Quantum computing and post-quantum cryptography

a gentle overview

Andrew Savchenko

FOSDEM 2017

Outline

1. Quantum computing
2. Impact on cryptography
3. What we can do (using free software)

Disclaimer

- Do not expect full strictness and completeness of this talk!
- It intends to be a short overview of the subject.
- You will encounter some equations :)

Terminology

- Classical cryptography — a *usual* cryptography, designed to withstand cryptanalysis using classical computers

- Quantum cryptography has *nothing* to do with post-quantum cryptography.
  - It uses quantum mechanical properties of the matter for crypto applications, e.g.:
    - secure key distribution using entangled particles
    - protection from data copying
  - Requires a very dedicated hardware and connection lines

- Postquantum cryptography — a cryptography *resilient* to quantum computing

Terminology

- Classical cryptography — a *usual* cryptography, designed to withstand cryptanalysis using classical computers

- Quantum cryptography has *nothing* to do with post-quantum cryptography.
  - It uses quantum mechanical properties of the matter for crypto applications, e.g.:
    - secure key distribution using entangled particles
    - protection from data copying
  - Requires a very dedicated hardware and connection lines

- Postquantum cryptography — a cryptography *resilient* to quantum computing

Terminology

- Classical cryptography — a *usual* cryptography, designed to withstand cryptanalysis using classical computers

- Quantum cryptography has *nothing* to do with post-quantum cryptography.
  - It uses quantum mechanical properties of the matter for crypto applications, e.g.:
    - secure key distribution using entangled particles
    - protection from data copying
  - Requires a very dedicated hardware and connection lines

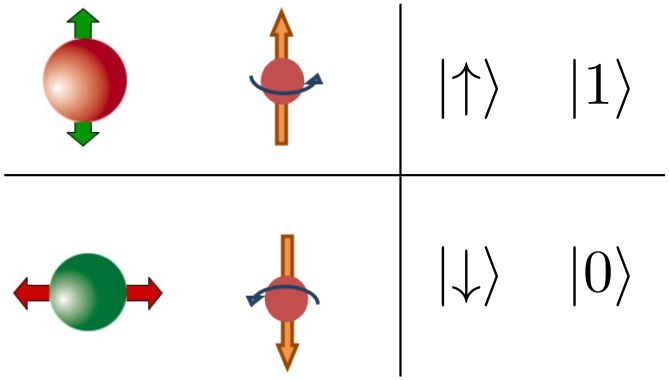
- Postquantum cryptography — a cryptography *resilient* to quantum computing

Quantum computing

Base elements:

- qubits (quantum bits)
- quantum logic gates
- quantum algorithm: sequence of quantum gates applied to qubits

Qubits



$\|\uparrow\rangle$	$\|1\rangle$
$\|\downarrow\rangle$	$\|0\rangle$

$$|Q\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

α_i — amplitude of the state i

$p(\text{"0"}) = |\alpha_0|^2$, $p(\text{"1"}) = |\alpha_1|^2$

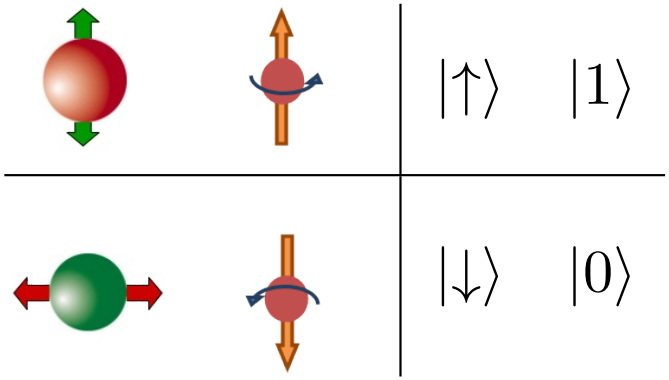
EPR paradox $\Rightarrow$ entangle them!

$$|Q_2\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle$$

$$|Q_n\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle , \quad \sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$$

- N qubits $\rightarrow 2^N$ states at once
- ...but with different probabilities

Qubits



$\|\uparrow\rangle$	$\|1\rangle$
$\|\downarrow\rangle$	$\|0\rangle$

$$|Q\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

α_i — amplitude of the state i

$p(\text{"0"}) = |\alpha_0|^2$, $p(\text{"1"}) = |\alpha_1|^2$

EPR paradox $\Rightarrow$ entangle them!

$$|Q_2\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle$$

$$|Q_n\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle, \quad \sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$$

- N qubits $\rightarrow 2^N$ states at once
- ...but with different probabilities

Qubits: capabilities

What can you do with N qubits?

- 4TB HDD $\rightarrow$ 42 qubits
- All atoms in the visible universe $(10^{80\pm2}) \rightarrow$ 273 qubits are enough!
- Manipulate individual states by affecting $|\alpha_i|^2$

Limitations:

- *Only N* bits can be extracted from 2^N states
- Random bits are read each time with different probabilities

Qubits: capabilities

What can you do with N qubits?

- 4TB HDD $\rightarrow$ 42 qubits
- All atoms in the visible universe $(10^{80\pm2}) \rightarrow$ 273 qubits are enough!
- Manipulate individual states by affecting $|\alpha_i|^2$

Limitations:

- *Only N* bits can be extracted from 2^N states
- Random bits are read each time with different probabilities

Qubits: implementation

Implementation ways:

- electron spin
- atomic nucleus
- photon
- quantum dots
- ...

Problems:

- stability: qubits tend to decay
- error correction: errors build up fast

Qubits: implementation

Implementation ways:

- electron spin
- atomic nucleus
- photon
- quantum dots
- ...

Problems:

- stability: qubits tend to decay
- error correction: errors build up fast

Qubits: implementation

Qutrits (3^n):

- more stable to decoherence
- hard to implement
- hard to manipulate

Quantum storage [1]:

- e^- coherent state transfer to ^{31}P
- storage for 1.75 s

Qubits: implementation

Qutrits (3^n):

- more stable to decoherence
- hard to implement
- hard to manipulate

Quantum storage [1]:

- e^- coherent state transfer to ^{31}P
- storage for 1.75 s

Quantum gates

Quantum logic gates:

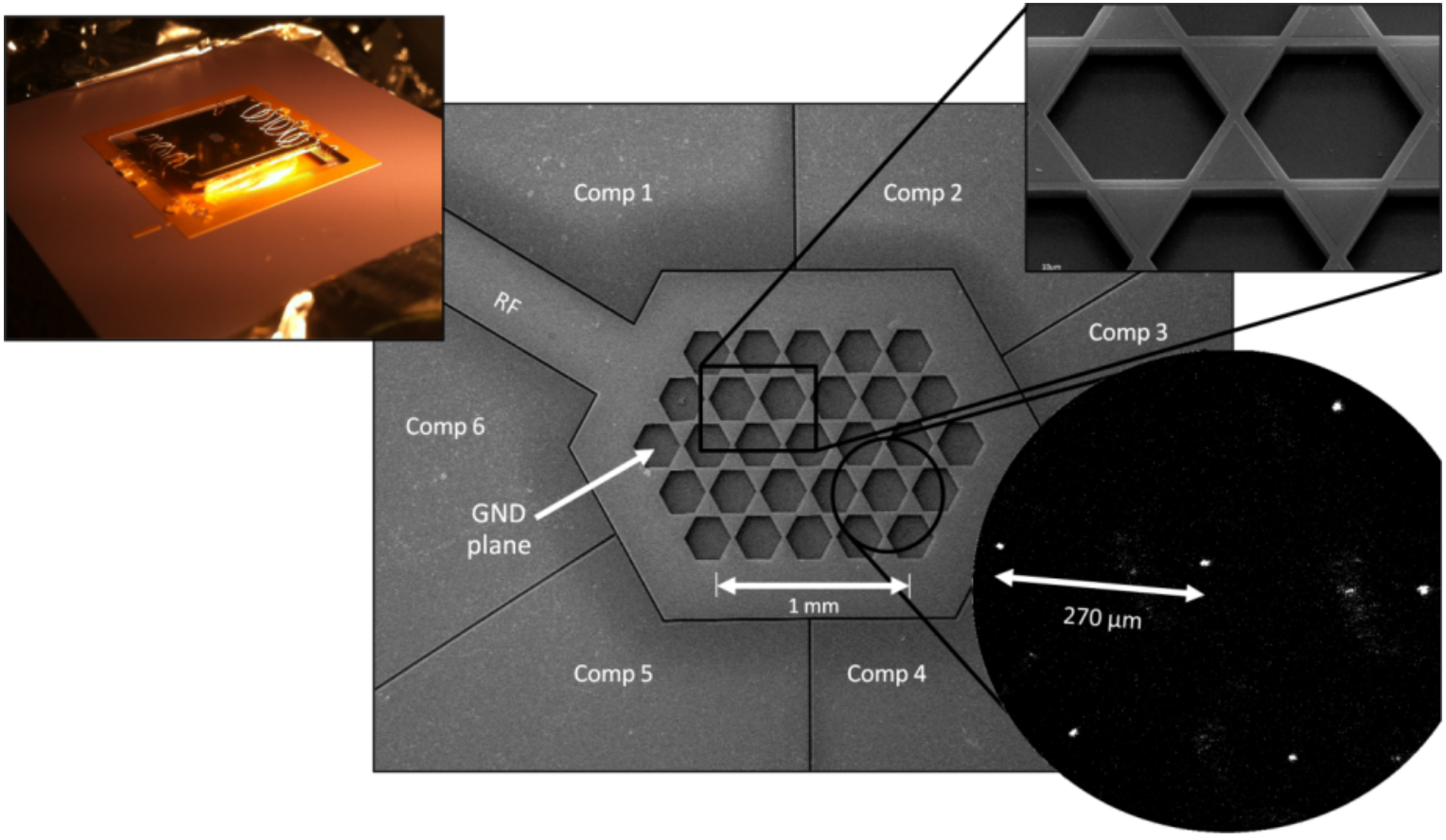
- Affects multiple amplitudes an once:
  - set with equal amplitude $f(x)$: $O(log\ N)$
- May be implemented using:
  - ion traps
  - nuclear magnetic resonance
- Provide full set of logical operations
- All quantum gates are reversible in contrast to classical gates

Quantum gates

Quantum logic gates:

- Affects multiple amplitudes an once:
  - set with equal amplitude $f(x)$: $O(log\ N)$
- May be implemented using:
  - ion traps
  - nuclear magnetic resonance
- Provide full set of logical operations
- All quantum gates are reversible in contrast to classical gates

Quantum hardware



Microchip Architectures for Scalable Ion Trap Quantum Computing [2], University of Sussex, UK

Quantum computing

Summary:

- *only* N bit can be extracted from 2^N states
- measurement (wave function collapse) is probabilistic:
  - $2 + 2 = 5$ — OK!
  - but $P(2 + 2 = 4) > P(2 + 2 = 5)$
- results must be either:
  - *checked* or
  - *repeated* several times

Further reading: “The Physics of Quantum Information” [3]

Quantum computing

Summary:

- *only N* bit can be extracted from 2^N states
- measurement (wave function collapse) is probabilistic:
  - $2 + 2 = 5$ — OK!
  - but $P(2 + 2 = 4) > P(2 + 2 = 5)$
- results must be either:
  - *checked* or
  - *repeated* several times

Further reading: “The Physics of Quantum Information” [3]

Period finding problem

$$f : Z_N \to Z$$

$$f(x + r) = f(x), \quad r = ?$$

- Classical computing: $O(N)$
- Let's apply Discrete FFT
- ...what?! Complexity: $O(N\ logN)$
- Quantum computing: $O\left((log\ N)^2\right)$ [4]

QC is a very effective DFFT machine!
$f(x)$ data can be initialized by $O(log\ N)$

Period finding problem

$$f : Z_N \to Z$$

$$f(x + r) = f(x), \quad r = ?$$

- Classical computing: $O(N)$
- Let's apply Discrete FFT
- ...what?! Complexity: $O(N\ logN)$
- Quantum computing: $O\left((log\ N)^2\right)$ [4]

QC is a very effective DFFT machine!
$f(x)$ data can be initialized by $O(log\ N)$

Shor's algorithm

Solves integer factorisation problem [5, 6]:

$$\text{for known } N \text{ find } P_1, P_2 : \; P_1 \cdot P_2 = N$$

Turn factorization problem into period finding problem!

1. If a and N are coprime:

$$a^r \equiv 1 \; mod \; N$$

2. r can be found using quantum DFFT
3.

$$\underbrace{\left(a^{r/2} - 1\right)}_{\alpha_1}\underbrace{\left(a^{r/2} + 1\right)}_{\alpha_2} \equiv 0 \; mod \; N$$

4. $P_i = gcd(N, \alpha_i); \quad p > 1/2$ [7]

Shor's algorithm

Solves integer factorisation problem [5, 6]:

$$\text{for known } N \text{ find } P_1, P_2 : \; P_1 \cdot P_2 = N$$

Turn factorization problem into period finding problem!

1. If a and N are coprime:

$$a^r \equiv 1 \; mod \; N$$

2. r can be found using quantum DFFT
3. $$\underbrace{\left(a^{r/2} - 1\right)}_{\alpha_1} \underbrace{\left(a^{r/2} + 1\right)}_{\alpha_2} \equiv 0 \; mod \; N$$
4. $P_i = gcd(N, \alpha_i); \quad p > 1/2$ [7]

Factorisation complexity

Complexity estimation for $N \sim 2^{4096}$:

Algo	Complexity	Operations
GNFS	$O\left(e^{1.9(\ln N)^{1/3}(\ln \ln N)^{2/3}}\right)$	$\sim 10^{46}$
Shor's	$O\left((\ln N)^2 (\ln \ln N)(\ln \ln \ln N)\right)$	$\sim 10^9$

GNFS — General number field sieve, the fastest classical factorisation algorithm.

Space requirements

Period finding problem:

$$q \sim O(N^2) \Rightarrow 2N \text{ qubits}$$

Year 2014: factorization of 56153:

- $56153 = 233 * 241$
- length(56153) = 16 bits
- 32 qubits are required

Factored on *4* qubits [8] at 300 K!

Space requirements

Period finding problem:

$$q \sim O(N^2) \Rightarrow 2N \text{ qubits}$$

Year 2014: factorization of 56153:

- $56153 = 233 * 241$
- length(56153) = 16 bits
- 32 qubits are required

Factored on *4* qubits [8] at 300 K!

Discrete logarithms

DSA:

- DSA time $\gtrsim$ RSA time
- DSA space $\sim$ RSA space

ECC [9]:

Algo	Bits	qubits	time
RSA	3072	**6144**	$120 * 10^9$
ECC	256	**1500 (1800)**	$6 * 10^9$

Grover's algorithm

A quantum brute-force (BF) algorithm.

Black box setup:

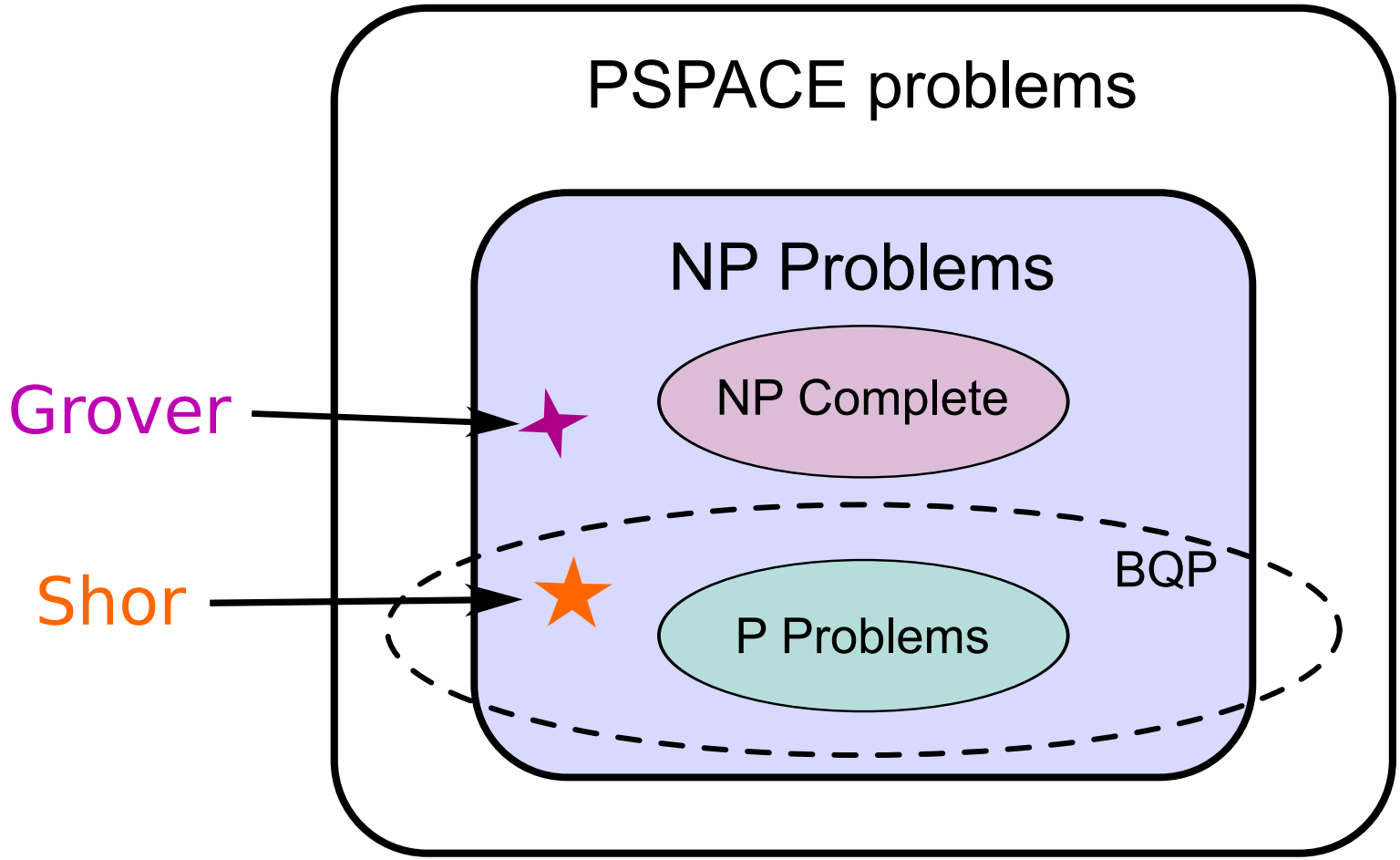
- 1 known output
- N unknown inputs

Complexity estimation for $N \sim 2^{256}$:

Algo	Complexity	Operations
BF	$O(N)$	$\sim 10^{77}$
Grover's	$O\left(\sqrt{N}\right)$	$\sim 10^{38}$

For details see [10, 11].

Complexity classes



P — easy to solve and verify
NP — hard to solve, but easy to verify
BQP — easy to solve on quantum computer and verify

Impact on crypto algos

Symmetric crypto:

- Key sizes are halved

Common asymmetric crypto:

- Elliptic curves are very *dead*
- RSA and alike are *dead*

Quantum resistant asymmetric crypto:

- Hash-based [12]
- Lattice-based [12]
- Code-based [12]
- Multivariative quadratic equations [12]
- Supersingular elliptic curve isogeny [13]
- ...

Impact on crypto algos

Symmetric crypto:

- Key sizes are halved

Common asymmetric crypto:

- Elliptic curves are very *dead*
- RSA and alike are *dead*

Quantum resistant asymmetric crypto:

- Hash-based [12]
- Lattice-based [12]
- Code-based [12]
- Multivariative quadratic equations [12]
- Supersingular elliptic curve isogeny [13]
- ...

Impact on crypto algos

Symmetric crypto:

- Key sizes are halved

Common asymmetric crypto:

- Elliptic curves are very *dead*
- RSA and alike are *dead*

Quantum resistant asymmetric crypto:

- Hash-based [12]
- Lattice-based [12]
- Code-based [12]
- Multivariative quadratic equations [12]
- Supersingular elliptic curve isogeny [13]
- ...

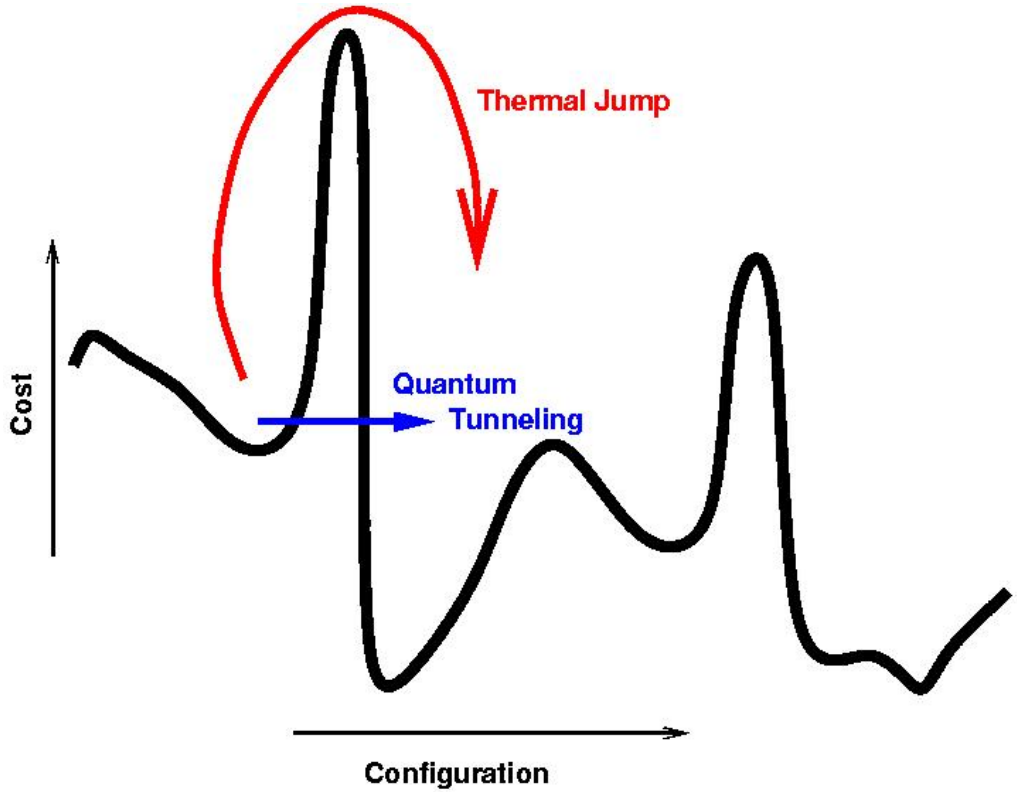
D-Wave Systems

Jan 24, 2017: 2000 qubits

D-Wave Systems

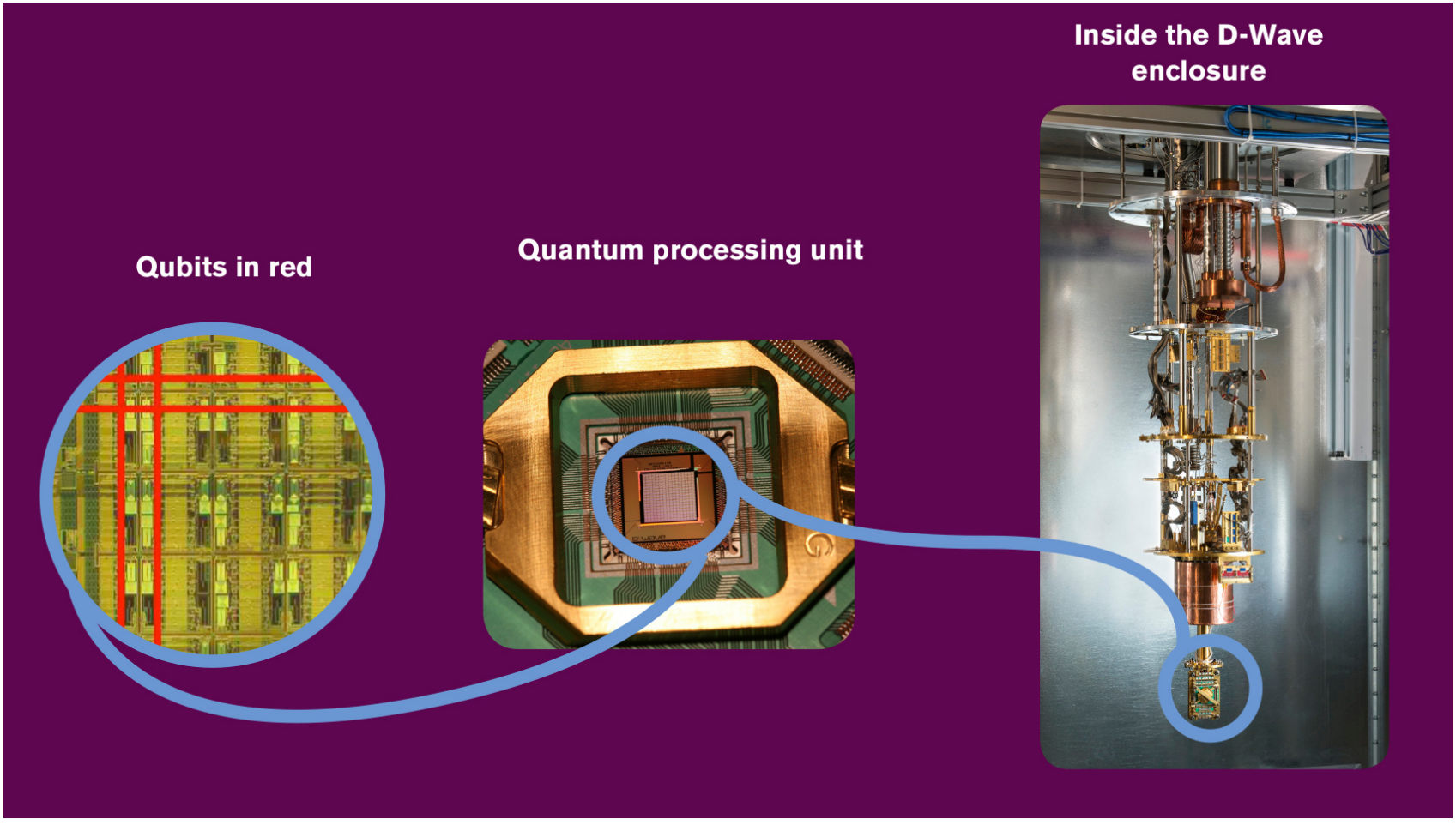
- Operates just at 0.015K :)
- Computing in adiabatic quantum approximation [14].
- Declared to be suitable only for discrete optimisation [14, 15] using quantum annealing.
- QPU Beats 2500 core GPU at factor $1000 \div 10000$ citedwave-2000
- It can simulate…itself. This is useful [16]

Quantum annealing



- search for global minimum
- discrete search space
- uses quantum tunneling

D-Wave QPU


Inside the D-Wave enclosure
Quantum processing unit
Qubits in red

D-Wave's Free software

D-Wave opened *some* of its software [17]:

Qbsolv,a decomposing solver:

- finds minimum for quadratic unconstrained binary system
- written in C
- only classical code?

QCL

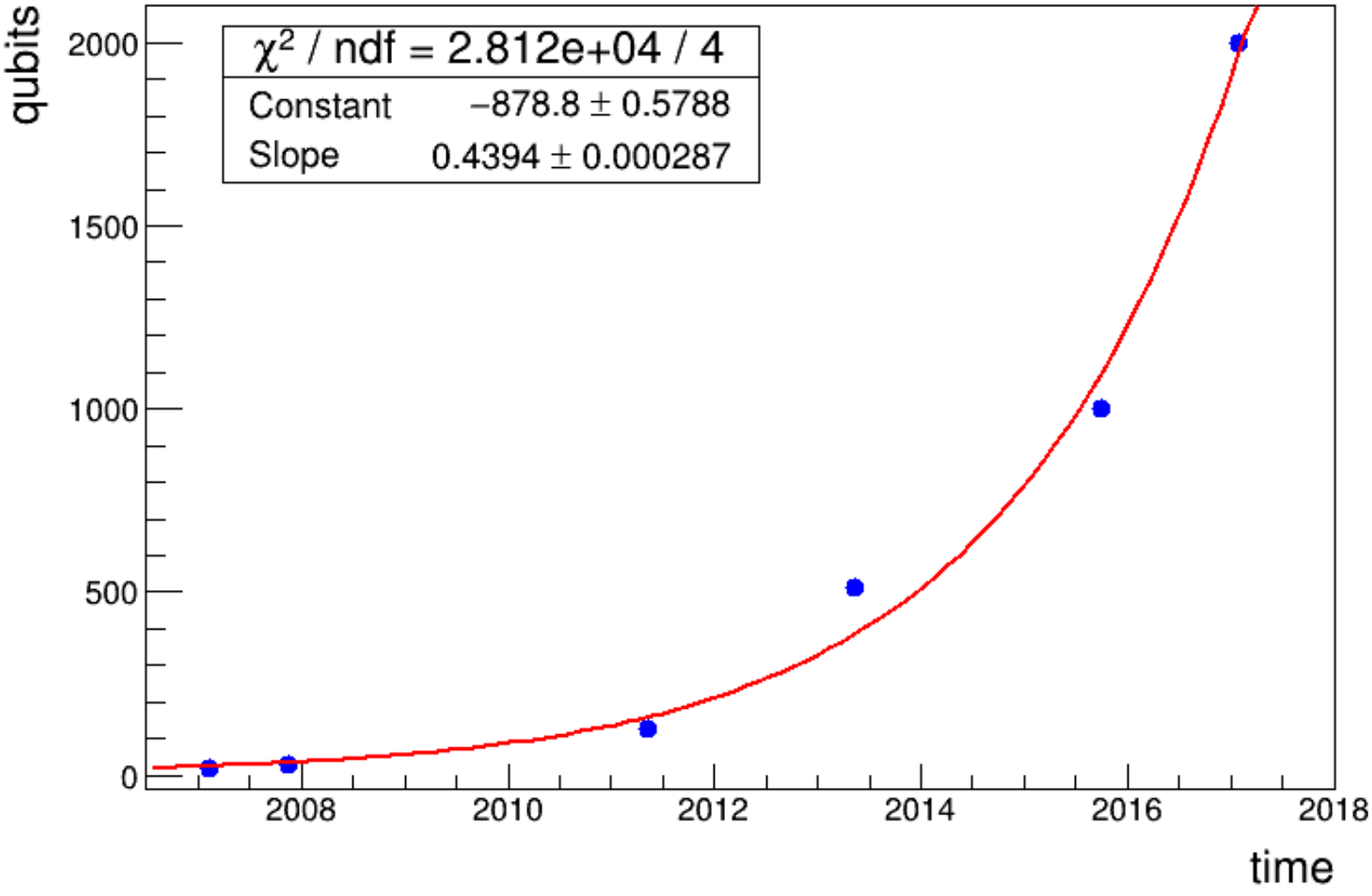
Quantum Computing Language: QCL [18]

- emulator of a quantum computer
- quantum C-like language
- widely used routines

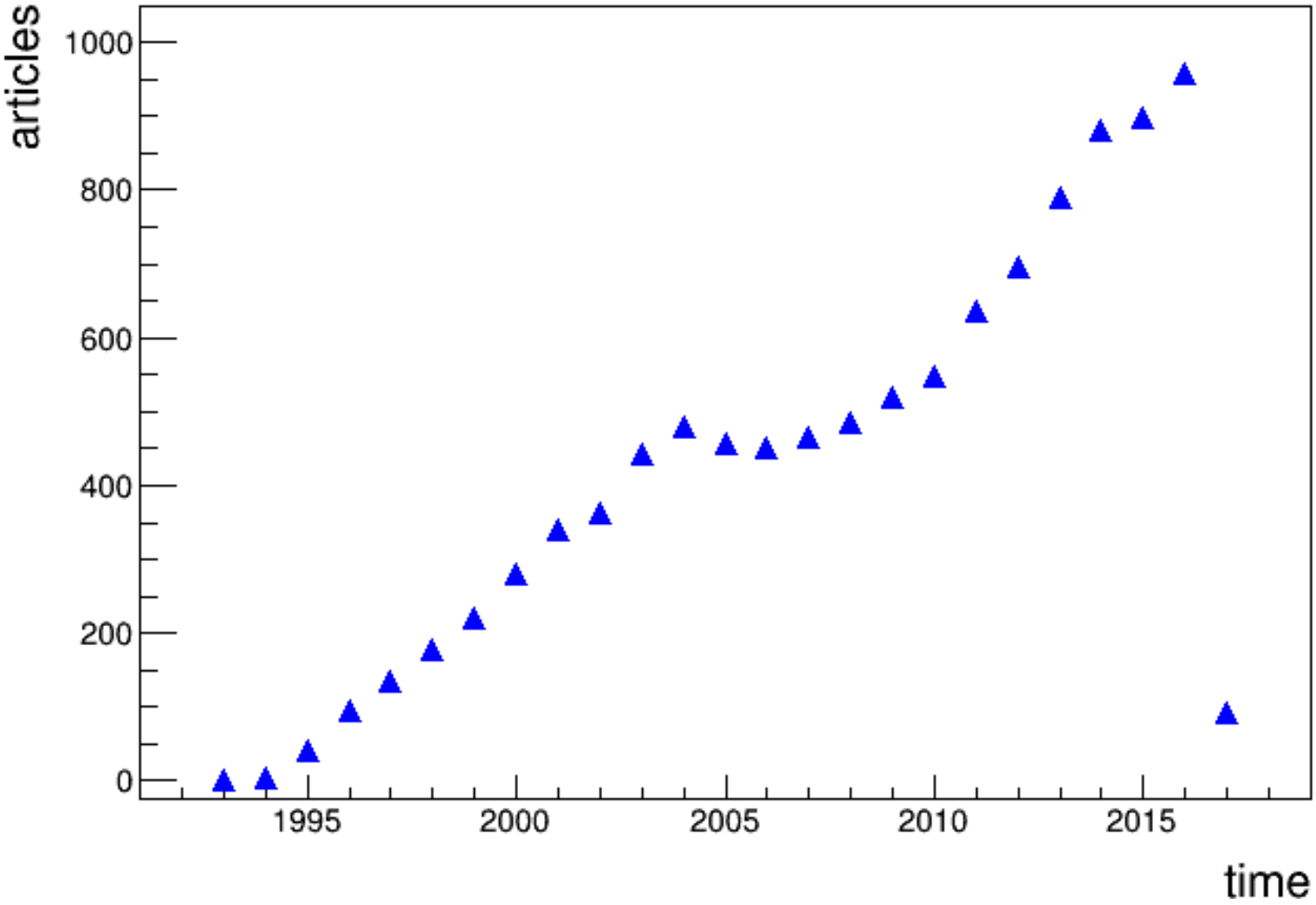
QCL DFFT

```
operator dft(qureg q) { // main operator
  const n=#q;           // set n to length of input
  int i; int j;         // declare loop counters
  for i=1 to n {
    for j=1 to i-1 {    // apply conditional phase gates
      V(pi/2^(i-j),q[n-i] & q[n-j]);
//      if q[n-i] and q[n-j] { Phase(pi/2^(i-j)); }
    }
    H(q[n-i]);          // qubit rotation
  }
  flip(q);              // swap bit order of the output
}
```

D-Wave Systems Timeline



Publications timeline



Based on arXiv.org queries for QC publications

Why should you care?

- Nobody will announce large Shor-capable QC
  - Mr. Snowden revealed that NSA spent $80m on a quantum computer development [19, 20]
- Development is fast:
  - exp grow of qubits
  - IBM estimate: 2022-2027
- Your data is **not** forward secure
- Cryptography takes decades to establish!

Security is complex

- well tested algorithms
- good protocols
- robust, auditable software
- secure environment
- reasonable users

Free Software Solutions

Many of them, search github.

Most interesting:

Crypto	PQ-Crypto
GnuPG	codecrypt [21]
OpenSSL	openssl/liboqs [22]
	sarkara [23]

Codecrypt

Codecrypt [21] — GnuPG-like encryption tool

- Signatures: hash-tree based (Merkle-tree signature)
- Asymmetric encryption: code based McEliece cryptosystem [24])
- Symmetric encryption: up to 4096-bit keys

- In-memory asymmetric encryption
- Keys on disk are not encrypted
- No key-server infrastructure

Codecrypt

Codecrypt [21] — GnuPG-like encryption tool

- Signatures: hash-tree based (Merkle-tree signature)
- Asymmetric encryption: code based McEliece cryptosystem [24])
- Symmetric encryption: up to 4096-bit keys

- In-memory asymmetric encryption
- Keys on disk are not encrypted
- No key-server infrastructure

Codecrypt

Keys generation:

```
ccr -g sig -N ”John” # signature key
ccr -g enc -N ”John” # encryption key

# the same with manual key choise
ccr -g FMTSEQ256H20C-CUBE512-CUBE256  -N ”John”
ccr -g MCEQCMDPC256FO-SHA512-CHACHA20 -N ”John”
```

Sign and encrypt:

```
ccr -se -r Frank < letter.txt > letter.ccr
```

Decrypt and verify:

```
ccr -dv -o reply.txt < reply.ccr
```

Upstream is very dynamic and responsive.

Codecrypt

Keys generation:

```
ccr -g sig -N ”John” # signature key
ccr -g enc -N ”John” # encryption key

# the same with manual key choise
ccr -g FMTSEQ256H20C-CUBE512-CUBE256  -N ”John”
ccr -g MCEQCMDPC256FO-SHA512-CHACHA20 -N ”John”
```

Sign and encrypt:

```
ccr -se -r Frank < letter.txt > letter.ccr
```

Decrypt and verify:

```
ccr -dv -o reply.txt < reply.ccr
```

Upstream is very dynamic and responsive.

Open Quantum Safe

liboqs [25] provides PQ key exchange:

- Ring learning with errors (New Reno, etc)
- NTRN
- Supersingular Isogeny Diffie-Hellman
- Error-correcting codes (McBits)

OpenSSL-1.0.2 fork with liboqs support.

Summary

- Symmetric cryptography is still secure, but *double* key size
- Drop RSA, DSA, ECC in the long run, minimize usage
- Use codecrypt and other systems, but with caution
- Combine multiple crypto systems
- Do not blindly trust standards with questionable constants

Summary

All depends on you:

- **use** it
- contribute and develop
- audit code
- peek into math [26, 27]

Thank you for your attention!

Bibliography I


Morton J.J.L., et al. Solid-state quantum memory using the ${}^{3}1P$ nuclear spin. —
2008. —
URL: https://arxiv.org/abs/0803.2021.


University of Sussex. —
Microchip Architectures for Scalable Ion Trap Quantum Computing. —
URL: http://www.sussex.ac.uk/physics/iqt/research/undergrad/microchip.html.


Baldauf H., et al. The Physics of Quantum information / Ed. by Dir Bouwmeester, Artur Ekert, Anton Zeilinger. —
Berlin : Springer, 2000.


Ekert A., Jozza R. // Phil. Trans. Roy. Soc. London. —
1998. —
P. 1769.


Shor's algorithm. —
URL: https://en.wikipedia.org/wiki/Shor's_algorithm.


Shor Peter W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. —
1996. —
URL: https://arxiv.org/abs/quant-ph/9508027.

Bibliography II


Ekert A., Jozza R. // Rev. Mod. Phy. — 1996. — Vol. 98. — P. 733.


Dattani Nikesh S., Bryans Nathaniel. Quantum factorization of 56153 with only 4 qubits. — 2014. — URL: http://arxiv.org/abs/1411.6758.


Proos John, Zalka Christof. Shor's discrete logarithm quantum algorithm for elliptic curves // QIC 3. — 2003. — Vol. 4. — P. 317. — URL: https://arxiv.org/abs/quant-ph/0301141.


Grover's algorithm. — URL: https://en.wikipedia.org/wiki/Grover's_algorithm.


Grover L. // Phys. Rev. Lett. — 1997. — Vol. 78. — P. 325. — URL: https://arxiv.org/abs/quant-ph/9508027.

Bibliography III


Bernstein Daniel J., Buchmann Johannes, Dahmen Erik. Post-quantum cryptography. —
Berlin : Springer, 2009. —
ISBN: 978-3-540-88701-0. —
URL: https://pqcrypto.org/www.springer.com/cda/content/document/cda_downloaddocument/9783540887010-c1.pdf.


Supersingular isogeny Diffie–Hellman key exchange. —
URL: https://en.wikipedia.org/wiki/Supersingular_isogeny_key_exchange.


D-Wave Systems. —
URL: http://www.dwavesys.com/.


D-Wave Systems. —
URL: https://en.wikipedia.org/wiki/D-Wave_Systems.


University of Sussex. —
Quantum Simulation. —
URL: http://www.sussex.ac.uk/physics/iqt/research/undergrad/simulation.html.


Qbsolv, a decomposing solver. —
URL: https://github.com/dwavesystems/qbsolv.

Bibliography IV


QCL (quantum computing language and quantum computer emulator). —
URL: http://tph.tuwien.ac.at/~oemer/qcl.html.


Snowden docs: NSA building encryption-cracking quantum computer. —
URL: http://www.theregister.co.uk/2014/01/03/snowden_docs_show_nsa_building_encryptioncracking_quantum_system/.


NSA seeks to build quantum computer that could crack most types of encryption. —
URL: https://www.washingtonpost.com/world/national-security/nsa-seeks-to-build-quantum-computer-that-could-crack-most-types-c
2014/01/02/8fff297e-7195-11e3-8def-a33011492df2_story.html.


Codecrypt (post-quantum crypto suite). —
URL: http://e-x-a.org/codecrypt/.


Open Quantum Safe. —
URL: https://openquantumsafe.org/.


Sarkara is a Post-Quantum cryptography library. —
URL: https://github.com/quininer/sarkara.

Bibliography V


McEliece cryptosystem. —
URL: https://en.wikipedia.org/wiki/McEliece_cryptosystem.


C library for quantum-resistant cryptographic algorithms. —
URL: https://github.com/open-quantum-safe/liboqs.


Post-quantum cryptography. —
URL: https://pqcrypto.org.


Post-quantum cryptography. —
URL: http://pqcrypto.eu.org/.