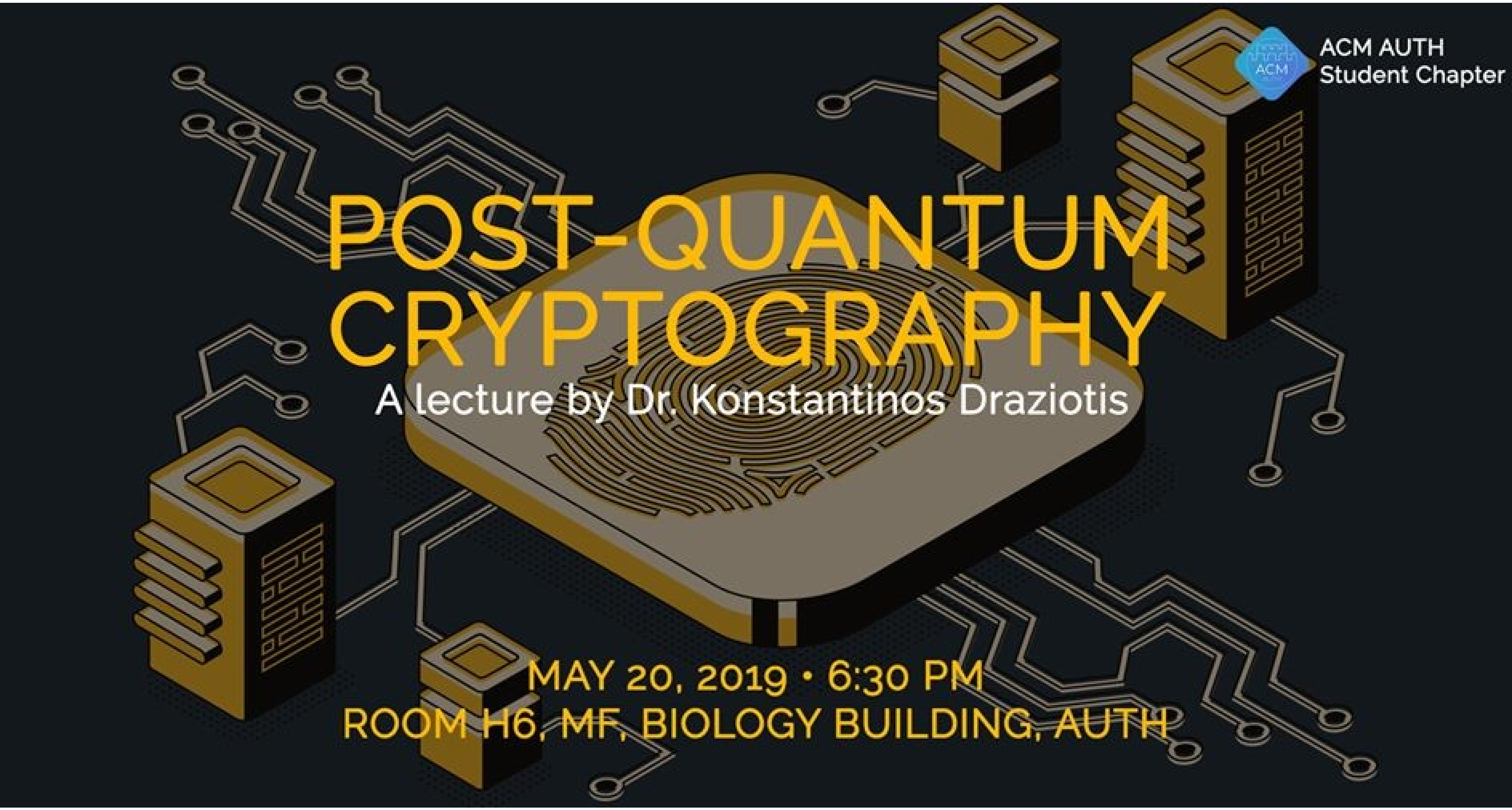

ACM AUTH
Student Chapter
POST-QUANTUM
CRYPTOGRAPHY
A lecture by Dr. Konstantinos Draziotis
MAY 20, 2019 • 6:30 PM
ROOM H6, MF, BIOLOGY BUILDING, AUTH

Post Quantum Cryptography

By K.Draziotis

Informatic's Department

Aristotle University of Thessaloniki

20 May 2019

Invitation by ACM student portal of AUTH

Quantum Cryptography and Post Quantum Cryptography

- Two completely different things.
- [*wikipedia definition*] **Quantum Cryptography** is the science of exploiting quantum mechanical properties to perform cryptographic tasks. For instance BB84 is a key exchange protocol, that exploits quantum mechanic properties.
- With the term **Post Quantum Cryptography** we mean all the cryptographic algorithms that are (probable) safe against an attack using a (powerful) quantum computer.

What is PQC?

- When we are talking about PQC we (usually) mean cryptography which is **not** based on **Factorization** or **Discrete Logarithm Problem (DLP).**
- We shall see later these problems.

History of Quantum Computers

- Quantum computers were first introduced by Paul Benioff (1980)
- Feynman (1982) considered the inverse problem, how a Turing machine can simulate a quantum systems.
- Also, the mathematician Y. Manin (1982) suggested the idea that a quantum computer can simulate things that a classic computer could not.
- Deutsh (1985) was first propose that quantum superposition might allow to perform classical computations in parallel.
- Loyd in 1993, proposed a quantum computer based on electromagnetic pulses.
- In 1994, Peter Shor presented his quantum **polynomial** algorithm for factorization and discrete logarithm and in 1995 proposed the first quantum error correction code.

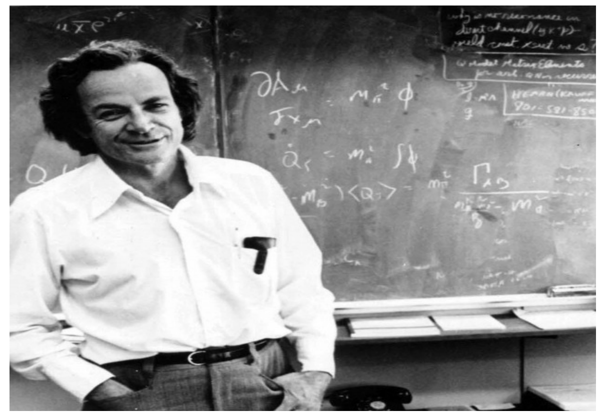
History of Quantum Computers

In 1996, Grover suggested a quantum algorithm for pattern recognition or data mining. For N elements in a database, after about sqrt(N) trials, his algorithm finds a given element. The corresponding classical algorithm needs about N/2 trials.

What is quantum mechanics?

I think I can safely say that nobody understands quantum mechanics

- Richard Feynman



or

If you are not completely confused by quantum mechanics, you do not understand it

- John Wheeler

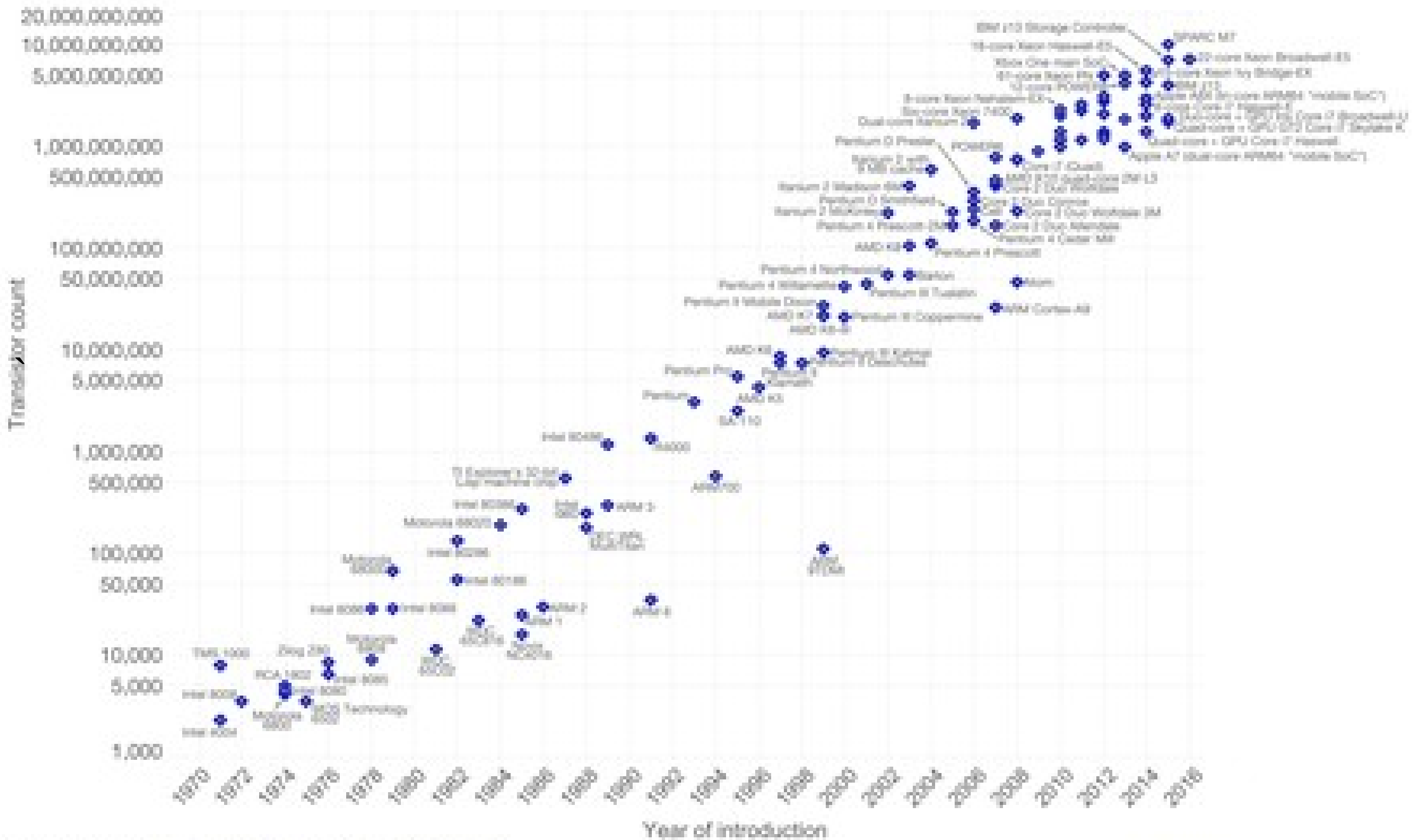
Moore’s law

Gordon Moore, the co-founder of Intel, observed that **the number of transistors in a integrated circuit doubles about every two years or so.**

- Many people start to believe that maybe this law is coming to an end.
- For instance, Quad-core + GPU Core i7 Haswell = **1,400,000,000**

Moore's Law – The number of transistors on integrated circuit chips (1971-2016)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Advantage of QC

- The main advantage of quantum computers is that can execute in parallel logic operations by using superposition of quantum states.
- Such a computer “runs” quantum algorithms. As we saw there are such algorithms, for instance Grover's and Shor's Algorithm.

qubits and bits

- In information theory the basic unit of information is bit.
- Similarly, in Quantum information theory, there is the qubit. In simple words a qubit is a quantum object that occupy some quantum states. We use two of these states to store quantum information.
- Quantum physics allow us to operate with a superposition of these two states.
- Utilization of quantum states allow us to work with quantum states that represent simultaneously different numbers. This is called **quantum parallelism.**
- Not all problems are suitable for quantum computers. We need a suitable quantum algorithm that exploits this quantum parallelism.

State of the art for Quantum Computers

- https://en.wikipedia.org/wiki/Timeline_of_quantum_computing
- D-Wave

But..., D-Wave isn't universal.

For instance, can not run Shor's Algorithm.

IBM-Q system one

- A universal approximate superconducting quantum computer.
- Is a 20-qubit quantum computer.
- Researchers managed to implement Shor's algorithm in a 5-qubit processor of IBM (ibm Q experience) using 11 gates (https://arxiv.org/pdf/1804.03719.pdf)

Well, do I have to save some bucks for the new extended quantum mobile?

No. The current status of quantum computers is embryonic.

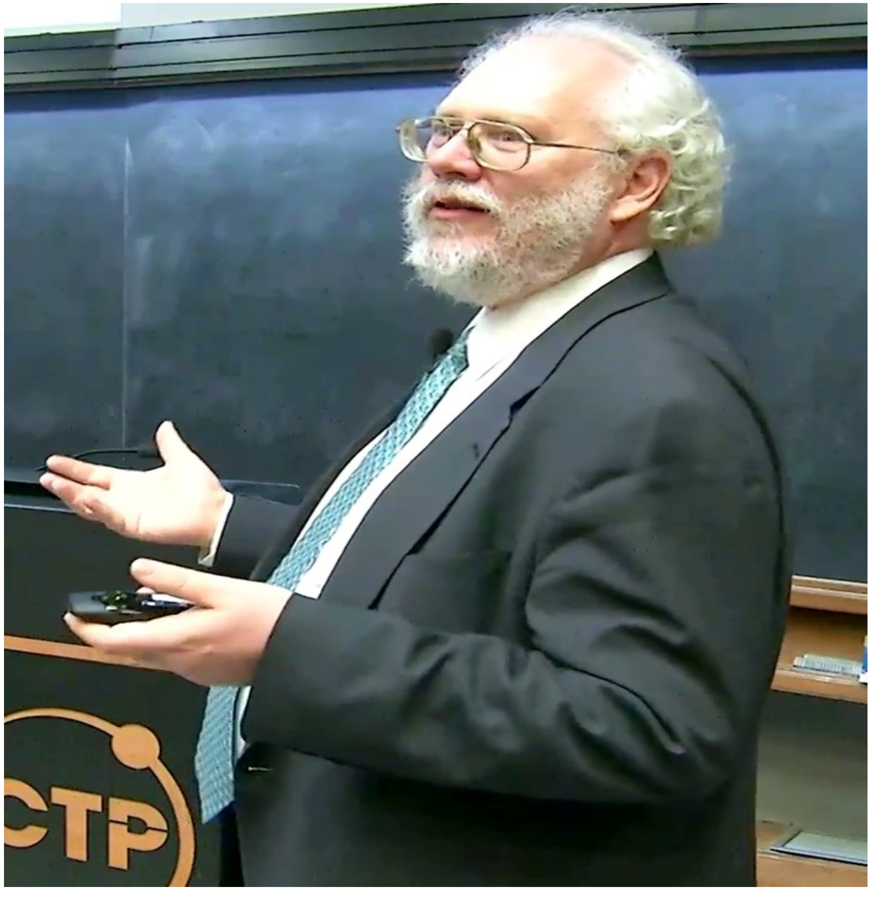
Although, Google and IBM starts to heavily invest in this technology.

- In 2017, venture investors spent $241 million into quantum computers.
- European Union pledged to give $1 billion for advancing this technology.

Why these (future) devices are useful?

- Well, to be honest such devices will be useful to any government, since they will comprise the current security in the Internet.
- Also, these devices does not always are more efficient than our usual computers.
- Satya Nadella, chief executive officer of Microsoft, calls *quantum computing* one of three emerging technologies that will radically reshape the world, along with *artificial intelligence* and *augmented reality*.
- Solve optimization problems e.g. discover new drugs

Peter Shor



- Peter Shor discovered a polynomial time (probabilistic) algorithm that solves DLP in a generic group G (1994). The same algorithm can be used for factoring integers.

QC and cryptography

- Peter Shor's algorithm is the most well known quantum algorithm.
- Although there are enough quantum algorithms,

  Deutsch - Jozsa Algorith (1992) (implies EQP$\neq$P)

  Simon’s Algorithm (1994) (implies BQP$\neq$BPP)

  Grover’s Algorithm (1996)

  [see : http://quantumalgorithmzoo.org/]
- If a quantum computer with large memory ever constructed, then the most well known public key cryptosystem, RSA, shall break and all the current security in Internet will be comprised.

Shor's algorithm

Shor's algorithm for an integer N with d digits, needs:

- Memory : 10d qubits
- Quantum gates : $O(\log_2 N(\log_2 \log_2 N)(\log_2 \log_2 \log_2 N))$

  more simple, it has running time d^3

  and so proving that FACTOR problem is in **BQP** complexity class : **B**ounded-error **Q**uantum **P**olynomial time.
- Today (2019) IBM-Q the quantum computer of IBM has 20 qubits memory.
- To factor a 2048-bit RSA modulus we need about 6100 qubits.

Hidden Subgroup Probem (HSP)

Shor’s algorithm reduces the problem of factorization to a specific class of a general problem called HSP. To define HSP we need a group G.

Hidden Subgroup Probem (HSP)

HSP : Given a description of a finite group **G** and a function f on **G** that is promised to be strictly **H**-periodic for some subgroup **H** find a generating set for **H.**

- [Simon, Shor, Kitaev] If G is Abelian then we can solve HSP in polynomial time with a bounded error in a quantum computer. That is, we can efficiently find a subset X of G that generates the subgroup H with probability 2/3.
- [Ettinger, Hoyer,Knill, 2004] They improve the probability to be exponentially small
- For non Abelian groups our knowledge is limited
- Regev (2008) provide a subexponential quantum algorithm for the **Dihedral group** D(N) (is non-Abelian for N>2). Also in 2004 showed that an efficient solution to HSP implies a solution to the SVP (a hard lattice based problem) .

Problem	Group	Time in Quantum Turing machine	Time in Turing Machine	cryptosystem
Factor	$\mathbf{Z}$	polynomial	subexponential	RSA
DLP	$\mathbf{Z}_{p-1}^2$	polynomial	subexponential	DSA, ElGammal, Diffie-Hellman
ECDLP	Elliptic curve	polynomial	exponential	ECDSA, ECDH
Principal ideal	$\mathbf{R}$	polynomial	exponential	Buchman-Williams
SVP	D_n	subexponential	exponential	NTRU, LWE
Graph Isomorphism Problem	S_n	exponential	exponential (*)	

* : Babai announced a subexponential time algorithm, but not yet full peer reviewed

Is it easy to factor integers with classic computers?

- **Short answer** : No.
- **Long answer** : The best algorithm to factor an integer N is called GNFS and has (heuristic) complexity

$$\exp\left(\left(\sqrt[3]{\frac{64}{9}}+o(1)\right)(\ln N)^{1/3}(\ln\ln N)^{2/3}\right)$$

What is the factorization problem?

With factorization in crypto we mean factorization of an integer in prime numbers.

- For instance, 15 = 3 x 5

  123 = 3 x 41

  1234 = 2 x 617

  12345 = 3 x 5 x 823

- What about

**1522605027922533360535618378132637429718068114961380688657908494580
122963258952897 ?**

270 bits

What is the factorization problem?

Well, in fact we can factor the previous integer.

- $msieve **1522605027922533360535618378132637429718068114961380688657908494580 122963258952897** -t 2 -q

After a few seconds....

p4: 1289

p6: 106319

p6: 261791

p8: 31463891

prp25: 3873557024207554787271121

prp36: 348214109347518254854468233839101667

GNU Implementations of GNFS

There are many implementations of GNFS, I used msieve.

- **GGNFS** by Chris Monico
- **msieve** by Jason Papadopoulos
- **YAFU** by Ben Buhrow
- **CADO-NFS** developed by Emmanuel Thomé, Lionel Muller, Alexander Kruppa, Pierrick Gaudry, Franĉois Morain, Jérémie Detrey and Paul Zimmerman

RSA-129

- RSA challenges :

$$
\begin{aligned}
RSA-129 = {} & 114381625757888867669235779976146 \\
& 612010218296721242362562561842935 \\
& 706935245733897830597123563958705 \\
& 058989075147599290026879543541
\end{aligned}
$$

129 decimal digits or 426 bits

RSA-129, was factored in 1994 by Arjen Lenstra et al.

RSA challenges

The largest RSA integer factored is

RSA-768, and was factored on December 12, 2009 over the span of two years.

- For larger integers, at least 2048 bits, the problem is too hard with the current state of the art algorithms.

Trying our example

Using openssl we can generate a RSA modulus, i.e. integers of the form p*q, with specific binary length.

For instance

- $openssl genrsa -out example.key 224

  will generate a 224 bit modulus.

  For the script that generates such RSA modulus

  see https://github.com/drazioti/msieve/blob/master/example_openssl
- For the talk see here

  https://github.com/drazioti/talks

msieve

We can use *msieve,* and try to compute their prime factors.

- $ time ./msieve -i rsa.out -t 4 -q

  224366344137519887757124359913900451863123509954290460
382737549 17719

  prp34: 4522437426744219274007933240434787

  prp34: 4961181835500708955939626680723837

  real 0m19.642s

  user 0m19.620s

  sys   0m0.012s

How the key length is changing.

See https://www.keylength.com/

Protection	Symmetric	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash
Legacy standard level *Should not be used in new systems*	80	1024	160	1024	160	160
Near term protection *Security for at least ten years (2019-2028)*	128	3072	256	3072	256	256
Long-term protection *Security for thirty to fifty years (2019-2068)*	256	15360	512	15360	512	512

So factorization is difficult. Now what?

Rivest **S**hamir **A**dleman

They managed to use this problem to cryptography. In fact they provided the first example of a trapdoor function (TDF). Using a TDF, two entities can exchange a secret key over an insecure channel.

- A TDF is a one way function and can be efficient inverted if a secret information (trapdoor) is known.
- TDF definition was first given by **Diffie** and **Hellman**.

RSA assumption

- RSA-TDF can be inverted (without knowing the secret information) if factorization of integers is an easy problem.
- Having a quantum computer we can break RSA.

RSA is everywhere

- All the major servers uses RSA, for **exchange keys** over the Internet

  If they do not use RSA, they use Diffie-Hellman which is based in another problem, Discrete Logarithm Problem, which again was broken by Shor's algorithm.

- Also, RSA is used to construct **digital signatures**. So, this signature will also considered broken in the case of a construction of a (large) quantum computer.

What about symmetric crypto?

- Symmetric cryptography (e.g. AES) is not vulnerable to quantum attacks.
- Also, the same for cryptographic hash functions.
- So only the public key cryptography is not safe against quantum attacks.

DLP:Discrete Logarithm Problem

- Let the function $E : G \to G, E(x) = g^x, \text{where } G = \langle g \rangle$

  $\text{Given } g^x = y \ \text{ find } x$
- In key exchange protocols instead of RSA we can use Diffie - Hellman which uses DLP.
- Again, Peter Shor’s algorithm can solve DLP in polynomial time.

Panic or not to panic?

Food for thought

But why we care?

- when you got nothing, you got nothing to lose
Bob Dylan

Although, there is a solution.

Elementary Dr. Watson

- The following problems are suitable for post quantum crypto!

  - Code based crypto 1978, McEliece
  - Hash based crypto 1979, Lamport and Diffie and Merkle
  - Multivariate Quadratic (MQ) system 1996
  - Lattice based crypto 1998 : e.g. NTRU, LWE
  - Supersingular Isogeny Problem 2006, (SIDH) Key exchange

Which problem/cryptosystem is the most suitable?

- Would you tell me, please, which way I ought to go from here?
Alice in Wonderland

NIST tries to solve the previous problem

Call of NIST for PQC

- NIST in 2016, made a call for proposals to evaluate and standardize post quantum cryptosystems.
- Today we are in the second round where **17** algorithms for key exchange passed and **9** for digital signatures.

Quantum safe algorithms

- We have to clarify what do we mean safe.
- With safe we mean that, **as far as we know**, there is not any quantum polynomial algorithm for these problems.

Why are we not using the previous cryptosystems instead of RSA and Diffie-Hellman?

- One reason is that the previous cryptosystems use larger keys so they are not so efficient.
- So in the pre-quantum computer era we prefer RSA/Diffie-Hellman, since they provide better performance.
- The previous is almost true, since NTRU cryptosystem (lattice based) is 60% faster than RSA in encrypting phase, and 90% faster in decrypting phase. Although RSA has smaller public keys than NTRU (for 80-bit security).

Also, we need time to produce reliable cryptosystems

- We need time for cryptanalysis of the system
- We need time to produce efficient and secure implementations for real world applications
- Study exotic attacks (fault, side channel attacks)
- Implement the cryptosystem to hardware
- Produce standards

CECPQ1- google’s experiment

Google in 2016 implemented a post quantum scheme TLS, and run this project for five months.

- They used, Newhope scheme, a lattice based key exchange system combined with an elliptic curve key exchange system So if for some reason the post quantum compromised the other system will provide the necessary security,
- Also, TLS 1.3 designed in a way to implement in the future hybrid algorithms (this is an intermediate step before we pass to “full” post quantum schemes).
- See : https://tools.ietf.org/id/draft-stebila-tls-hybrid-design-00.html

Code based cryptography

The first code-based public-key cryptosystem was introduced in 1978 by McEliece.

- The security of this cryptosystem is based on the General decoding problem of linear codes :
- Let C be an [n,k]-linear code over F and $y \in F[n]$. Find a codeword $c \in C$ such that the distance $d(y,c)$ is minimal.

Hash based signatures

- This primitive can only provide signatures schemes. The first example is the Merkle signature which improved the One Time Signature (OTS) of Lamport and Diffie.
- These schemes are based on cryptographic security hash functions. Since, hash functions are well understood, we believe that it is a good candidate for standardization.
- NIST chose the hash based digital signature SPHINCS+ to continue to the second round.
- Today to sign messages we use RSA, DSA, ECDSA, which are not quantum resistant.

MQ based crytptography

- NIST passed the digital signature MQDSS which is based on MQ problem, to the second round.
- The problem of finding a solution to a quadratic system over a finite field is called MQ problem.

$$\begin{cases} x_0x_1 + x_1x_2 + x_2 & = & 0 \\ x_1x_2 + x_0 & = & 1 \\ x_0x_2 + x_1x_2 & = & 1 \\ x_0x_1 + x_0 + x_2 & = & 1 \\ x_0x_2 + x_1x_2 + x_1 & = & 0 \end{cases}$$

(0,1,1) is a solution over GF[2].

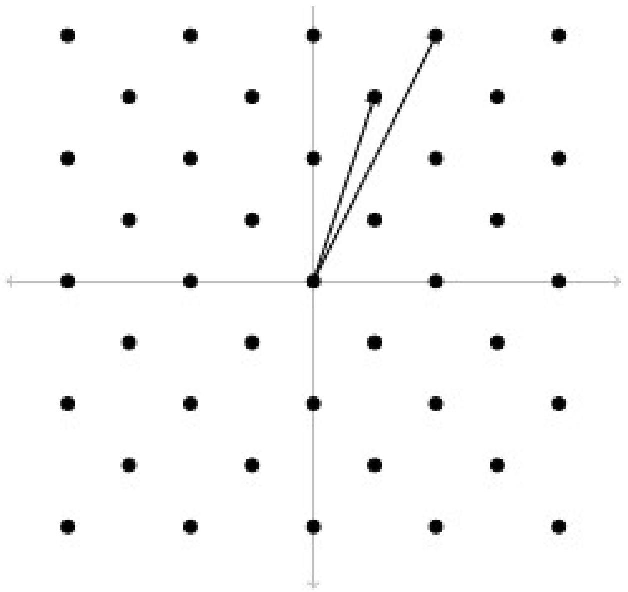
- There are many cryptosystems/digital signatures based on this problem.
- For intance

  Matsumoto-Imai,

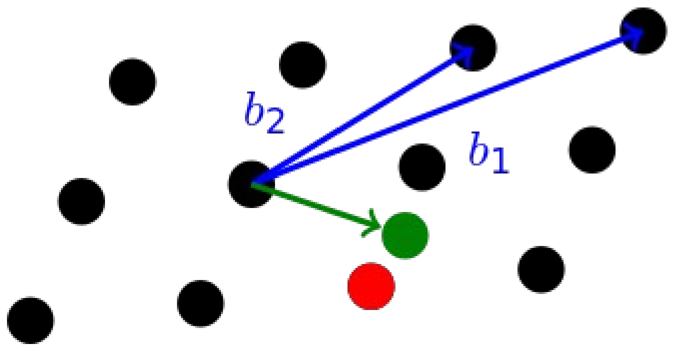
  Hidden Field Equation cryptosystem (HFE).
- This problem is important for two other reasons. We can attack AES cryptosystem and ECDLP (over prime finite fields).

Lattice based cryptography

A two dimensional lattice



- The problem of finding a shortest vector (which always exists) is called Shortest Vector Problem (SVP).
- If **t** is a vector not in lattice, then the problem of finding a lattice vector that is closest to **t** is called Closest Vector Problem (CVP)



Lattice based cryptography

- The first major result for Lattice based crypto was presented by Ajtai iSn 1996.
- Ajtai described a problem that is *hard on average* if some lattice problems are hard on the *worst case.*
- Ajtai constructed a hash function and with Dwork constructed an encryption scheme that based on SVP.

Lattice based cryptography

- Goldreich – Goldwasser - Halevi, provides a trapdoor function that relies on the Closest Vector Problem (1997)
- Their idea was to choose two bases, one “good” and one “bad”. The good one could solve efficiently a CVP. Show the encrypted the messages by using the bad one and decrypting by using the good one.
- The message m, was a point of the lattice and the encryption was the addition of a random error vector e to m. Set t=m+e
- To decrypt you had to solve a CVP with target vector **t.** If someone knows the *good basis* then he can solve the CVP, and so decrypt the ciphertext.
- Similar ideas were used to MacEllice cryptosystem

Lattice based cryptography

- In 2016, Eldar and Shor submitted the following paper in arxiv.org

  An Efficient Quantum Algorithm for a Variant of the Closest Lattice-Vector Problem

  For almost one day, cryptographers believed that this was the end of lattice based crypto.

  Although Regev found a mistake and the paper withdrawn.

- The most well known lattice based cryptosystem is the NTRU.
- It was first proposed by Hoffstein,Pipher, Silverman (1996)
- It was standardized by

  IEEE Std 1363.1-2008 and ANSI X9.98-2010.
- StrongSwan is a OpenSource IPsec-based VPN Solution that implements NTRU.

Except NTRUencrypt there is a digital signature based on NTRU.

- The cryptanalysis of NTRU based on lattices. We have to solve a SVP or BDD problem in order to attack NTRU

Elliptic curves (SIDH)

- NIST chose for the second round *SIKE* cryptosystem which uses the SIDE.
- The proposed protocol uses 2688-bit public keys for 128-bit security.
- Further, provides forward secrecy, i.e. protects past sessions against future compromises of secret keys or passwords.
- The security of SIDH is closely related to the problem of finding the isogeny mapping between two supersingular elliptic curves with the same number of points
- Andrew Childs, David Jao, and Vladimir Soukharev, provided a subexponential quantum of attack for isogeny problem for ordinary elliptic curves (2010).

Thank you!