QoSCOM: 86. K-Stammtisch

Quantum-Safe Cryptography

Marcellus Buchheit
mabu@wibu.com
President and CEO
Wibu-Systems USA Inc.

Overview

- What is Quantum Computing?
- Introduction into Encryption Algorithms
- How Quantum Computing can crack today’s algorithms
- Post-Quantum Cryptography (PQC): NIST Initiative

Phrases and Definitions

- Mostly used: **Post Quantum Cryptography** (PQC)
- Alternative: **quantum Proof**, **quantum-safe**, **quantum-resistant**
- https://en.wikipedia.org/wiki/Post-quantum_cryptography

Q-Day or **Y2Q**: time when Quantum Computer can crack today's secure algorithms

I prefer **quantum-safe** in this presentation over PQC:

- We need **now** safe cryptography, not after Q-Day actually will happen
  - We have secret information **today** which needs to **stay** secret in **future**
    - Hackers could save encrypted data today and crack it tomorrow after Q-Day
  - Think about "potentially smarter" aliens already exist today

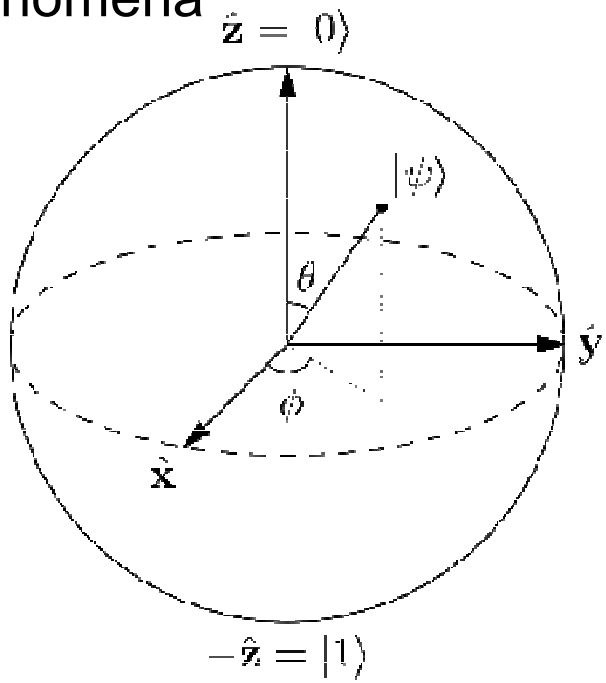
What is Quantum Computing?

Quantum Computing based on Quantum-mechanical phenomena

- **Superposition** (Superposition)
  quantum states of particles added together
  as a state
- **Entanglement** (Quantenverschränkung)
  quantum state of each particle cannot independently
  be described of the state of the others

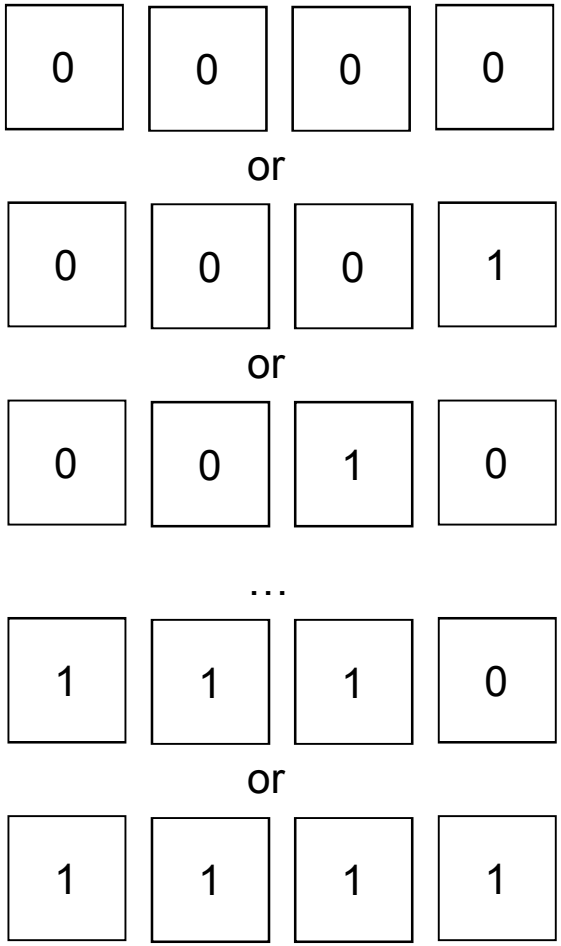
Implementation Techniques: **Qubit**

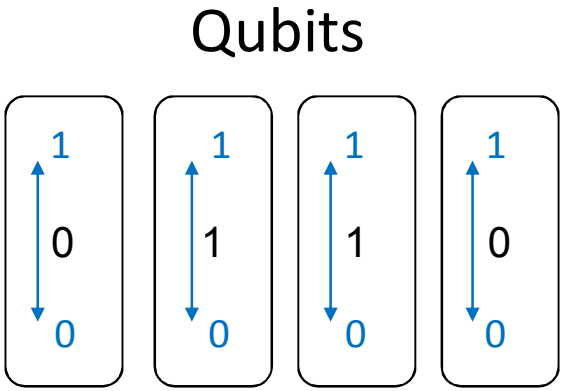
- Like computer bits can be 0 or 1
- But can also be the other value at same time

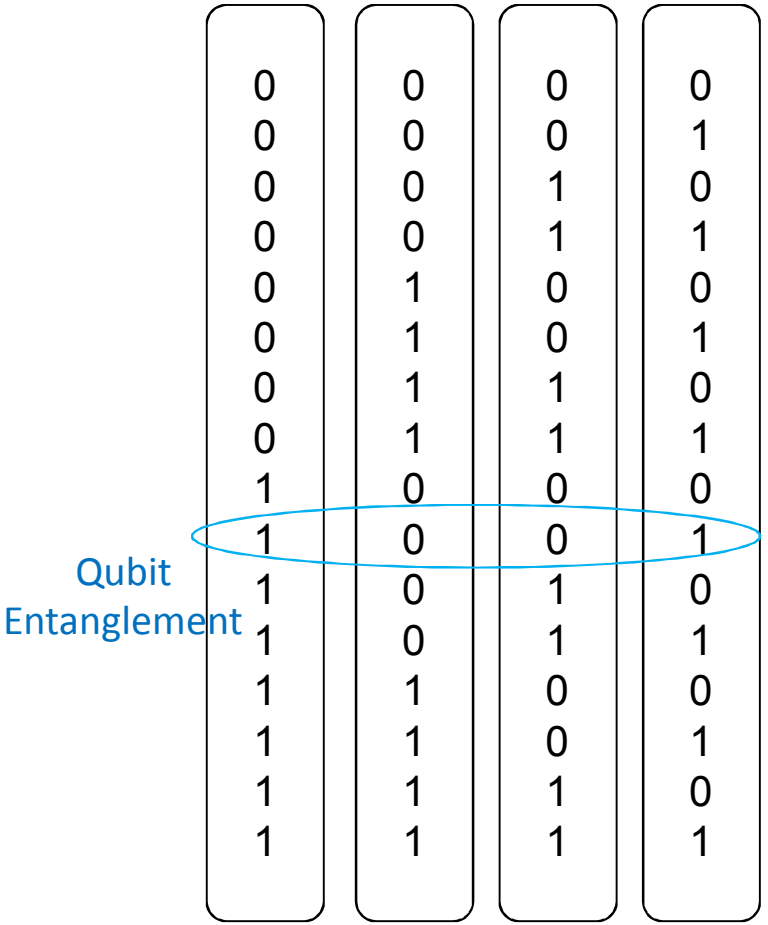


Bloch sphere as
representation of a qubit:
the fundamental building block
of quantum computers

4 Bits versus 4 Qubits







Qubits

- An array of N qubits can have **2^N** states at same time
  - In contrast N **bits** can only have **one** state in range 0 to 2^{N-1}
- **While *reading* and *writing* a qubit array: there is only *one* state**
  - A qubit can only be read *one* time, after that all other states are *lost*
  - Which state is actually read is a *probabilistic* experience
- **Quantum (logic) Gates connect qubits to do specific operations**
  - Same functions as binary gates but very different implemented
    - Names: *Hadamard*, *Pauli-X*, *Pauli-Y*, *Phase-shift*, *Toffoli*, *Fredkin*, *Ising* etc.
- Quantum computers will have much more Quantum Gates than Qubits

Usage of Quantum Computing

Solving very complex problems of today's sciences

- Understanding global weather
- Detecting and understanding behavior of new materials
- Simulating of global human society
- Quantum search: high speed in large data sets

(Negative) Side effect: cracking of cryptographic algorithms (topic of this presentation)

Related topics: quantum cryptography, quantum teleportation

Quantum Supremacy (or **Quantum Advantage**): the goal of demonstrating that a programmable quantum device can solve a problem that classical computers practically cannot.

https://en.wikipedia.org/wiki/Quantum_supremacy

Research and Development in Quantum Computing

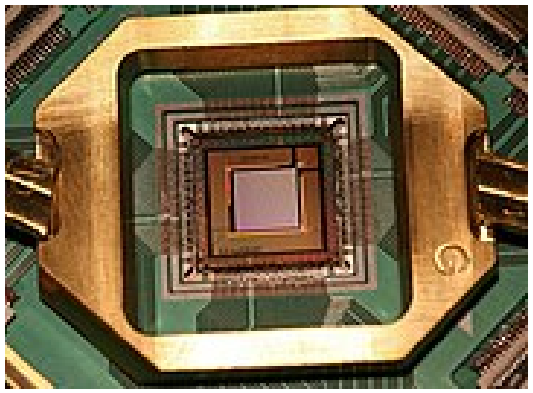
- All large IT research centers are involved
- Microsoft, IBM, Google, D-Wave Systems, Toshiba, Intel, HP, Alibaba, 1WB
- D-Wave Systems: first commercial Quantum Computing specialized company
- Billion-$ public research money by USA, EU, China etc.
- Architectures: Gate-based (Microsoft) versus Adiabatic (D-Wave)

- Challenges: increase number of Qubits
  - About 100 are working today, we needs many thousands or more
- Distortion in Qubits: Avoid or reduce noise (*Quantum Error Correction*)

Available D-Wave Systems

D-Wave, located in Burnaby, BC, Canada (founded 1999, 220+ employees)

	D-Wave One	D-Wave Two	D-Wave 2X	D-Wave 2000Q[53][54]	"Next-gen"[52]
Release date	May 2011	May 2013	August 2015	January 2017	Mid 2020
Code-name	Rainier	Vesuvius	W1K	W2K	
Qubits	128	512	1152	2048	>5000
Couplers[55]	352	1472	3360	6016	>37500
Josephson junctions	24,000	?	128,000	128,000	
I/O lines / Control lines	?	192	192	200[56]	
Operating temperature (K)	?	0.02	0.015	0.015	
Power consumption (kW)	?	15.5	25	25	
Buyers	Lockheed Martin	Lockheed Martin	Lockheed Martin	Temporal Defense Systems	
		Google/NASA/USRA	Google/NASA/USRA	Google/NASA/USRA[57]	
			Los Alamos National Laboratory		

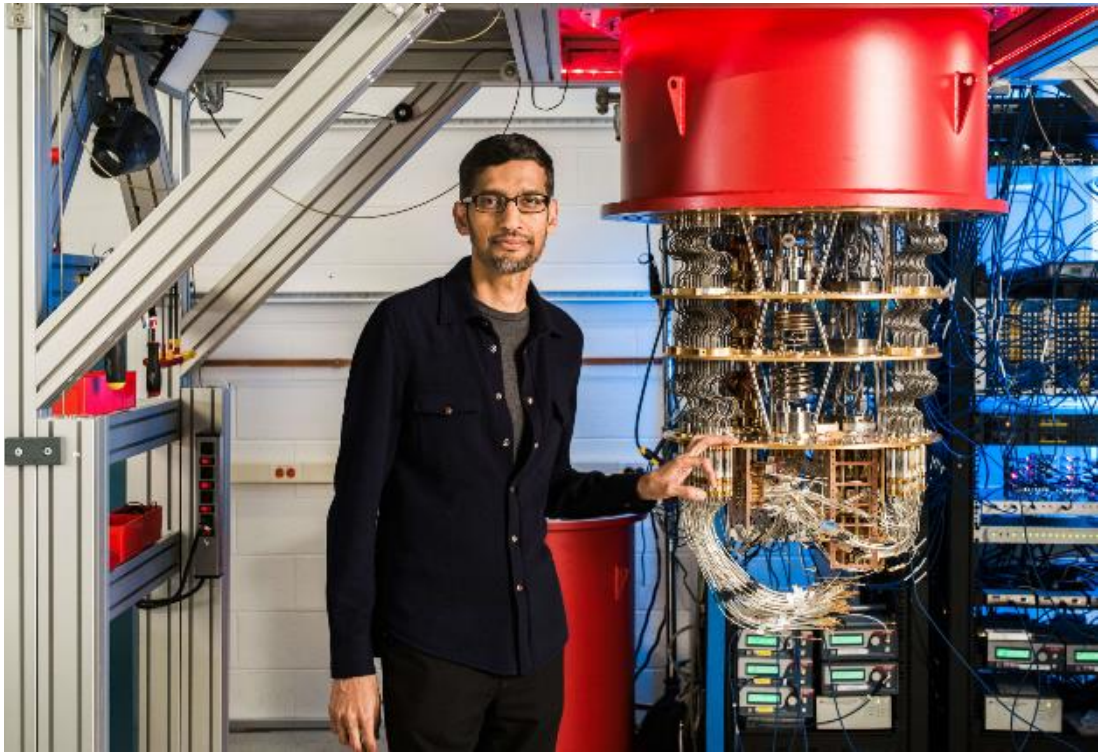


D-Wave 2X with 1152 Qubits

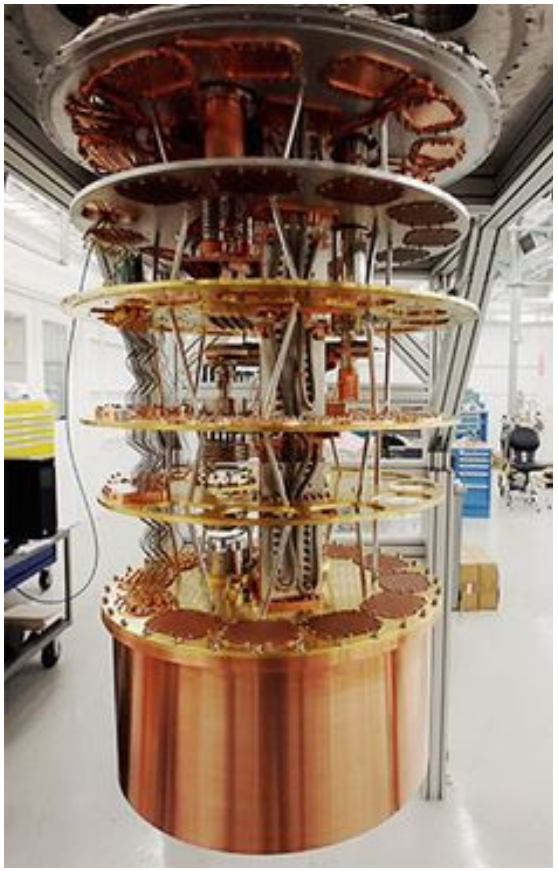
Quantum Computing Implementation Visions

- **First quantum computers are very expensive and very special hardware**
  - Superconducting, Trapped Ion, optical Lattices
  - Spin-based quantum dots, carbon nanospheres
  - Molecular magnet, nuclear spin, rare-earth metal ion doped crystals
- **Quantum gate arrays**
  - flexible quantum gates around qubits
- **Implementation will start in special datacenters**
  - Similar as supercomputers today
  - Can be addressed by “anyone” through cloud computing
- **Future mass implementations could be co-processors in desktop computers**
  - Relatively small number of qubits, still general-purpose use

Sommer 2019: Google Quantum Computer



Google CEO Sundar Pichai with a quantum computer at Google laboratory in Santa Barbara, California

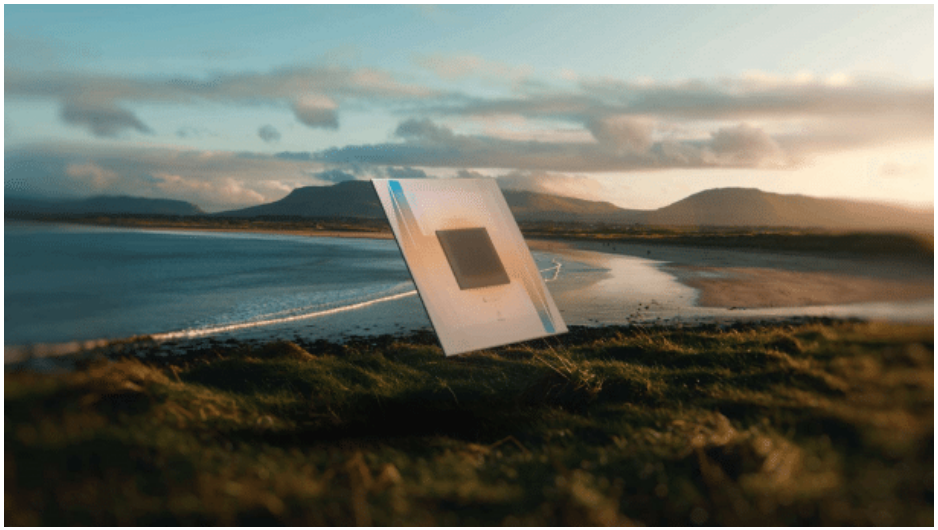


https://www.technologyreview.com/s/544421/googles-quantum-dream-machine/

Google Quantum Chip “Willow”

Presented December 2024

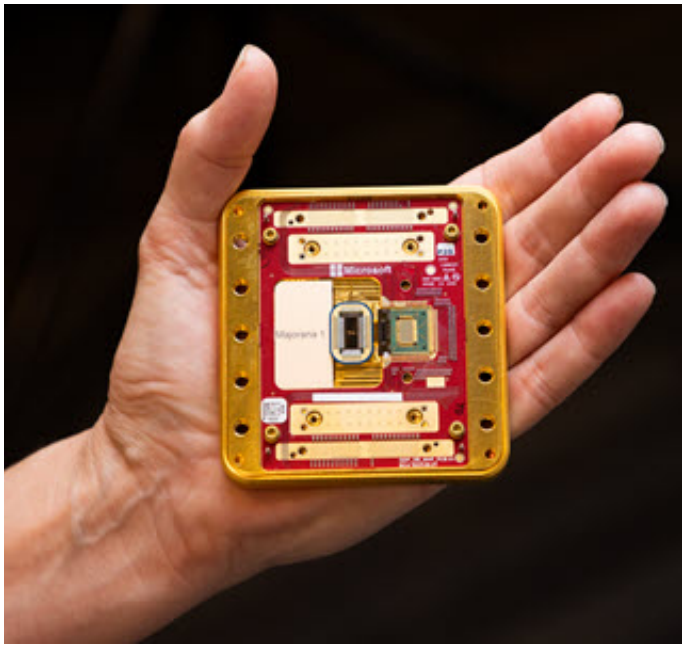
- 105 qubits
- Focused on error reduction and correction and high performance
- Prototype for future large-scale quantum computing
- https://en.wikipedia.org/wiki/Willow_processor



Microsoft Quantum Chip “Majorana”

Presented February 2025

- Large Scale prototype: Design can fit up to 1 million qubits
- Focused on error prevention by new qubits design (“Majorana Quasiparticles”)
- https://news.microsoft.com/azure-quantum/



Symmetric Encryption / Decryption

Shared **secret** key

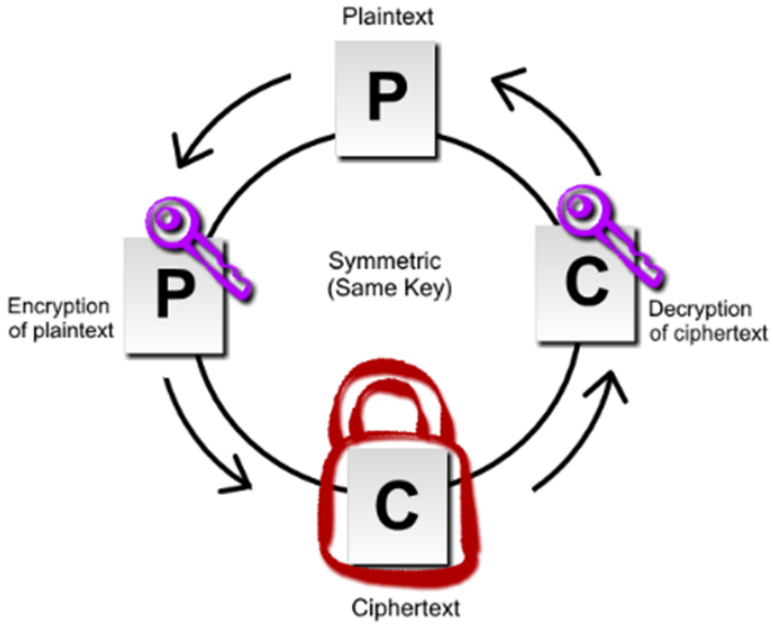
- Same key for encryption and decryption

Faster than asymmetric cryptography

Examples: DES, Triple DES, Blowfish, **AES**

Challenges

- Key exchange: needs secret path
- Key storage: vulnerable for hacks
- Authentication not possible



Asymmetric Encryption / Signing

Private/public key pair

- **Private** Key
- **Public** Key

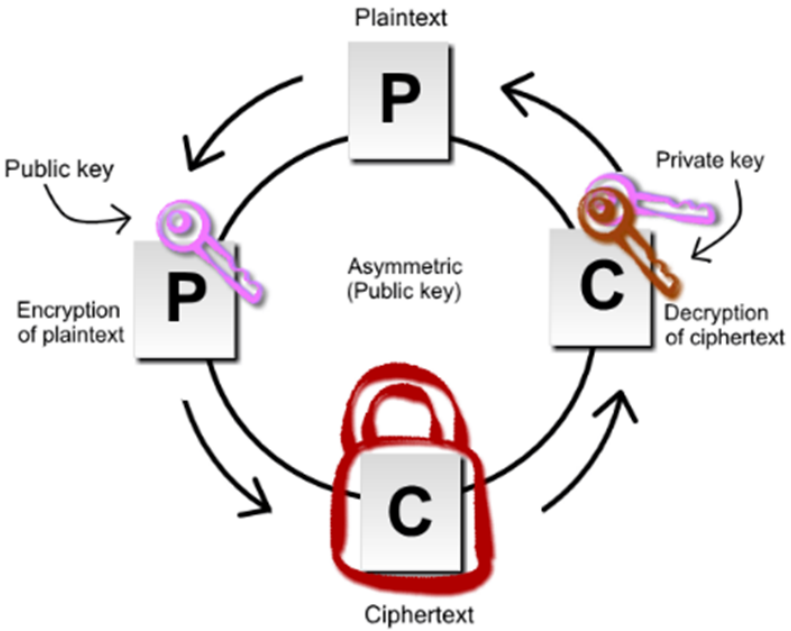
Examples: RSA, ECC (elliptic curve)

Asymmetric encryption

- Sender **uses receiver's public** key for encryption
- Receiver uses his **private** key for decryption

Signatures

- Sender uses **private** key to sign plain text data
- Receiver uses **sender's public** key to verify the plain text data signature



Asymmetric Security Principles

Private key created first and kept secret by creator

- **Public key** can be easily created out of Private key
- But: *practically impossible* to recreated private key from public key

Public key can be *freely distributed* to encrypt files or verify signatures

Symmetric operations using private and public key:

- **Encryption**: Dec(Enc(Text, public key), private key) = Text
- **Signatures**: Verify(Sign(Text, private key), public key) = yes / no

Asymmetric Algorithms based on complex math problems

DSA (Digital Signature Algorithm)

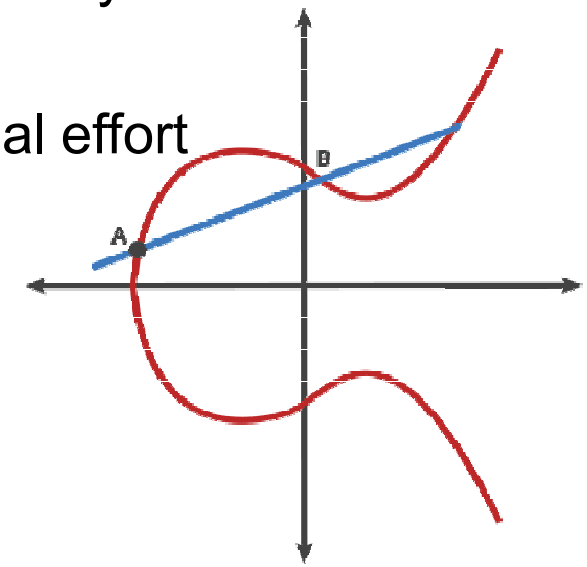
- $y = n^x$: use x as private key, y as public key
- y can be easily calculated from x but x not from y

RSA (Rivest–Shamir–Adleman)

- private key based on two prime numbers, product is public key
  - Multiplication of two prime numbers is easy
  - Finding the factors of the multiplication needs exponential effort

ECC (Elliptic Curve)

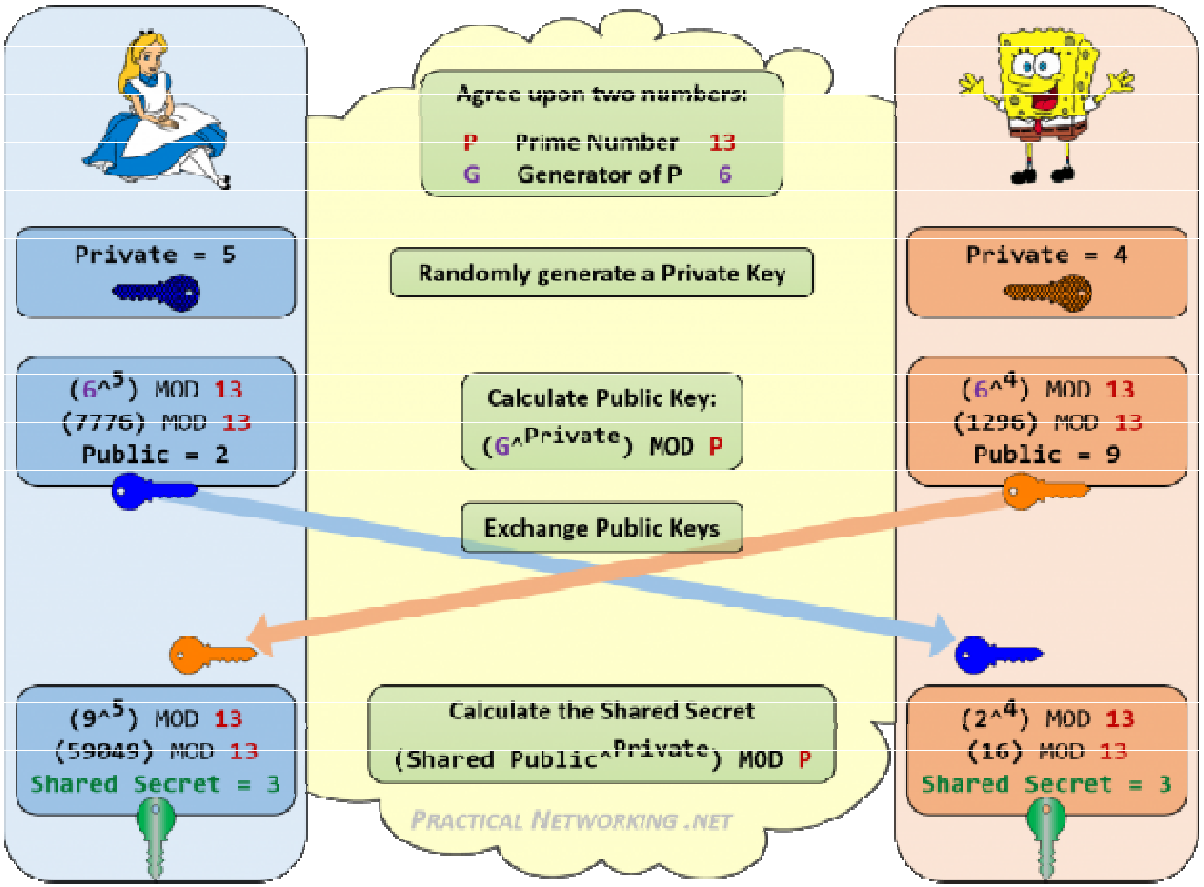
- Determine a point at curve $y^2 = x^3 + ax + b$
- Use A as private key, calculate B as public key from A
- https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/



Key Encapsulation/Exchange Using Diffie Hellman

- *Asymmetric* crypto algorithms much **slower** than *symmetric* crypto algorithms
- Unusable to *larger amount of data* (document texts, pictures, website etc.)
- **Solution**
  - Encrypt and decrypt data with (same) symmetric key
  - Sender and receiver can calculate this key using asymmetric encryption
  - Published 1976 by *Whitfield Diffie* and *Martin Hellman*
- **No actual transfer of symmetric key required**
  - Side A and Side B exchange random session code and their public keys
  - Side A: symmetric key_A = DH(private key_A, public key_B, session code)
  - Side B: symmetric key_B = DH(private key_B, public key_A, session code)
  - Result: symmetric Key_A and symmetric Key_B are identical

Key Encapsulation/Exchange Using Diffie Hellman (II)



Source: https://www.practicalnetworking.net/series/cryptography/diffie-hellman/

Practical Usage of Asymmetric Encryption: Access Control

Access Management using public key (“certificate”)

- In this case the key is not *real* public but type of *confidential* information

Principle

- System to be accessed sends *random session key*
- Access system *encrypts* session key by *public key* (**certificate**)
- Accessed system *decrypts* encrypted session key by *private key* (never leaves this system)

Advantages

- No one in public can simulate *faked* accessed system due missing private key
  - Every access starts with *different* random session key: former session key *invalid*

Practical Usage of Asymmetric Encryption: Block Chain

- **Chain of transactions**
  - Bitcoin: payments with digital coins
  - Each transaction information united with hash of predecessor transaction
  - New hash is created and signed with private key of transaction operator
- **Result**
  - No transaction can later be modified
  - No transaction can be removed in chain
  - No transaction can be later added
  - All transactions can be validated by anyone with public key of operator
- **Distributed Block Chain (Bitcoin)**
  - every location can have automatic update of the chain

How can Quantum Computing crack cryptography?

Symmetric Algorithms (AES)

- Grover's Algorithm: Cracking a "complex black box" by $O(\sqrt{2^n})$
  - TripleDES with 112-bit key mutated Post-Quantum to DES 56-bit security
  - AES-256 quantum-safe still same security as AES-128 today
  - Similar results with hash algorithms: SHA-2 and SHA-3 are safe

Asymmetric Algorithms (RSA, ECC, DSA)

- Shor's Algorithm (Peter Shor, 1994)
  - Highly parallel integer factorization
- Quantum Annealing (since 2002)
  - Metaheuristic finding for finding global minimum of specific functions

How can Quantum Computing crack RSA 2048?

Today's "classic THz computer" (trillions of operations per second):

- 10^{34} steps
- ~317 trillion years

Shor's algorithm optimizes factorization by fast multiplications

- runs to factor integer N in O(log(N))
- Needs Quantum Gates in order $O((\log N)^2(\log \log N)(\log \log \log N))$ for multiplication
- 2001 first practical demonstration: factoring 15 into 3 x 5 with 7 qubits

Shor’s Algorithm principles

- Developed 1994 by *Peter Shor*
- Challenge: Find factors **N=p*q**
- Calculate in superposition exponential amount of data (“super parallelism”)
- Finally do a *Fourier transform* to find period of sequence
- Use period to find factors

Shor’s algorithms is not specific for Quantum Computing

- But not efficient (practically unusable) on classic computers

Source: https://en.wikipedia.org/wiki/Shor%27s_algorithm

Shor's Algorithm Advances

- Fowler et al, **2012**: **1 billion** qubits for 2048 RSA (0.1% gate error rate)
- O'Gorman et al, **2017**: low-noisy **230 million** qubits
- Gidney, Ekera, **2019**: low-noisy **20 million** qubits
- Alternative with error-free Quantum computing:
  error-free **4099** logical qubits, 100 million gates

General Progress by Google in last 5 years

- **Sommer 2019**: Quantum computer with **72** qubits and **0.6%** error rate
- **December 2024**: Quantum computer with **105** qubits and **0.1%** error rate

Summary: How can Quantum Computing crack RSA 2048?

Today's "classic THz computer"
(trillions of operations per second):

- 10^{34} steps
- ~317 trillion years

Shor's Algorithm on "quantum MHz computer"
(millions of operations per second):

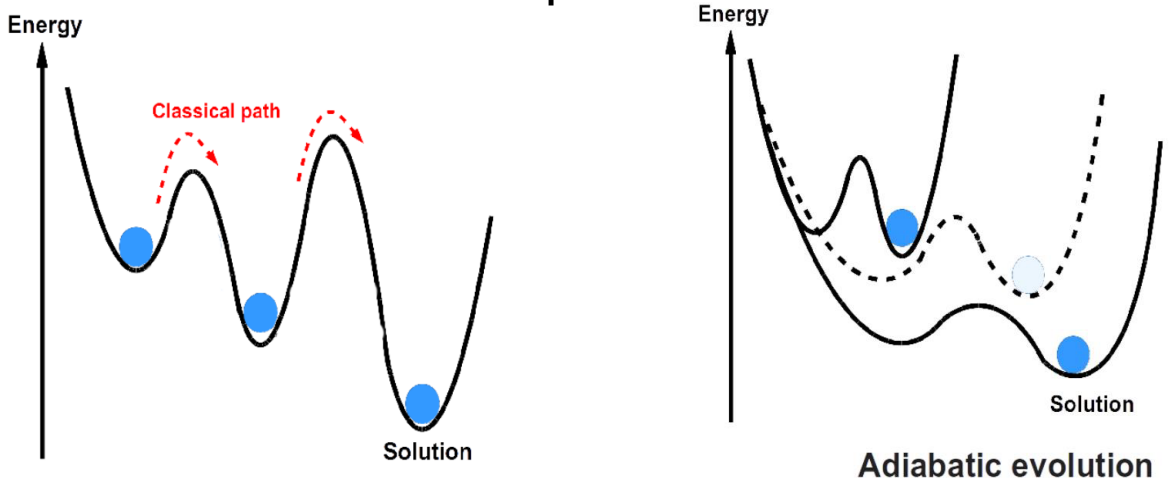
- 10^{7} steps, error-free 4099 logical qubits, 100 million gates
- ~10 seconds

Alternative to Shor's Algorithm: Quantum Annealing

- **Quantum Annealing** (Quantum-Ausglühen)
  process of finding a global minimum of given objective function

Quantum Computer codifies optimization problem into a physical system

- Hamiltonian Path: graph-mathematical problem
- Optimal solution to optimization problem = minimum energy state of the system



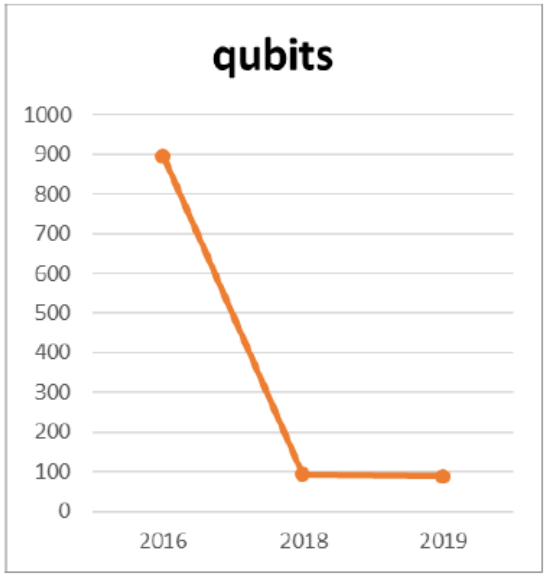
Quantum Annealing advances

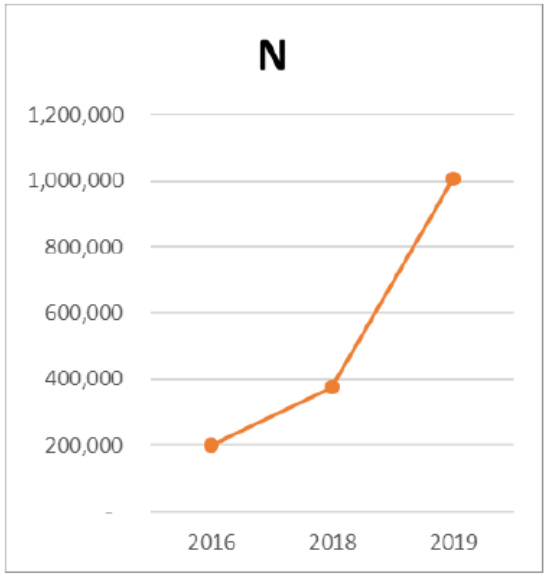
- **2002**: Burges (Microsoft Research) formulates the method
- **2016**: factoring 200,099 (18 bit) with **897** qubits
  - Implementation by D-Wave (Adiabatic architecture)
- **2018**: Jiang et al (Purdue University): factoring 376,289 (19 bits) with just **94** qubits
- **2019**: Peng et al (Shanghai/Beijing): factoring 1,005,973 (20 bits) with just **89** qubits
  - RSA-768 needs 147,454 qubits of today's available quality

Quantum Annealing advances (II)

Graphic motivation

year	qubits	number
2016	897	200,099
2018	94	376,289
2019	89	1,005,973





Solution: Quick-as-possible replacement of RSA, ECC and DSA

NIST global leader for standardizing cryptographic algorithms

- NIST = National Institute of Standards and Technology (USA, founded 1901)
- Standardized DES, AES, SHA-1, SHA-2, SHA-3, DSA

NIST started post-quantum cryptography (PQC) initiative

- Started in April 2016, deadline for proposal ended November 2017
- first round had 69 submissions
- January 30, 2019: reduced to 26 algorithms for second round
- July 2020: reduced 7 algorithms and 8 alternatives for 3rd round
- July 2022: first group of winners, 4 candidates
- August 2024: first release, 3 algorithms **ML-KEM**, **ML-DSA**, **SLH-DSA**
- August 2024: 4th algorithm announced, **FN-DSA**

Quantum-Safe Key-Exchange Algorithm ML-KEM

- Based on CRYSTALS-Kyber algorithm candidate
- Defined by NIST in **FIPS 203** (FIPS = *Federal Information Processing Standard*)
- Original information: https://pq-crystals.org/kyber/index.shtml
- **Lattice** (Gitter)-based cryptography
  - Theory introduced in 1996 by *Ajtai Miklós* (IBM)
  - First public encryption scheme in 2006 by *Oded Regev* (NY State University)
  - *Lattice* = mathematical group in which addition and subtraction of two points produces another lattice point
- Defeats *Shor's Algorithm* by dropping factorization during encryption

Quantum-Safe Signature Algorithm ML-DSA

- Based on **CRYSTALS-Dilithium** algorithm candidate
- Original information: https://pq-crystals.org/dilithium/index.shtml
- Defined by NIST in **FIPS 204**
- Like ML-KEM Lattice (Gitter)-based cryptography

Quantum-Safe Signature Algorithm SLH-DSA

- Based on **SPHINCS+** algorithm candidate
- Original information: https://sphincs.org/
- Defined by NIST in **FIPS 205**
- Alternative to ML-DSA in case this is shown vulnerable
- Hash-based, stateless algorithm
  - Better understood in the cryptography world than Lattice-based
- For quantum-safe algorithms unusual short key size but long signatures
- Much slower than ML-DSA with today's hardware

Quantum-Safe Signature Algorithm FN-DSA

- Based on **FALCON** algorithm candidate
- Original information: https://falcon-sign.info/
- Will be defined by NIST in **FIPS 206**
- Alternative to ML-DSA and SLH-DSA in case they are shown vulnerable
- Slower than ML-DSA but much faster than SLH-DSA
- Keys shorter than ML-DSA but much longer than SLH-DSA
- Like ML-KEM and ML-DSA Lattice (Gitter)-based cryptography
  - Important variant: “FN” standard for “Fast-Fourier Transformation”
- Caveat: Requires *floating points operations*
  - leads to more security challenges: FP execution speed typically *not constant*

Key Exchange: Compare Algorithms

Algorithm	Standard	Public Key Length (bytes)	Secret Key Length (bytes)
RSA 2048	(SP 800-57)	256	256
RSA 4096	(SP 800-57)	512	512
ECDH		32	32
ML-KEM	FIPS 203	800 / 1184 / 1568	1632 / 2400 / 3168

Key Exchange: Performance (OpenSSL)

Algorithm	Standard	Keygen/s	Encaps/s	Decaps/s
RSA 3072	(SP 800-57)	5.0	23,688.8	1091.8
ECDH (256 bit)	Curve25519		27,541.2	27,541.2
ECDH (192 bit)	NIST		4,564.0	4,564.0
ECDH (384 bit)	NIST		1,117.4	1,117.4
ML-KEM-512	FIPS 203	95,681.7	122,685.2	115,464.4
ML-KEM-768	FIPS 203	61,139.1	76,171.7	73,051.6
ML-KEM-1024	FIPS 203	46,623.3	53,260.3	51,514.8

Larger is better

Source:
https://www.linkedin.com/pulse/benchmarking-different-pqc-libraries-in-depth-analysis-coranlabs-4oevc/

Digital Signature Parameter Sizes

Algorithm	Standard	Public Key Length (bytes)	Secret Key Length (bytes)	Signature Length (bytes)
DSA	FIPS 186-5	256 / 384	256 / 384	20
RSA 2048	FIPS 140-2	256	256	256
RSA 4096	FIPS 140-2	512	512	512
ECDSA	FIPS 186-5	32	32	64
ML-DSA	FIPS 204	1,312 / 1,952 / 2,592	2,560 / 4,032 / 4,896	2,420 / 3,309 / 4,627
SLH-DSA	FIPS 205	32	64	7,856 / 17,088 / 35,664
FN-DSA	(FIPS 206)	897 / 1,793	1,281 / 2,305	690 / 1,330

Source: https://blog.cloudflare.com/another-look-at-pq-signatures/

Digital Signature Performance

Algorithm	Standard	CPU time for Signing	CPU time for Verification
ML-DSA	FIPS 204	1 (reference)	1 (reference)
RSA 2048	FIPS 140-2	80	0.4
ECDSA	FIPS 186-5	0.15	1.3
SLH-DSA (7,856)	FIPS 205	14,000	40
SLH-DSA (17,088)	FIPS 205	720	110
FN-DSA	(FIPS 206)	3	0.7

Smaller is better

Source: https://blog.cloudflare.com/another-look-at-pq-signatures/

Summary

- Quantum computers in research levels are **already reality**
- Quantum computers today **not powerful enough to crack** asymmetric encryption
- This can change in near future and will very likely become reality one day
- NIST **standardizes three algorithms** so far, one is preparation
  - Quantum-safe algorithms for key exchange very promising
    - Caveat: *longer keys*
  - Quantum-safe algorithms for signature are still *controverse* regarding *performance* and *security* (especially traditional non-quantum cryptoanalysis)