Quantum Cryptanalysis: Let's build a quantum computer

Gustavo Banegas[1]

February 9, 2021

[1] INRIA & LIX - École polytechnique, France
gustavo@cryptme.in

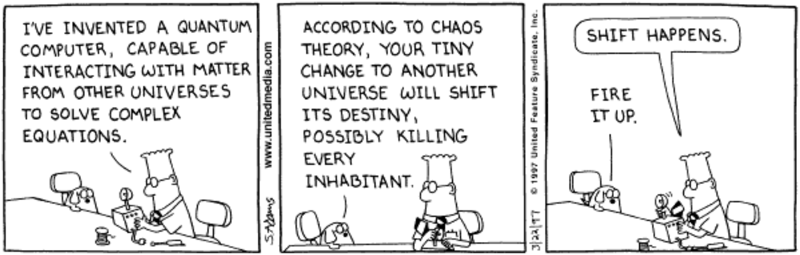
Outline

Table of Contents

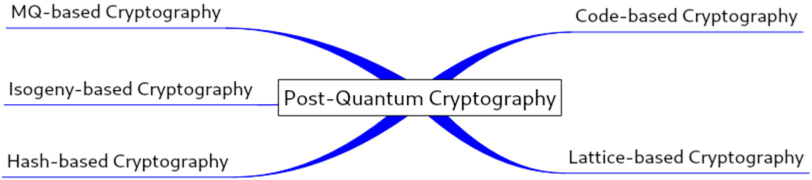
Why study post-quantum cryptography?

"Somebody announces that he's built a large quantum computer. RSA is dead. DSA is dead. Elliptic curves, hyperelliptic curves, class groups, whatever, dead, dead, dead."(Bernstein, 2005)

In other words..



There is already an alternative



A little bit of history

- **1994** - Peter Shor presents "Algorithms for quantum computation: discrete logarithms and factoring";

A little bit of history

- **1994** - Peter Shor presents "Algorithms for quantum computation: discrete logarithms and factoring";
- **1996** - Grover presents "Quantum search algorithm for unoreded database";

A little bit of history

- **1994** - Peter Shor presents "Algorithms for quantum computation: discrete logarithms and factoring";
- **1996** - Grover presents "Quantum search algorithm for unoreded database";
- **2003** - Daniel J. Bernstein introduces the concept of Post-quantum cryptography;

A little bit of history

- **1994** - Peter Shor presents "Algorithms for quantum computation: discrete logarithms and factoring";
- **1996** - Grover presents "Quantum search algorithm for unoreded database";
- **2003** - Daniel J. Bernstein introduces the concept of Post-quantum cryptography;
- **PQCrypto 2006** - 1st International Workshop on Post-Quantum Cryptography;

A little bit of history

- **1994** - Peter Shor presents "Algorithms for quantum computation: discrete logarithms and factoring";
- **1996** - Grover presents "Quantum search algorithm for unoreded database";
- **2003** - Daniel J. Bernstein introduces the concept of Post-quantum cryptography;
- **PQCrypto 2006** - 1st International Workshop on Post-Quantum Cryptography;
- **PQCrypto 2008** - 2nd International Workshop on Post-Quantum Cryptography... and goes on;

A little bit of history

- **1994** - Peter Shor presents "Algorithms for quantum computation: discrete logarithms and factoring";
- **1996** - Grover presents "Quantum search algorithm for unoreded database";
- **2003** - Daniel J. Bernstein introduces the concept of Post-quantum cryptography;
- **PQCrypto 2006** - 1st International Workshop on Post-Quantum Cryptography;
- **PQCrypto 2008** - 2nd International Workshop on Post-Quantum Cryptography... and goes on;
- **2014** - EU publishes H2020 call including post-quantum crypto as topic;

A little bit of history

- **1994** - Peter Shor presents "Algorithms for quantum computation: discrete logarithms and factoring";
- **1996** - Grover presents "Quantum search algorithm for unoreded database";
- **2003** - Daniel J. Bernstein introduces the concept of Post-quantum cryptography;
- **PQCrypto 2006** - 1st International Workshop on Post-Quantum Cryptography;
- **PQCrypto 2008** - 2nd International Workshop on Post-Quantum Cryptography... and goes on;
- **2014** - EU publishes H2020 call including post-quantum crypto as topic;
- **2015** - NSA admits that the world needs post-quantum crypto;

A little bit of history

- **1994** - Peter Shor presents "Algorithms for quantum computation: discrete logarithms and factoring";
- **1996** - Grover presents "Quantum search algorithm for unoreded database";
- **2003** - Daniel J. Bernstein introduces the concept of Post-quantum cryptography;
- **PQCrypto 2006** - 1st International Workshop on Post-Quantum Cryptography;
- **PQCrypto 2008** - 2nd International Workshop on Post-Quantum Cryptography... and goes on;
- **2014** - EU publishes H2020 call including post-quantum crypto as topic;
- **2015** - NSA admits that the world needs post-quantum crypto;
- **2016** - NIST calls for submissions to "Post-Quantum Cryptography Standardization Project".
- **2019** - NIST receives 69 proper submissions;

A little bit of history

- **1994** - Peter Shor presents "Algorithms for quantum computation: discrete logarithms and factoring";
- **1996** - Grover presents "Quantum search algorithm for unoreded database";
- **2003** - Daniel J. Bernstein introduces the concept of Post-quantum cryptography;
- **PQCrypto 2006** - 1st International Workshop on Post-Quantum Cryptography;
- **PQCrypto 2008** - 2nd International Workshop on Post-Quantum Cryptography... and goes on;
- **2014** - EU publishes H2020 call including post-quantum crypto as topic;
- **2015** - NSA admits that the world needs post-quantum crypto;
- **2016** - NIST calls for submissions to "Post-Quantum Cryptography Standardization Project".
- **2019** - NIST receives 69 proper submissions;
- **2020** - NIST is going to the 3rd round.

Introduction

How a quantum computer works?

- It perform computations based on **probabilities** of an object's state before it is measured;

Introduction

How a quantum computer works?

- It perform computations based on **probabilities** of an object's state before it is measured;
- We can change the probabilities of a **state**;

Introduction

How a quantum computer works?

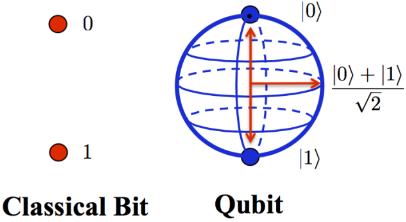
- It perform computations based on **probabilities** of an object's state before it is measured;
- We can change the probabilities of a **state**;

Ok! How can we use a quantum computer?

Table of Contents

Quantum Computation - qubits

Classical bit vs Qubit



$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \; |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\alpha |0\rangle + \beta |1\rangle ,$$

$$|\alpha|^2 + |\beta|^2 = 1$$

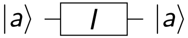
.

Measure quantum state

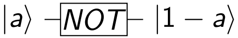
Measuring collapses the state.

Quantum gates

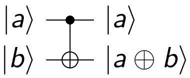
Identity gate:



NOT gate:

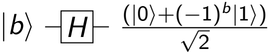


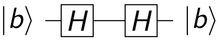
CNOT gate:



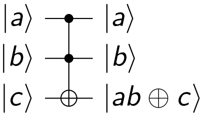
Hadamard Gate:

- $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$





Toffoli gate:



n-Qubit system

Definition

$|\psi\rangle \in \mathbb{C}^2$ such that $|| \, |\psi\rangle \, || = 1$,

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

where

$$\sum_{x \in \{0,1\}^n} |\alpha_x|^2 = 1.$$

Example 2-qubit system

- 4 basis states: $|0\rangle \otimes |0\rangle$, $|0\rangle \otimes |1\rangle$,$|1\rangle \otimes |0\rangle$, $|1\rangle \otimes |1\rangle$.
- It is common to use just: $|0\rangle \, |1\rangle$,$|10\rangle$

Table of Contents

Deutsch-Jozsa problem

- Input: $f : \{0,1\}^n \to \{0,1\}$ either constant or balanced
- Output: 0 iff f is constant
- Constrains: f is a black box

For $n = 1$ we have that
If $f(0) = 0$ and $f(1) = 1$ or $f(0) = 1$ and $f(1) = 0$ the function is balanced.
If $f(0) = 0$ and $f(1) = 0$ or $f(0) = 1$ and $f(1) = 1$ the function is constant.

Query complexity

- Deterministic: $2^{n-1} + 1$

Deutsch-Jozsa problem

- Input: $f : \{0,1\}^n \to \{0,1\}$ either constant or balanced
- Output: 0 iff f is constant
- Constrains: f is a black box

For $n = 1$ we have that
If $f(0) = 0$ and $f(1) = 1$ or $f(0) = 1$ and $f(1) = 0$ the function is balanced.
If $f(0) = 0$ and $f(1) = 0$ or $f(0) = 1$ and $f(1) = 1$ the function is constant.

Query complexity

- Deterministic: $2^{n-1} + 1$
- Quantum: 1

Deutsch-Jozsa quantum circuit

Simple quantum circuit:

$$|b\rangle - \boxed{\;S_f\;} - (-1)^{f(b)}\,|b\rangle$$

Deutsch-Jozsa quantum circuit

Simple quantum circuit:

$$|b\rangle \;-\!\boxed{\;S_f\;}\!-\; (-1)^{f(b)}\,|b\rangle$$

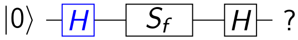
$$|b\rangle \;-\!\boxed{H}\!-\!\boxed{\;S_f\;}\!-\!\boxed{H}\!-\; ?$$

Deutsch-Jozsa quantum circuit analysis

$$|0\rangle \; - \boxed{H} - \boxed{\;S_f\;} - \boxed{H} - \; ?$$

- Initialization: $|0\rangle$.

Deutsch-Jozsa quantum circuit analysis



- Initialization: $|0\rangle$.
- Parallelization: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

Deutsch-Jozsa quantum circuit analysis

$$|0\rangle \; - \boxed{H} - \boxed{S_f} - \boxed{H} - \; ?$$

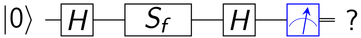
- Initialization: $|0\rangle$.
- Parallelization: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
- Query: $\frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)$.

Deutsch-Jozsa quantum circuit analysis

$$|0\rangle \;-\boxed{H}-\boxed{\;S_f\;}-\boxed{H}-\; ?$$

- Initialization: $|0\rangle$.
- Parallelization: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
- Query: $\frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)$.
- Interferences: $\frac{1}{2}((-1)^{f(0)}(|0\rangle + |1\rangle) + (-1)^{f(1)}(|0\rangle - |1\rangle))$.

Deutsch-Jozsa quantum circuit analysis



- Initialization: $|0\rangle$.
- Parallelization: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
- Query: $\frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)$.
- Interferences: $\frac{1}{2}((-1)^{f(0)}(|0\rangle + |1\rangle) + (-1)^{f(1)}(|0\rangle - |1\rangle))$.
- Final State:
  $\frac{1}{2}(((-1)^{f(0)} + (-1)^{f(1)})|0\rangle + ((-1)^{f(0)} - (-1)^{f(1)})|1\rangle)$.

It is easy to expand for n-qubits.

Deutsch-Jozsa quantum circuit analysis

Deutsch-Jozsa analysis

If $f(0) = 0$ and $f(1) = 1$ or $f(0) = 1$ and $f(1) = 0$
The function is balanced. In our quantum system we will end up with:

$$\frac{1}{2}((0)\,|0\rangle + (2)\,|1\rangle)$$

or

$$\frac{1}{2}((0)\,|0\rangle + (-2)\,|1\rangle)$$

If $f(0) = 0$ and $f(1) = 0$ or $f(0) = 1$ and $f(1) = 1$
The function is constant. In our quantum system we will end up with:

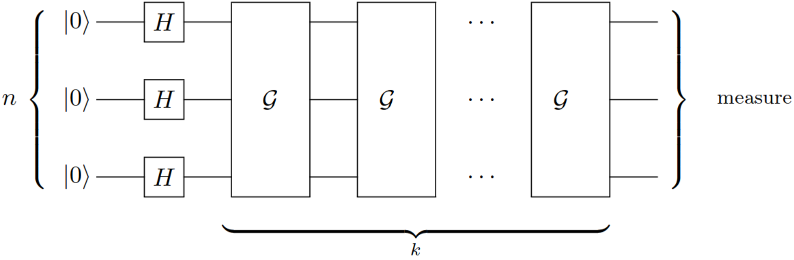
$$\frac{1}{2}((2)\,|0\rangle + (0)\,|1\rangle)$$

or

$$\frac{1}{2}((-2)\,|0\rangle + (0)\,|1\rangle)$$

Table of Contents

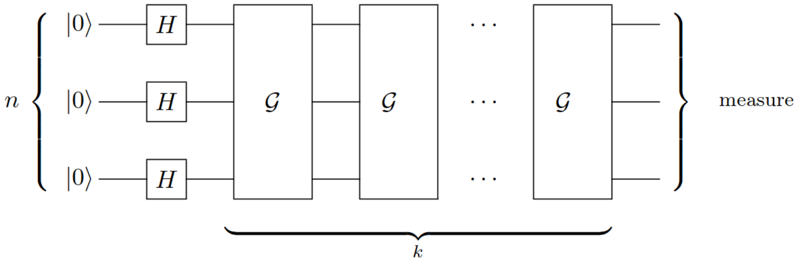
Grover's algorithm in a nutshell



- Originally described as search of an element in an unoreded database.

Grover's Algorithm

Grover's algorithm in a nutshell



- Originally described as search of an element in an unoreded database.
- Needs $O(\sqrt{N})$ queries in database of size $N = 2^n$ elements.

Grover's Algorithm

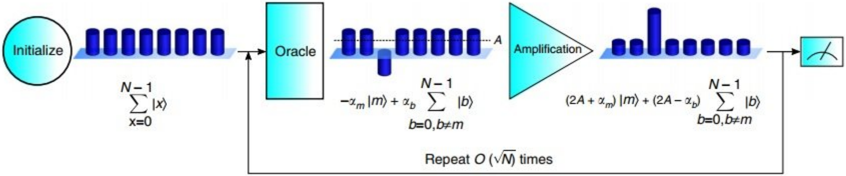
Grover's algorithm in a nutshell

Grover(f, t):

1. Start with $|\phi_0\rangle = |1^n\rangle$
2. Apply $\mathbf{H}^{\otimes n}$
3. Repeat $O\left(\sqrt{2^n}\right)$ times
4. Query to oracle $\mathcal{O}_f$
5. Amplification;
6. Return $x = |\phi\rangle$ with $f(x) = 1$.

Grover's Algorithm

Grover's algorithm in a nutshell



Ok! Can we use Grover's algorithm?

Preimage search

Security of a hash function

Given a hash-function H. The following three security properties should hold:

- Collision resistance: It is computationally infeasible to find any two distinct inputs x, x' which hash to the same output, i.e., such that $H(x) = H(x')$.
- Preimage resistance: It is computationally infeasible to find any preimage x' such that $H(x') = y$ when given any image y.
- 2nd preimage resistance: It is computationally infeasible to find any second input which has the same output as any specified input, i.e., given x, to find a 2nd-preimage $x' \neq x$ such that $H(x) = H(x')$.

Pre-quantum preimage search

Threat to AES

- van Oorschot–Wiener "parallel rho method".
  - Uses a mesh of p small processors.
  - Each running $2^{128}/pt$ fast steps, to find one of t independent AES keys $k_1, \ldots, k_t$, using a fixed plaintext, e.g, AES(0).

NIST has claimed that AES-128 is secure enough.

"Grover's algorithm requires a long-running serial computation, which is difficult to implement in practice. In a realistic attack, one has to run many smaller instances of the algorithm in parallel, which makes the quantum speedup less dramatic."

Introduction - Parallel rho method

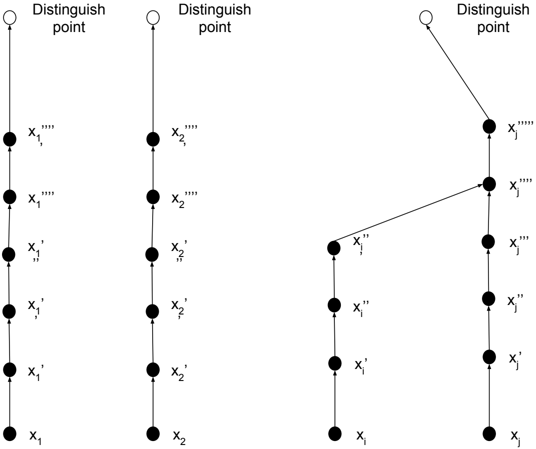
Distinguish Point

Consider $H : \{0,1\}^b \rightarrow \{0,1\}^b$
Take x an input of H, $x' = H(x)$.
Thereafter, take x' and apply H again, $x'' = H(x')$.
It is possible to do it n times (H^n), until a given condition is satified. In our case, we want the first $0 < d < b/2$ bits as 0.
$H^n_d(x)$ means d bits of x, computed n times.

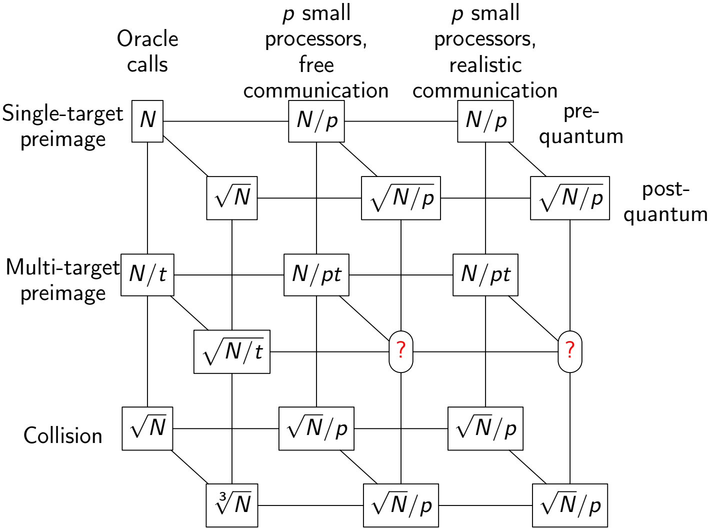
$$H^n_d(x) = \underbrace{0 \ldots 0}_{d \text{ zeros}}\{0,1\}^{b/2}$$

Introduction - Parallel rho method

Distinguish Point



Results in pre and post-quantum preimage search



Grover's algorithm to find a preimage

Grover's algorithm to find a preimage

- Design AES as a quantum circuit.
- Design a quantum circuit for Grover's algorithm that uses the AES quantum circuit.
- Put the previous circuits in p processors using t keys.
- Quantum computer work in a way that requires all algorithms to be reversible.
  - We need an reversible AES circuit to run with Grover's algorithm

Grover's algorithm to find a preimage

Grover's algorithm to find a preimage

- Design AES as a quantum circuit.
- Design a quantum circuit for Grover's algorithm that uses the AES quantum circuit.
- Put the previous circuits in p processors using t keys.
- Quantum computer work in a way that requires all algorithms to be reversible.
  - We need an reversible AES circuit to run with Grover's algorithm
- We need to have low memory/resources.

Distinguish point in quantum setting

Trade-off from Bennett–Tompa

Example to compute $H^4(x)$:

Distinguish point in quantum setting

Trade-off from Bennett–Tompa

Example to compute $H^4(x)$:

time 0:	x	0	0	0	0

Distinguish point in quantum setting

Trade-off from Bennett–Tompa

Example to compute $H^4(x)$:

time 0:	x	0	0	0	0
time 1:	x	0	$H(x)$	0	0

Distinguish point in quantum setting

Trade-off from Bennett–Tompa

Example to compute $H^4(x)$:

time 0:	x	0	0	0	0
time 1:	x	0	$H(x)$	0	0
time 2:	x	0	$H(x)$	$H^2(x)$	0

Distinguish point in quantum setting

Trade-off from Bennett–Tompa

Example to compute $H^4(x)$:

time 0:	x	0	0	0	0
time 1:	x	0	$H(x)$	0	0
time 2:	x	0	$H(x)$	$H^2(x)$	0
time 3:	x	0	$H(x)$	$H^2(x)$	$H^3(x)$

Distinguish point in quantum setting

Trade-off from Bennett–Tompa

Example to compute $H^4(x)$:

time 0:	x	0	0	0	0
time 1:	x	0	$H(x)$	0	0
time 2:	x	0	$H(x)$	$H^2(x)$	0
time 3:	x	0	$H(x)$	$H^2(x)$	$H^3(x)$
time 4:	x	$H^4(x)$	$H(x)$	$H^2(x)$	$H^3(x)$

Distinguish point in quantum setting

Trade-off from Bennett–Tompa

Example to compute $H^4(x)$:

time 0:	x	0	0	0	0
time 1:	x	0	$H(x)$	0	0
time 2:	x	0	$H(x)$	$H^2(x)$	0
time 3:	x	0	$H(x)$	$H^2(x)$	$H^3(x)$
time 4:	x	$H^4(x)$	$H(x)$	$H^2(x)$	$H^3(x)$
time 5:	x	$H^4(x)$	$H(x)$	$H^2(x)$	0

Distinguish point in quantum setting

Trade-off from Bennett–Tompa

Example to compute $H^4(x)$:

time 0:	x	0	0	0	0
time 1:	x	0	$H(x)$	0	0
time 2:	x	0	$H(x)$	$H^2(x)$	0
time 3:	x	0	$H(x)$	$H^2(x)$	$H^3(x)$
time 4:	x	$H^4(x)$	$H(x)$	$H^2(x)$	$H^3(x)$
time 5:	x	$H^4(x)$	$H(x)$	$H^2(x)$	0
time 6:	x	$H^4(x)$	$H(x)$	0	0

Distinguish point in quantum setting

Trade-off from Bennett–Tompa

Example to compute $H^4(x)$:

time 0:	x	0	0	0	0
time 1:	x	0	$H(x)$	0	0
time 2:	x	0	$H(x)$	$H^2(x)$	0
time 3:	x	0	$H(x)$	$H^2(x)$	$H^3(x)$
time 4:	x	$H^4(x)$	$H(x)$	$H^2(x)$	$H^3(x)$
time 5:	x	$H^4(x)$	$H(x)$	$H^2(x)$	0
time 6:	x	$H^4(x)$	$H(x)$	0	0
time 7:	x	$H^4(x)$	0	0	0

$$\begin{array}{ccc} x_1 & \cdots & x_t \\ \downarrow & & \downarrow \\ H_d(x_1) & \cdots & H_d(x_t) \\ \vdots & & \vdots \\ H_d^n(x_1) & \cdots & H_d^n(x_t) \end{array}$$

$$
\begin{array}{ccc}
x_1 & \cdots & x_t \\
\downarrow & & \downarrow \\
H_d(x_1) & \cdots & H_d(x_t) \\
\vdots & & \vdots \\
H_d^n(x_1) & \cdots & H_d^n(x_t)
\end{array}
$$

$$
\begin{array}{ccc}
y_1 & \cdots & y_t \\
\downarrow & & \downarrow \\
H_d(y_1) & \cdots & H_d(y_t) \\
\vdots & & \vdots \\
H_d^n(y_1) & \cdots & H_d^n(y_t)
\end{array}
$$

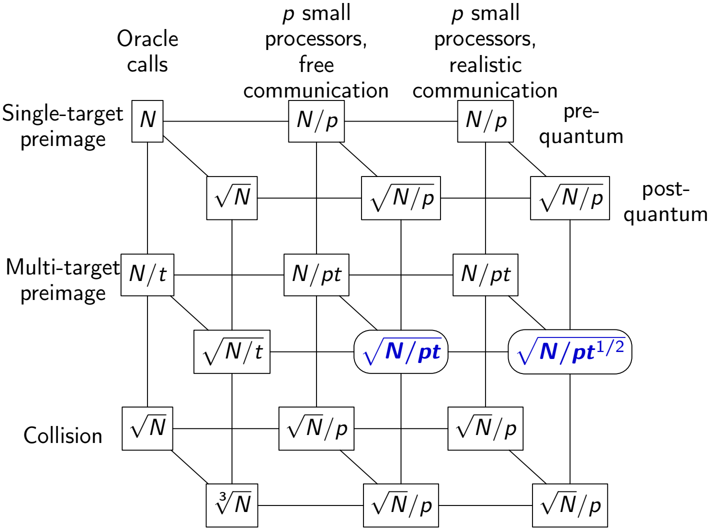
$$
\begin{array}{ccc}
x_1 & \cdots & x_t \\
\downarrow & & \downarrow \\
H_d(x_1) & \cdots & H_d(x_t) \\
\vdots & & \vdots \\
H_d^n(x_1) & \cdots & H_d^n(x_t) \\
 & & \\
y_1 & \cdots & y_t \\
\downarrow & & \downarrow \\
H_d(y_1) & \cdots & H_d(y_t) \\
\vdots & & \vdots \\
H_d^n(y_1) & \cdots & H_d^n(y_t)
\end{array}
\qquad\qquad
H_d^n(y_i) \stackrel{?}{=} H_d^n(x_j)
$$

Low-communication parallel quantum multi-target preimage search

Gustavo Banegas & Daniel J. Bernstein

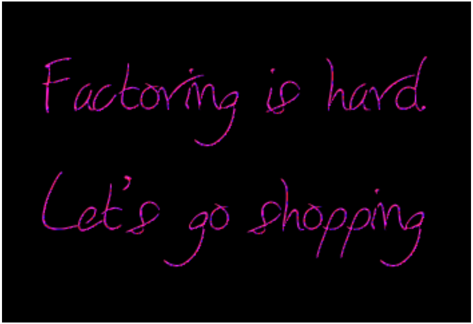
- Bennett-Tompa technique to build a reversible circuit for distinguished points.
- Possible to achieve using low communication costs and no memory.

Result:


p small processors, free communication
p small processors, realistic communication
Oracle calls
Single-target preimage
Multi-target preimage
Collision
pre-quantum
post-quantum
N
N/p
N/p
√N
√N/p
√N/p
N/t
N/pt
N/pt
√N/t
√N/pt
√N/pt^{1/2}
√N
√N/p
√N/p
∛N
√N/p
√N/p

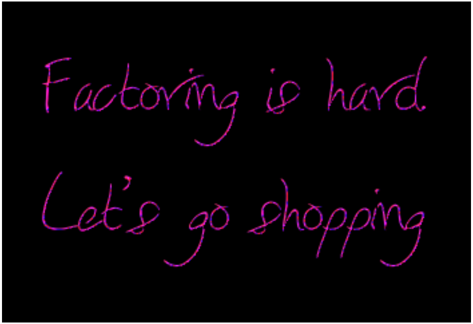
Factoring prime numbers

Factoring Integers with Shor's algorithm



Factoring prime numbers

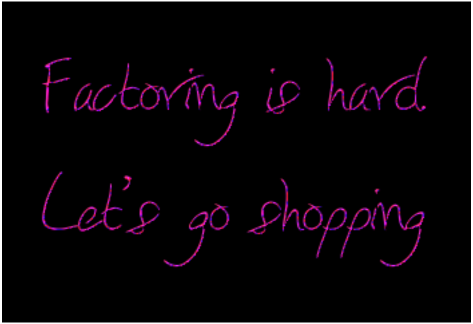
Factoring Integers with Shor's algorithm



- Develop by Peter Shor in 1994;

Factoring prime numbers

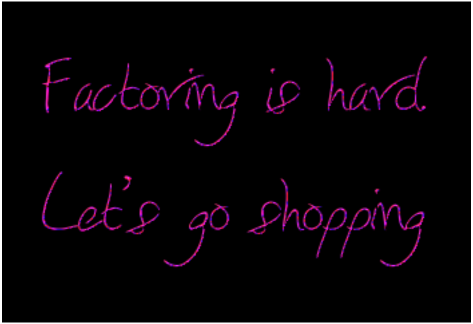
Factoring Integers with Shor's algorithm



- Develop by Peter Shor in 1994;
- Brings apocalypse to cryptography;

Factoring prime numbers

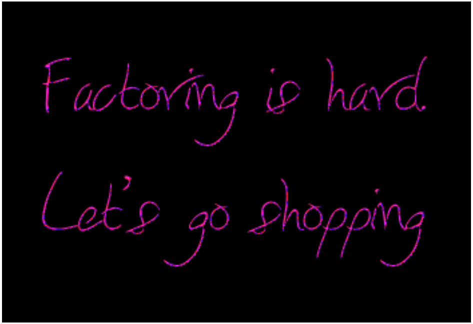
Factoring Integers with Shor's algorithm



- Develop by Peter Shor in 1994;
- Brings apocalypse to cryptography;
- It breaks RSA, ECDSA and DSA;

Factoring prime numbers

Factoring Integers with Shor's algorithm



- Develop by Peter Shor in 1994;
- Brings apocalypse to cryptography;
- It breaks RSA, ECDSA and DSA;
- How many qubits and gates do we need to run Shor's algorithm?

Shor's algorithm

In summary Shor's algorithm has two parts:

- A reduction of the factoring problem to the problem of **order-finding**, which can be done on a classical computer;

Shor's algorithm

In summary Shor's algorithm has two parts:

- A reduction of the factoring problem to the problem of **order-finding**, which can be done on a classical computer;
- A quantum algorithm to solve the **order-finding** problem.

Shor's algorithm

A toy example can be when we have $N = 15$. Let's see how Shor's algorithm works:

1 Select an arbitrary number, such as $a = 2$ (< 15)

2 $gcd(a, N) = gcd(2, 15) = 1$

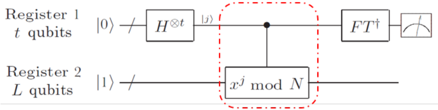
3 Find the period of function $f(x) = a^x \mod N$, which satisfies $f(x + r) = f(x)$;

4 Get $r = 4$ through the circuit below;

5 $gcd(a^{\frac{r}{2}} + 1, N) = gcd(5, 15) = 5$;

6 $gcd(a^{\frac{r}{2}} - 1, N) = gcd(3, 15) = 5$;

7 For $N = 15$, the two decomposed prime numbers are 3 and 5.



Ressource Estimation

Break RSA (Integer Factoring)

From Gidney & Ekerå(2019)[2] uses "$3n + 0.002 n lg(n)$ logical qubits, $0.3n^3 + 0.0005 n^3 lg(n)$ Toffolis, and $500n^2 + n^2 lg(n)$ measurement depth to factor n-bit RSA integers"

RSA Bits	Qubits	Toffoli + T Gates (billions)
1024	3092	0.4
2048	6189	2.7
3072	9287	9.9

[2]Craig Gidney and Martin Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. arXiv preprint quant-ph/1904.09749, 2019. https://arxiv.org/abs/1905.09749

Ressource Estimation

Break Binary ECC (DLP)

From Banegas, Bernstein, von Hoof and Lange(2021)[3] we have that for breaking binary ECC we have $7n + \lfloor \log(n) \rfloor + 9$ qubits, $48n^3 + 8n^{\log(3)+1} + 352n^2 \log(n) + 512n^2 + O(n^{\log(3)})$ Toffoli gates and $O(n^3)$ CNOT gates.

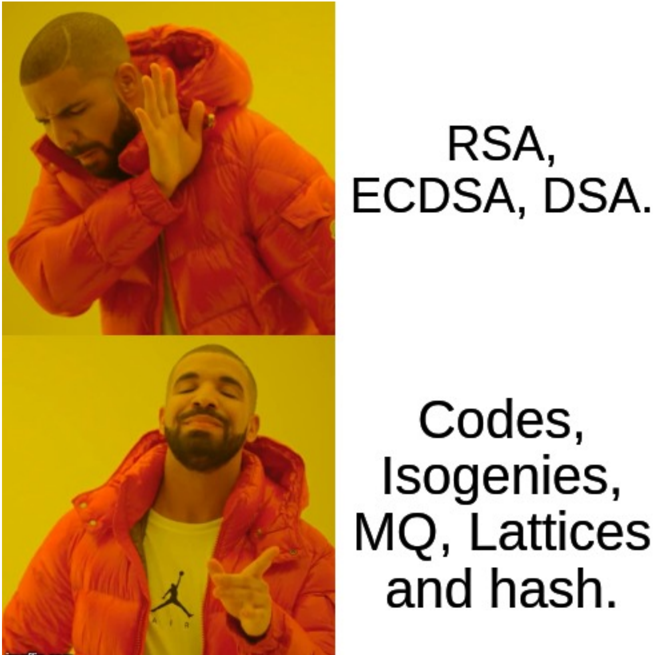
		Single step			Total
n	qubits	TOF gates	CNOT gates	depth upper bound	TOF gates
163	1,157	893,585	827,379	1,262,035	293,095,880
233	1,647	1,669,299	1,614,947	2,405,889	781,231,932
283	1,998	2,427,369	2,358,734	3,503,510	1,378,745,592
571	4,015	8,987,401	9,080,190	13,237,682	10,281,586,744

[3] Banegas, G., Bernstein, D. J., van Hoof, I., Lange, T. Concrete quantum cryptanalysis of binary elliptic curves. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2021(1)

Other Quantum algorithms

- Simon's Algorithm (QFT);
- Ambaini's Algorithm (Element disticness);
- Claw finding Algorithm;
- Kuperberg's Algorithm (dihedral hidden subgroup problem);

Remember....



Questions

Thank you for your attention.
Questions?
gustavo@cryptme.in