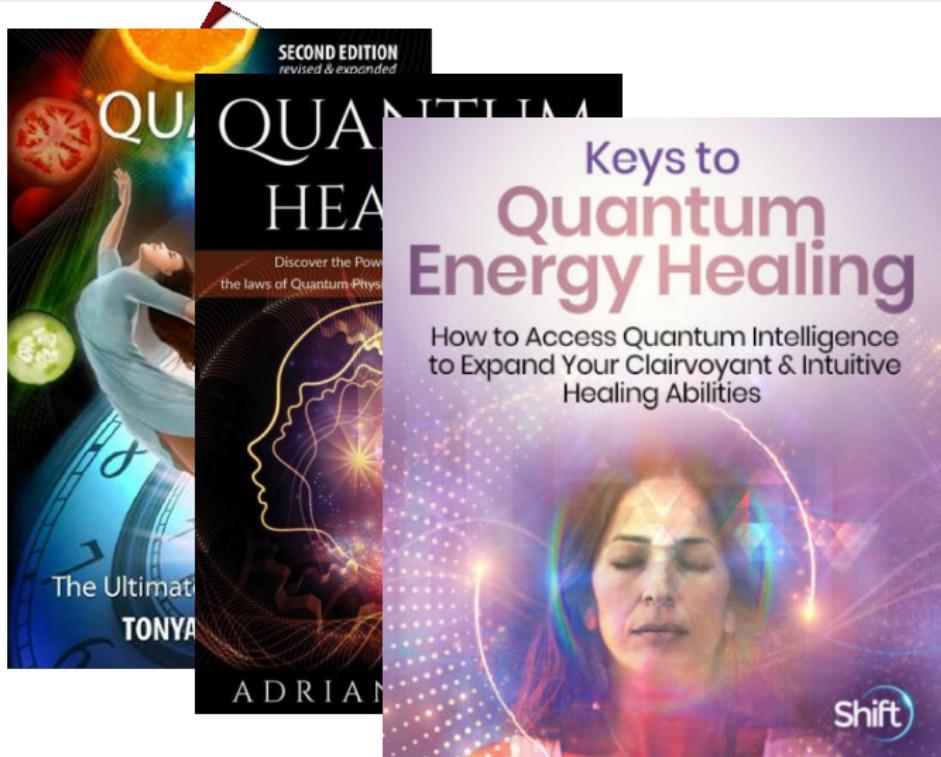
Images stolen from Xavier Bonnetain.

The quantum revolution



Images stolen from Xavier Bonnetain.

The quantum revolution



Images stolen from Xavier Bonnetain.

Meanwhile...

The race to save the Internet from quantum hackers

Researchers have created a new and potentially dangerous encryption-breaking quantum algorithm

Some cryptography researchers see the claim as misleading, others see it as a potential warning sign

By Jeremy Pezzano | January 14, 2023 at 2:38 PM | 23 comments

Quantum computers will crack your encryption – maybe they already have

Research teams worldwide are racing to create a computer so powerful it will be able to read encrypted messages.



IBM: Quantum computing poses an 'existential threat' to data encryption

Tim Keary
@tim_keary

January 17, 2023



Will quantum computers break RSA encryption in 2023?

Everybody knows that we should prepare ourselves for a "quantum future", but it was expected to come about in 10-20 years' time. Is a breakthrough possible this year?

Quantum computers can break major encryption method, researchers claim

It has long been known that one day quantum computers will probably be able to crack the RSA encryption method we use to keep data safe, but a team of researchers is now claiming it is already possible, while others say the results require more scrutiny.

Have Chinese scientists really cracked RSA encryption with a quantum computer?

Outline

- 1 Quantum Computing Basics
- 2 Examples of Quantum Algorithms
- 3 Quantum Algorithms vs. Cryptography

Quantum Computing Basics

Brief summary of quantum physics

I think I can safely say that nobody understands quantum mechanics.

– Richard Feynman (1918-1988)

- interpreting quantum physics is difficult
- good for us: we're not here to interpret, just to calculate

Brief history of quantum computing

- **Quantum computing** initiated in the 80s with the prospect of **simulating quantum mechanical systems**
 - ⇒ e.g., to understand protein folding
- Could it also be used to speed up **classical computations?**
 - ⇒ first significant quantum speedups appeared in the 90s



Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer", Proc. R. Soc. Lond. 1985

Qubits and superposition

A **bit** is a classical system which can be in the state 0 **or** 1.

$b = 0$ or 1

A **qubit** is a quantum system with two **basis states** $|0\rangle$ **and** $|1\rangle$.

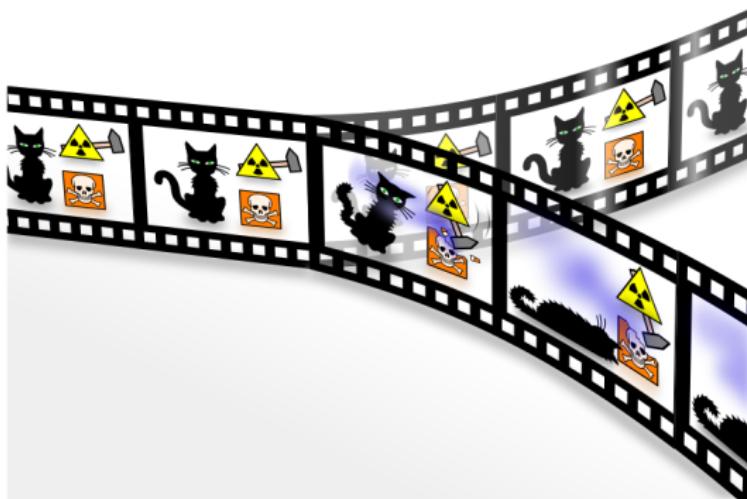
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$

Measurement

- The state is a **superposition**
- **Measuring** the qubit **destroys** the state and **collapses** the superposition to $|0\rangle$ or $|1\rangle$
- $|0\rangle$ is measured with probability $|\alpha|^2$
- $|1\rangle$ is measured with probability $|\beta|^2$

Qubits and superposition (ctd.)



$$|\text{cat}\rangle = \frac{1}{\sqrt{2}} |\text{cat is alive}\rangle + \frac{1}{\sqrt{2}} |\text{cat is dead}\rangle$$

- any two-state quantum system can be used as a qubit: even a cat

Qubits and entanglement

- **Two bits** can be in the state 00 **or** 01 **or** 10 **or** 11.
- **Two qubits** form a quantum system with **4 basis states**
 $|00\rangle, |01\rangle, |10\rangle, |11\rangle$
(4-dimensional vector space)

Consider the following state:

$$|\psi\rangle = \frac{1}{2} |00\rangle + \frac{1}{2} |01\rangle + \frac{1}{2} |10\rangle + \frac{1}{2} |11\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$$

Measure the first qubit: the second **always collapses** to $\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$.
⇒ the two qubits are **disentangled**

Qubits and entanglement (ctd.)

Consider the following state:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

Measure the first qubit:

- if the state **collapses** to $|00\rangle$: we measure 0 and the other becomes 0 with certainty
- if the state collapses to $|11\rangle$: we measure 1 and the other is 1 with certainty

Qubits and entanglement (ctd.)



- It still works if you send the second qubit to space: its state will collapse on 0 or 1
depending on the measurement result

- Experiments in the 1980s confirmed the theory
- Unfortunately for sci-fi, this doesn't allow faster-than-light communication



Picture: École polytechnique / Jérémie Barande

Qubits and entanglement (ctd.)

n qubits form a 2^n -dimensional quantum system with 2^n basis states:

$$|\psi\rangle = \alpha_{00..0} |00..0\rangle + \alpha_{01..0} |01..0\rangle + \dots + \alpha_{11..1} |11..1\rangle \in \mathbb{C}^{2^n}$$

It is (and remains) normalized: $\sum_i |\alpha_i|^2 = 1$.

An n -qubit quantum system is described by 2^n **complex amplitudes**. If the system evolves, we must recompute the 2^n amplitudes.

- this gets rapidly out of hand for classical computers
- this is why quantum computers were proposed in the first place!

Computations

- We start from a set of qubits initialized to $|00\dots0\rangle$
- We describe quantum algorithms as a sequence of basic, elementary **quantum gates**
- The quantum gates modify the current state of the algorithm
- Eventually we will **measure** the state

Starting from classical circuits

- Any **classical (reversible)** circuit can be applied to our qubits
- It will just apply **in superposition** to all possible states

$$\begin{aligned}|x\rangle &\rightarrow |f(x)\rangle \\ |x\rangle + |y\rangle &\rightarrow |f(x)\rangle + |f(y)\rangle\end{aligned}$$

A quantum computation is a **linear operator**.

“Quantum parallelism”

Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ be your favorite function (e.g., SHA-3). There exists a reversible circuit doing:

$$x, 0 \mapsto x, f(x)$$

i.e. a quantum algorithm:

$$|x\rangle |0\rangle \mapsto |x\rangle |f(x)\rangle$$

Start from a uniform superposition over x :

$$\left(\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \right) |0\rangle$$

apply f :

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$$

“Quantum parallelism” (ctd.)

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$$

Let's say we want a preimage of f , i.e., we fix y , and want x such that $f(x) = y$. It's here!

Are we simply computing “all the possibilities” in parallel?

NO

In superposition \neq in parallel

If we measure the state, we obtain a random $x, f(x)$: this is useless

Adding more quantum operations

We have additional operations that modify the amplitudes.

Summary

The 3 principles of quantum computing:

- ❶ superposition
- ❷ entanglement
- ❸ **interference** (next slides)

And the 4th one:

- Quantum computation is **not** “doing everything in parallel”

Quantum Algorithms: Shor and Grover

The QFT

The “real” quantum stuff happens when we **modify the amplitudes**.

Something that we really like is the

quantum Fourier transform (QFT)

If you see the **amplitudes** as some data series, the QFT takes the Fourier Transform of this series:

The 2^n -Quantum Fourier Transform:

$$\forall x, QFT_{2^n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_y \omega^{xy} |y\rangle \quad \left(\omega = \exp(2i\pi/2^n) \right)$$

Shor's algorithm for factorization

First of all, **reduce factorization to order-finding**.

- Let $N \leq 2^n$ be the number to factor. Select a constant a
- Finding r such that $a^r \equiv 1 \pmod{N}$ allows (w.h.p.) to factor N

$$(a^{r/2} - 1)(a^{r/2} + 1) = 0N$$

- r is solution to a hidden-period problem

Define $f(x) = a^x \pmod{N}$. Then for all x , $f(x + r) = f(x)$.

Shor's algorithm: step 1

Let's start with "everything in parallel on a large input space":

$$\frac{1}{\sqrt{2^m}} \sum_x |x\rangle |f(x)\rangle$$

Measuring $f(x)$ gives an output value b and **collapses**:

$$\sum_{x|f(x)=b} |x\rangle$$

The periodicity intervenes here:

$$\sum_{x|f(x)=b} |x\rangle = |x_0\rangle + |x_0 + \textcolor{orange}{r}\rangle + |x_0 + 2\textcolor{orange}{r}\rangle + \dots$$

Shor's algorithm: step 2

Apply a QFT on this state:

$$\begin{aligned} & \sum_y \omega^{x_0 y} |y\rangle + \sum_y \omega^{(x_0 + \textcolor{orange}{r})y} |y\rangle + \sum_y \omega^{(x_0 + 2\textcolor{orange}{r})y} |y\rangle + \dots \\ &= \sum_y \omega^{x_0 y} (1 + \omega^{\textcolor{orange}{r}y} + \omega^{2\textcolor{orange}{r}y} + \dots) |y\rangle \end{aligned}$$

What we obtain (ctd.)

After QFT, the probability to measure y is proportional to:

$$|\omega^{x_0 y} (1 + \omega^{ry} + \omega^{2ry} + \dots)|^2 = |1 + \omega^{ry} + \omega^{2ry} + \dots|^2$$

It gets bigger when ω^{ry} is closer to 1 $\implies \frac{ry}{2^m}$ closer to an integer.

- We measure y such that $\frac{ry}{2^m}$ is “close to an integer”.
- With sufficient precision, recover r using a classical post-processing.

What happened?

- The Fourier Transform of a periodic sequence gives “peaks” (and information on the sequence)
- We encoded a periodic sequence in the amplitudes
- Taking the QFT transforms the amplitudes into “peaks”
- By measuring, we get information on the period

Most common speedup: Grover's algorithm

Given a large search space:

- you can pick a guess
- you can test if your guess is “good” with a quantum circuit
- the probability to be good is p

\implies Grover search runs in time $\simeq \frac{1}{\sqrt{p}}$

An important message

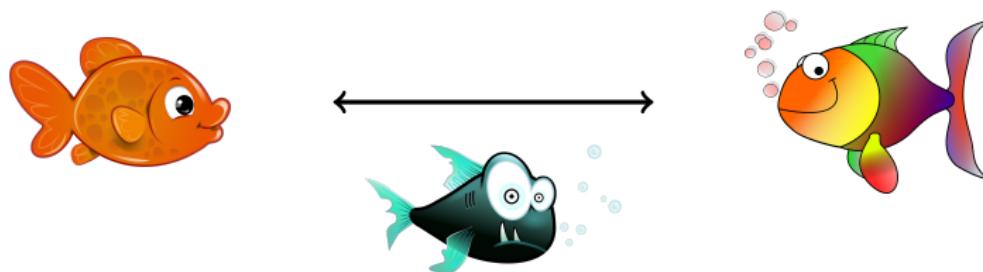
If the test is a **black box**, the quadratic speedup is **optimal**.

$\implies \sqrt{\cdot}$ speedup of many NP-complete problems, crypto problems, etc.

Quantum Algorithms vs. Cryptography

Cryptography in a nutshell

Enable (cheap) secure communications over insecure channels.



Public-key

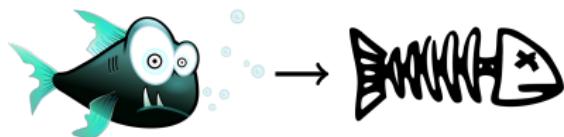
- **No shared secret**
- Key-exchange, signatures...
- RSA, elliptic curve cryptography ...

Secret-key

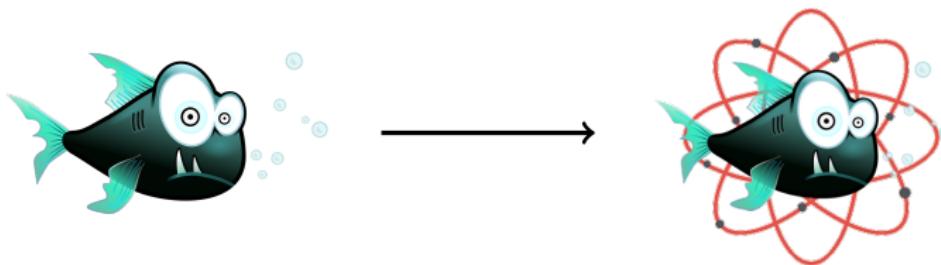
- **Shared secret**
- Block ciphers, stream ciphers, hash functions...
- AES, SHA-3 ...

Computational hardness

- Cryptography is based on **conjectured** hard computational problems
- To decrypt the communication,  has to factor large numbers, or find a secret AES key, etc.
- We estimate the **time** it would take to reach these goals, and ensure that it's infeasible



Example: the RSA cryptosystem



- Factoring a 2048-bit RSA public key $N = PQ$ should be infeasible
- If you do it, you break RSA
- Shor's algorithm solves factoring in polynomial time

⇒ and a “small” polynomial: only $\simeq 10^9$ quantum operations for RSA-2048

- Same for cryptosystems based on discrete logs (ECC)

⇒ breaks all public-key crypto used today

Post-quantum cryptography

- Solution: do not use DLP and factoring-based crypto anymore!

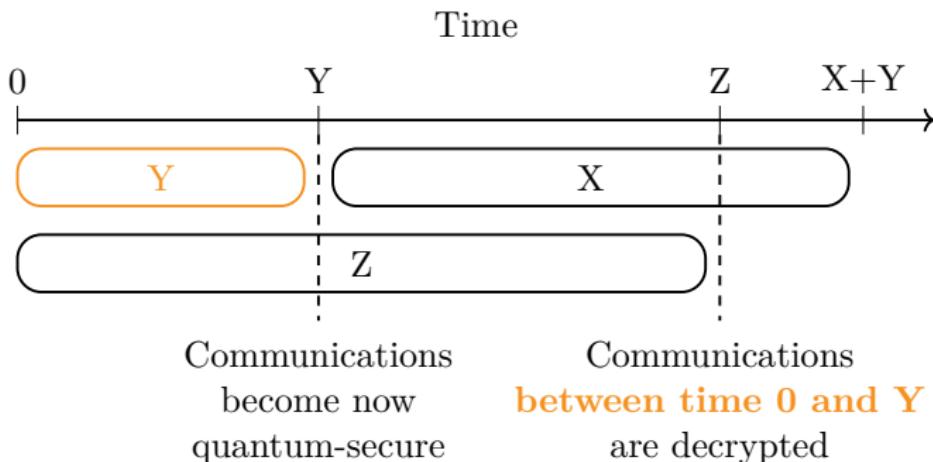
Post-quantum crypto = crypto that remains secure in the presence of a quantum adversary.

“But the quantum computer does not exist yet!”

- ⇒ The communication should remain secret for a time X (50 years?)
- ⇒ Changing to post-quantum crypto will take time Y (10 years?)
- ⇒ Building a QC will take time Z (30 years?)

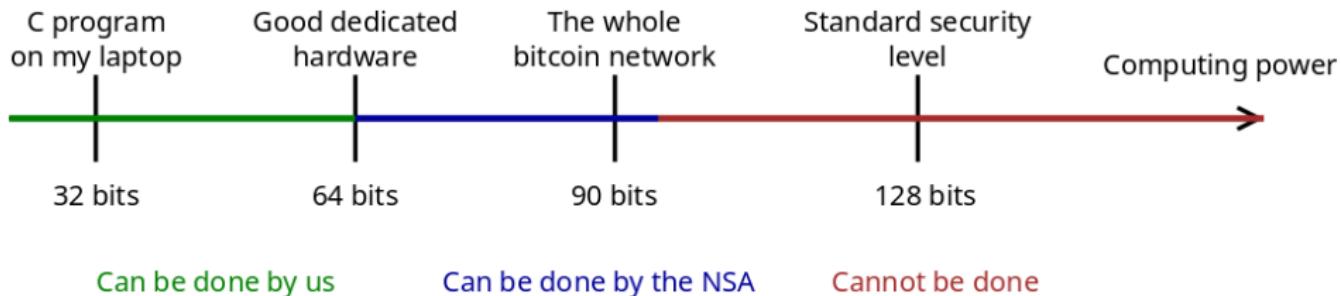
“Mosca’s theorem”

If $Y + X > Z$, you have a problem



Reasoning about quantum adversaries

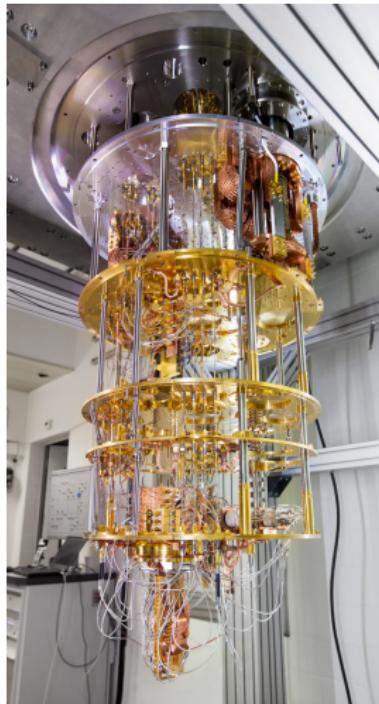
- Crypto is already about **attackers that do not exist**
- Many attacks are **theoretical algorithms** which show a weakness, but which will never run in practice



- Upgrading to PQC means just updating the notion of “algorithm”, and the landscape of attacks

Conclusion

Where are we today?



Several competing technologies, giants (IBM, Google, Rigetti. . .) and start-ups.



- Alice & Bob (cat qubits)
- Pasqal (neutral atoms)
- Quandela (photonics)

Where are we today? (ctd.)

- Step 1.** physics experiments (check that the theory works) **[Done]**
- Step 2.** quantum advantage (do “something” faster than supercomputers) **[Done]**
- Step 3.** perform quantum error correction **[In progress]** (recent results by Google)
- Step 4.** scale up, solve **useful problems** (physics, chemistry...) with true quantum advantage **[TODO]**
- Step 5.** scale up, break crypto **[TODO]**

Where are we today? (ctd.)

Current numbers: $\simeq 10^3 - 10^4$ gates on $\simeq 10^2 - 10^3$ qubits.

⋮

What we need: $\simeq 10^8 - 10^9$ gates on $\simeq 10^2 - 10^3$ qubits (*)

- But current qubits are “physical”: they have lots of errors
- To run large-scale computations, we will need to **correct the errors**

⇒ first results have been reported, but they are still very preliminary

(*) I've updated the numbers since last year!

Conclusion

- Quantum computers are **extremely good** at solving some **specific problems** (e.g., Shor)
- ... and quite good at solving other (less specific) problems (e.g., Grover)
- ... and totally useless at solving many other problems!

- Quantum computers today are still **experimental**
- Still a long way from breaking crypto

Cryptographers have been unlucky with Shor's algorithm, but we're going to make our cryptography **post-quantum**. See part 2!