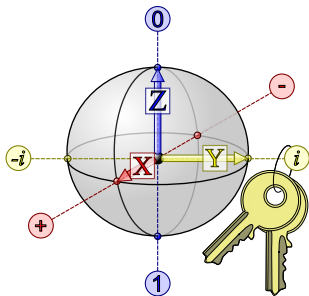


Quantum Computers & Cryptography



Benjamin Jurke

Postdoctoral Research Associate @
Department of Physics
Dana Research Center
Northeastern University, Boston MA

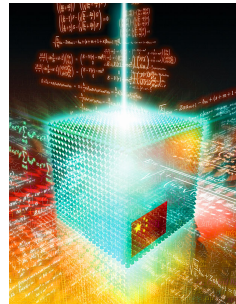
Boston Security Meetup

— Oct 20, 2011 —

Outline



1. Introduction to basic **quantum phenomena**.
2. Difference between **qubits and bits**. /
What is a quantum computer?
3. **Computational complexity** in
classical vs. quantum computers.
4. Implications for **cryptography**.

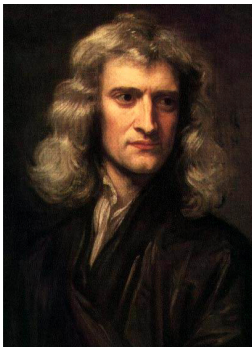


(c) Mondolithic Studios



What is Quantum Mechanics?

A very brief history of modern physics:



1687: Sir Isaac Newton publishes
“*Philosophiæ Naturalis Principia
Mathematica*”



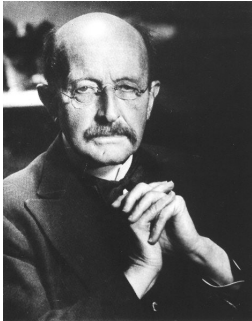
Birth of **classical mechanics**:

- **deterministic** behaviour
- **no chance** involved
- measurable **states** at any time



What is Quantum Mechanics?

A very brief history of modern physics:



1900: Max Planck introduces the quantum principle



Birth of **quantum mechanics**:

- probabilistic behaviour
- discretization of quantities
- quantum **entanglement**
- **superposition** of states
- uncertainty principle



What does Superposition mean?

Unlike in classical mechanics, **essentially different states can mix** in quantum mechanics — at least **until a specific property** is measured!

Well known thought experiment: **Schrödinger's cat**

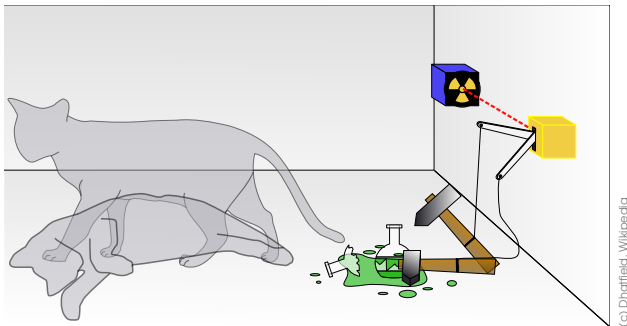
- Put a (living) **cat in box**.
- Add a container with **deadly poison**, that can be remotely released.
- Close & seal the box.
- Connect the remote to a **randomness** source (e.g. nuclear decay).

In what “state” is the cat? Is it **alive** or **dead**? Who knows?

Disclaimer: No animals were harmed during the making of those slides...



What does Superposition mean?



As long as no one checks (!!!), it is a sensible way to think about this to consider the cat being in an intermediate state:

$$\text{e.g.} \quad \langle \text{cat} \rangle = \frac{1}{3} \langle \text{alive} \rangle + \frac{2}{3} \langle \text{dead} \rangle$$

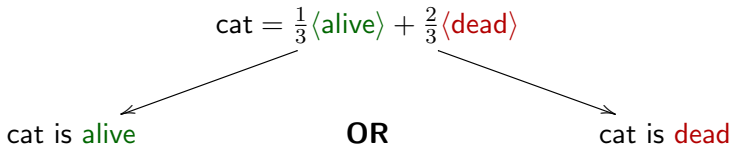


What does Superposition mean?

As long as no one checks (!!!) the cat is in a superposition of two states that are classically impossible to consider at the same time.

Now open the box.

Obviously, you will see (“measure”) either a dead or alive cat.



By “measuring” the system, you put it back in a classical, pure state!

→ **Measuring affects the system!**



What does Superposition mean?

But what do the numbers of the mixed state tell us? Take 1000 cats and **perform 1000 times the same experiment** in exactly the same arrangement such that for each cat we have

$$\langle \text{cat} \rangle = \frac{1}{3} \langle \text{alive} \rangle + \frac{2}{3} \langle \text{dead} \rangle.$$

After checking on the 1000 cats, we will **most likely** find

333 cats are **alive** and 667 cats are **dead**.

Could also be that 330 cats are alive and 670 are dead.

Or 300 alive and 700 dead.

Or none of the cats is alive and all 1000 are dead (do I hate cats?)...

Disclaimer: Again, no animals were harmed during the making of those slides...



What does Superposition mean?

Why does Schrödinger's cat appear to be so weird???



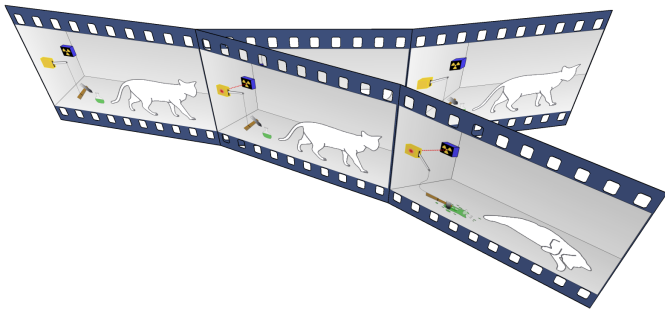
copied from 3dmajo.com

We're not accustomed to observing such quantum phenomena in our day-to-day lives.

Schrödinger's cat effectively “upscales” the rather common probabilistic **quantum behaviour** of a **microscopic object** (e.g. radioactively decaying nucleus) to **macroscopic proportions**.



What does Superposition mean?



(c) Dharfield, Wikipedia

Bottom line:

- It's sensible to consider a mixing of states.
- Measuring turns a mixed state back into ("picks out") a pure state.



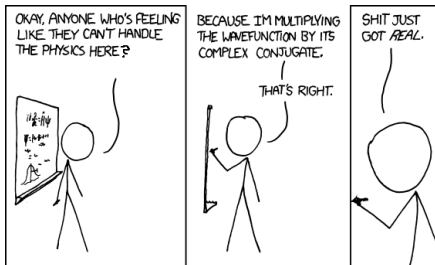
Superposition Revisited

Neglected: A quantum state is actually described by a **wave-function** ψ , a **solution** to Schrödinger's equation.

Important: The wave-function is in terms of complex numbers \mathbb{C} !

The **wave-function** encodes the **probability** distribution and the **superposition principle applies to the wave-function**, *NOT* the *probability*.

$$\begin{array}{c} \text{wave-function } \psi \\ \downarrow \\ \text{probability } P = |\psi|^2 \end{array}$$



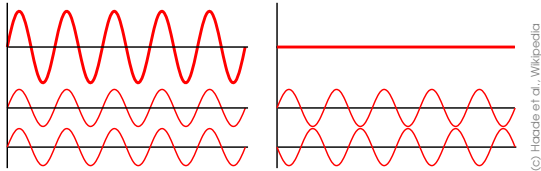
(c) XKCD Webcomic #849



Superposition Revisited

Waves are much more than just probability distributions!

Interference:



Quantum mechanics “happens” at the level of the wave functions ψ .

The (somewhat weird) effects that interference may have on the probability distribution $P = |\psi|^2$ are the quantum phenomena.



Multi-Cat Situations

states of a single Schrödinger cat: $\mathcal{H}_{1\text{cat}} = \{|\text{alive}\rangle, |\text{dead}\rangle\}$

Take a second cat, same arrangement and consider the **2-cat system**:

$$\begin{aligned}\mathcal{H}_{2\text{cats}} &= \mathcal{H}_{1\text{cat}} \otimes \mathcal{H}_{1\text{cat}} \\ &= \{|\text{alive}, \text{alive}\rangle, |\text{alive}, \text{dead}\rangle, |\text{dead}, \text{alive}\rangle, |\text{dead}, \text{dead}\rangle\}\end{aligned}$$

Situation **without interference**:

Starting from $\langle \text{cat} \rangle = \frac{1}{3}\langle \text{alive} \rangle + \frac{2}{3}\langle \text{dead} \rangle$ as before, we have

$$\langle \text{cats} \rangle = \frac{1}{9}\langle \text{alive}, \text{alive} \rangle + \underbrace{\frac{2}{9}\langle \text{alive}, \text{dead} \rangle + \frac{2}{9}\langle \text{dead}, \text{alive} \rangle}_{\text{two separate outcomes for "one dead cat" situation}} + \frac{4}{9}\langle \text{dead}, \text{dead} \rangle$$



Multi-Cat Situations

In multi-cat situations the **number of pure states grows exponentially**.
For 3 cats we already have $2^3 = 8$ possible pure states:

$$\mathcal{H}_{3\text{cats}} = \{|\text{alive, alive, alive}\rangle, \dots, |\text{dead, dead, dead}\rangle\}$$

In order to describe a mixed state wave-function ψ in this system we therefore **need 8 complex numbers!**

Have fun describing the
mixed state of a typical
Chinese market
multi-cat system...



copied from sirius.2kat.net



Introducing Qubits

A **single qubit** is nothing else than the “1-bit quantum system”!

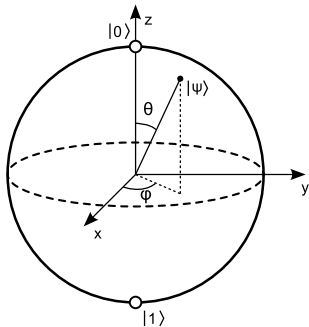
We have two classical states in a traditional bit: $|0\rangle$ and $|1\rangle$.

The qubit is described by the superposition of those two states:

$$\text{qubit} = \alpha|0\rangle + \beta|1\rangle \quad \text{for } \alpha, \beta \in \mathbb{C}$$

The probabilities of $P = |\text{qubit}|^2$ have to add up to one, it follows that

$|\alpha|^2 + |\beta|^2 = 1$. \rightarrow Represent as a sphere:



$$\alpha = \cos\left(\frac{\theta}{2}\right)$$

$$\beta = e^{i\varphi} \sin\left(\frac{\theta}{2}\right)$$



Introducing Qubits

For N qubits we need 2^N complex numbers to describe the state:

$$\alpha|000\rangle + \beta|001\rangle + \gamma|010\rangle + \delta|011\rangle + \epsilon|100\rangle + \zeta|101\rangle + \eta|110\rangle + \theta|111\rangle$$

For a “qubyte” = 8 qubits already 256 variables are required...

For a **1000 qubit system** we would need $2^{1000} \approx 10^{301}$ **variables!!!**
(thats MORE than the number of elementary particles in the universe)



What to do???

Trying to store all these complex numbers would not be possible on any conceivable classical computer...

Quantum Computer



Simulating a quantum system on a classical machine seems unfeasible.

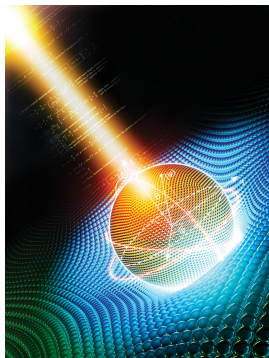
→ Turn the perspective around!

Richard Feynman, **1982**:

Proposes the first basic model for a quantum computer, capable of efficiently simulating a quantum system.



Quantum Computer



Central idea of quantum computing:

- Encode information in a superposition.
 N qubits $\rightarrow 2^N$ numbers
- By manipulating the N -qubit system, all 2^N numbers are affected.

But when the N -qubit system is measured, only a single pure state will be picked out.



A quantum algorithm must modify the probability distribution, such that the correct “end result” state has an (almost) 100% probability.

Quantum Computer = Über computer?



However, those 2^N numbers are just describing the superposition...

Holevo's bound (1973):

N qubits cannot carry more than N classical bits of information.



A quantum computer is most certainly **NOT** an “Über machine”...

...but provides a **completely different perspective on the same information**, by encoding it in a superposition.



Make use of this different perspective!

Quantum Computer = Über computer?



One **striking difference** of the “superposition perspective” compared to classical computation:

Reversibility

As long as no measurement has happened a quantum computation remains completely reversible.

Example:

If you compute some **hash value** on a quantum computer and DO NOT measure the hash value, you can **recreate the original state** by simply going in reverse through the algorithm.

What is a Quantum Computer?



- A quantum computer is **NOT** an “Über machine”
- Encodes information in a **superposition of states**
- Performs completely **reversible computations**
- Instead of deterministic results **gives probabilistic results**



copied from maximumpc.com

Original motivation: Simulate quantum systems efficiently.

Quantum Computing 101



How to carry out computations?

Classical bits: NOT:

1	0
0	1

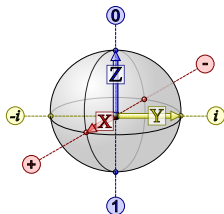
AND:

1	1	1
0	1	0
1	0	0
0	0	0

OR:

1	1	1
0	1	1
1	0	1
0	0	0

This makes no sense for qubits...



→ Need to operate on the qubit “sphere”.

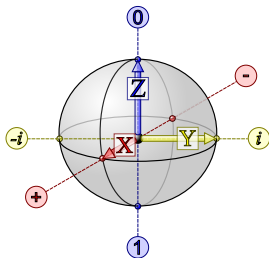
More generally, a k -qubit operator is represented by an $2^k \times 2^k$ -matrix acting on the 2^k complex numbers of the superposition.

Quantum Computing 101

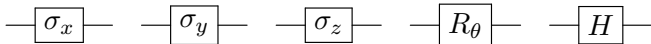


Single qubit gates: \rightarrow Operating on $\alpha|0\rangle + \beta|1\rangle$

- **Pauli- X , $-Y$, $-Z$ gate:**
180° rotation around the X , Y , Z -axis
 \rightarrow Pauli- X corresponds to NOT
- **Phase change gate R_θ :** $|1\rangle \mapsto e^{i\theta}|1\rangle$
- **Hadamard gate:**
90° rotation around X - and Z -axis




$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \begin{aligned} |0\rangle &\mapsto |+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\ |1\rangle &\mapsto |-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \end{aligned} \quad H^2 = \text{id}$$



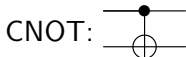
Quantum Computing 101



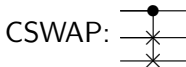
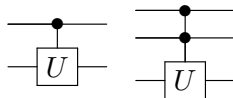
Multi qubit gates: \rightarrow Operating on e.g. $\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$

- **SWAP gate:** Exchanges two qubits 
- **Controlled gates:** An operation on a qubit only becomes active, if the other qubit is $|1\rangle$.

Examples: CNOT (“controlled NOT” — only performs NOT / Pauli-X on qubit 2 if qubit 1 is a $|1\rangle$)



controlled U :



CCNOT:

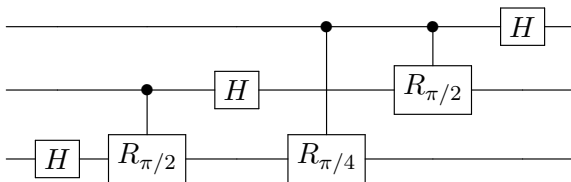


...



Quantum Computing 101

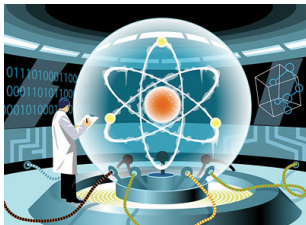
Using single qubit gates and the controlled multi qubit gates, complicated systems can be build:



→ “Quantum algorithms” = “quantum gate arrangements”



Quantum Computing 101



copied from Sarah Callihan's website

Ultimately, the idea of quantum computing is therefore to connect quantum gates in a suitable fashion while protecting the superposition between the N qubits from any external influence.

- Prepare N qubit input superposition.
- Run through the quantum gates.
- At the end one state should have almost 100% probability.
- Measure the N qubit system, i.e. the “result” pure state.



Complexity Classes

What can a quantum computer do for us that an ordinary computer cannot? → Classify problem “difficulties”.

Problem complexity: How does the number of required steps to a solution increase when more elements are considered?

Main complexity classes:

- **P** = problem solvable in polynomial time $\mathcal{O}(n)$, $\mathcal{O}(n^2)$, ...
 - Easily solveable and checkable problems

Example:

- Multiplication of two n -digit numbers takes $\mathcal{O}(n^2)$ steps.
- Checking if a number is prime.



Complexity Classes

- **NP** = “nondeterministic polynomial” problems
 - ➔ Hard to solve, but easy (in poly. time) to check problems

Example:

- Find route in a graph that visits each node once.
- **NP-complete** = NP problems to which all other NP problems can be reduced (in polynomial time)
 - ➔ Solving one NP-complete means solving all NPs!

Example:

- Sudoku
- **NP-hard** = problems at least as hard as hardest NP problem

Example:

- Traveling salesman problem: Find shortest route that visits each node in a graph with distances.



Complexity Classes

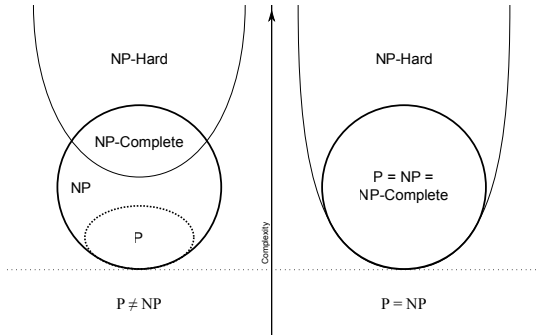


Want to make some quick & easy **\$1,000,000**?

→ Prove that $P \neq NP$ or $P = NP$!

“ $P = NP$ ” implies:

For a problem that can be easily verified, we should find the solution efficiently as well.





Complexity Classes

What about a quantum computer?

- **BQP** = “bounded error quantum polynomial time” problems
→ In polynomial time solvable on a quantum computer
- **QMA** = “Quantum Merlin Arthur” problems
→ Quantum analogue of NP problems

Big question: How are those related to the classical classes?

Bigger? Smaller? Equal?

Equivalently: Is a quantum computer fundamentally better / worse / equal to a classical machine?



Modern Everyday Cryptography

We encounter encryption on daily basis: ATM machines, SSL connections in the web, Blackberry Enterprise communication, ...

- **Symmetric ciphers:** Encryption key = Decryption key
Quite fast, but you can't transmit the key over an unsecure line.
Example: AES, DES, Triple-DES, Serpent, Twofish, ...
- **Asymmetric ciphers:** Encryption key \neq Decryption key
Very slow, but you can send out the encryption key without worry.
Example: RSA, Elliptic curve cryptography, ...



In practice: During initialization use an asymmetric cipher to exchange the keys for a subsequent symmetric cipher.

→ **Security entirely rests on the asymmetric cipher!!!**



Modern Everyday Cryptography

In the main asymmetric cipher, the **RSA algorithm**, a message is raised to an **exponent**, modulo a **composite number N** whose **factors are not known**.

Breaking RSA is *at least as easy as* factorizing N back into its two prime factors.

→ **integer factorization problem**

Best known classical method (“general number field sieve”):

$$\mathcal{O}\left(\exp\left(b^{\frac{1}{3}}(\log b)^{\frac{2}{3}}\right)\right) \rightarrow \text{exponential in number of bits } b$$

NOTE: Doesn't imply a NP problem, but no better algorithm known!
But suspected to be an NP problem...



Shor's Algorithm



Peter Shor, **1994**:

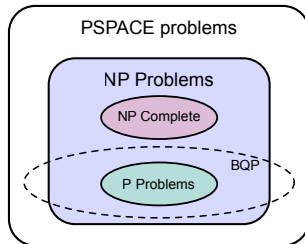
Quantum algorithm that solves both
integer factorization and **discrete logarithm**
problem in polynomial time $\mathcal{O}(b^3)$!



Breakthrough in theoretical computer science!

Both problems therefore are in the
quantum complexity class **BQP**!

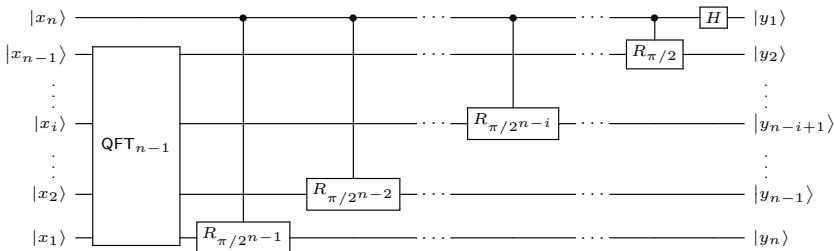
suspected relation to other classes →



Shor's Algorithm



The algorithm's efficiency is primarily based on the idea of “**period finding**” and the efficiency of the “**quantum Fourier transformation**”:



Requires $\mathcal{O}(N^2)$ quantum gates for N input qubits.

Post-Quantum Cryptography



Are there any public key ciphers not affected by this discovery?

- **Lattice-based ciphers:** relatively efficient, security rests on two problems:
 - Shortest vector problem: Given a basis for a lattice, find the shortest vector in that lattice.
 - Closest vector problem: Given a basis for a lattice and vector \vec{v} , find the closest vector approximating \vec{v} within the lattice.
- **McEliece cryptosystem:** very fast, but key sizes are very large (500+ KB), uses random components during the encryption



In the long run we may need to upgrade a few things...

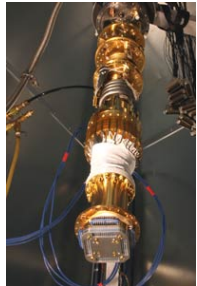


Current State of the Art

All the theory is nice, but what can we actually do?

Foremost problem: **Keeping the system shielded from environment!**

- **1995:** first quantum logic gate CNOT
- **2000:** 7-qubit system; uses nuclear magnetic resonance (NMR) to manipulate particles in acid
- **2001:** Shor's algorithm tested on 7-qubit system
- **2005:** First qubyte / 8-qubit system
- **2006:** 12-qubit system
- **2007:** 16-qubit system from D-Wave
- **2009:** First 2-qubit solid state quantum computing chip



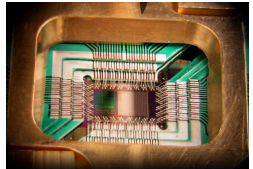
(c) D-Wave Systems, Inc.



Current State of the Art

- **Early 2011:** D-Wave announced the first **128-qubit** quantum processor!?!?
→ **May 21, 2011:** First customer contract:

LOCKHEED MARTIN



(c) D-Wave Systems, Inc.

It is **not clear at this point what this new system can do** (criticism has been raised), but quantum computers are *certainly a technology to watch out for*. The implications for current ubiquitous encryption methods are profound, but solutions are available.

However: One always needs to look into the specific details!
A general-purpose code-breaker is NOT just around the corner!



Summary

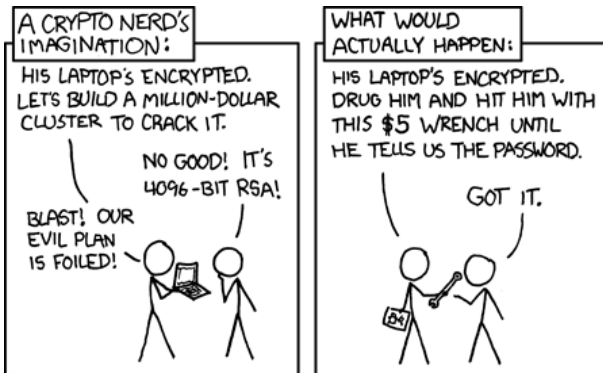
So what have we discussed?

- The principle of quantum superposition.
 - Qubits and the fundamentals of quantum computing, like basic quantum gates.
 - Problem difficulties and complexity classes.
 - Day-to-day cryptography.
 - Shor's factorization algorithm and its implications.
 - The “State of the Art” in Quantum Computers.
(but note certain criticism for some of the recent announcements)
- Entirely new cryptographic systems may want to take the long-term possibilities of quantum computing into account as well.



...Security Demolished?!

In the end, though, security always depends on a lot of other factors as well...



(c) xkcd Webcomic #538

Still time?!



How do RSA and the Shor algorithm actually work?



RSA cipher in 2 minutes

Key generation:

- p, q distinct prime number
- $n = p \cdot q$ is the “public key modulus”, set $\phi(n) = (p - 1)(q - 1)$
- Choose “public key exponent” e such that $1 < e < \phi(n)$ and $e, \phi(n)$ are co-prime (i.e. $\gcd(e, \phi(n)) = 1$).
- private key: $d = \frac{1}{e} \bmod \phi(n)$, meaning $d \cdot e = 1 \bmod \phi(n)$

Encryption:

- have: public key (n, e) and message m where $0 < m < n$
- encrypted: $c = m^e \bmod n$

Decryption:

- have: private key (n, d) and encrypted message c
- decrypted original: $m = c^d \bmod n$



Breaking the RSA cipher

Long story short, in order to break the RSA cipher we need to compute the **private exponent** $d = \frac{1}{e} \bmod \phi(n)$ from the **public key** ($n = pq, e$).

problem:

Compute $(p-1)(q-1)$ from $n = pq$



Integer factorization problem:

Compute p, q from $n = pq$

Suspected to be an NP problem, at least no efficient factorization algorithm is known. → Blast, our evil plan is foiled!



Reducing to period-finding

Idea: Reduce the factorization problem to a period-finding problem.

Given modulus N and a number a with $1 < a < N$, consider powers of a modulo N .

Example: Consider powers of 2 modulo 15:

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	...	powers
1	2	4	8	16	32	64	128	256	512	...	values
1	2	4	8	1	2	4	8	1	2	...	modulu 15

→ The period r is here 4, since $2^n = 2^{n+4} \pmod{15}$.

Big question:

Given a and N , what is the period length r ?



Shor's algorithm

Shor's algorithm can be split in two parts:

- The **quantum part** provides an efficient method to compute the period r of a number a modulo N .
- The **classical part** uses the period to solve the factorization problem of N .



Shor's algorithm: Classical part

How does the classical part work?

- Pick a random number a , such that $1 < a < N$.
- If $\gcd(a, N) \neq 1$ we've found a non-trivial factor of N . DONE!
- Compute the period r of a modulo N with the quantum part.
- Check if r is odd, pick another a and START OVER!
- Check if $a^{\frac{r}{2}} = -1 \pmod{N}$, pick another a and START OVER!
- Then $\gcd(a^{\frac{r}{2}} \pm 1, N)$ is a non-trivial factor of N . DONE!



Shor's algorithm: Quantum part

How does the quantum part work (outline)?

The idea is that the period r is a global property encoded in the numbers $a \bmod N$, $a^2 \bmod N$, $a^3 \bmod N, \dots$



Create a superposition of those numbers and perform a quantum Fourier transformation to approximate the period r .