

Post Quantum Cryptography and the future of the Internet

Willy Malvault

Snowcamp 2023



Merci à nos sponsors

« Etoile »

VISEO
— POSITIVE DIGITAL MAKERS —

open
WE EMPOWER
YOUR DIGITAL WORLD

CRITEO

gravitee.io

**sopra
steria**

zenika

THALES
Building a future we can all trust

ABYLSSEN

Microsoft

salesforce

CGI

OVHcloud

« Flocon »

AVISTO

diabeloop

LIGHT UP
YOUR FUTURE...
**HARDIS
GROUP**

KLS GROUP
La French Logistique

kelkoogroup Moody's

It all begins with

← Tweet



Stéphane Bortzmeyer
@bortzmeyer

...

Ce mardi 5, l'organisme de normalisation étatsunien
#NIST a annoncé qu'il avait choisi les algorithmes de
cryptographie post-quantiques qu'il allait maintenant
normaliser. Ce sont Kyber pour l'échange de clés et
Dilithium pour les signatures.
bortzmeyer.org/nist-pq.html

#quantique

9:54 AM · 6 juil. 2022 · Twitter Web App

32 Retweets 1 Citer le Tweet 47 J'aime

My Approach

- Willy Malvault
- Principal Consultant at Sogilis
 - Cloud Native related stuff
 - Cybersecurity
- I started my career because of the need to understand how Internet works
- I needed to understand Post Quantum Cryptography

The image is a very blurry and overexposed scan of a document or whiteboard. It contains several mathematical expressions and diagrams that are difficult to decipher due to the poor quality of the image. Some legible elements include:

- $\sqrt{\frac{1}{12} + \frac{1}{48}}$
- $(x+y)$
- $\sqrt{x-y}$
- x^2
- b
- z
- x^2
- $\Delta = \frac{1}{2}bh$
- $y^2 = \frac{\sqrt{y}}{x+2}$
- $\sqrt{\frac{1}{12} + \frac{1}{48}}$
- $x-y$
- y^{x^2}
- $(x+y)^2$
- $(a+b)$

What is a public key cryptography scheme ?

$Pk, Sk = \text{Keygen}()$

Pk = Public Key

Sk = Secret Key

What is a public key cryptography scheme ?

$Pk, Sk = Keygen()$



Encryption

$c = Enc(Pk, m)$

$m = Dec(Sk, c)$

$Pk = \text{Public Key}$

$Sk = \text{Secret Key}$

What is a public key cryptography scheme ?

$Pk, Sk = Keygen()$



Encryption

$c = Enc(Pk, m)$

$m = Dec(Sk, c)$

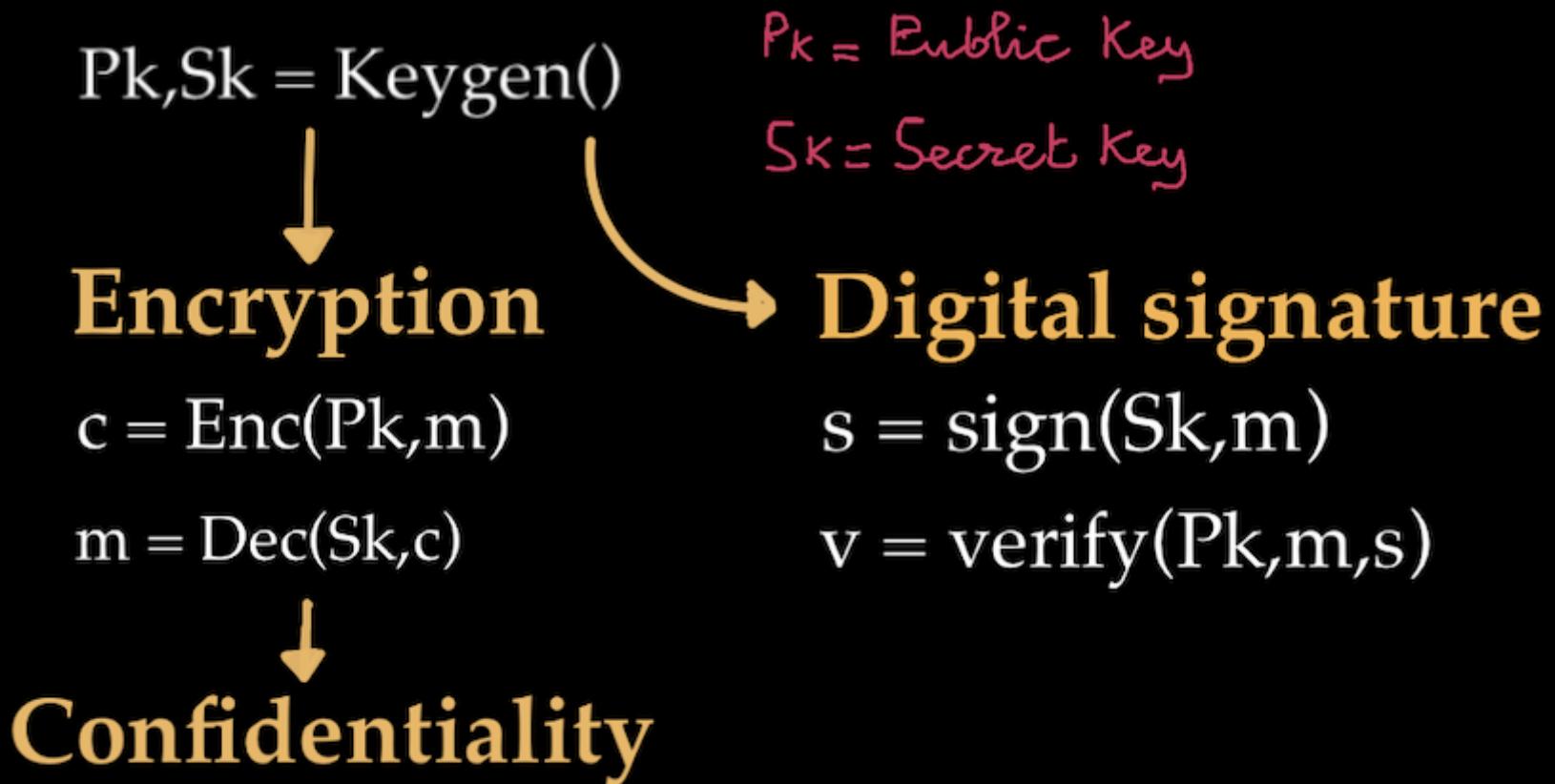
$Pk = \text{Public Key}$

$Sk = \text{Secret Key}$

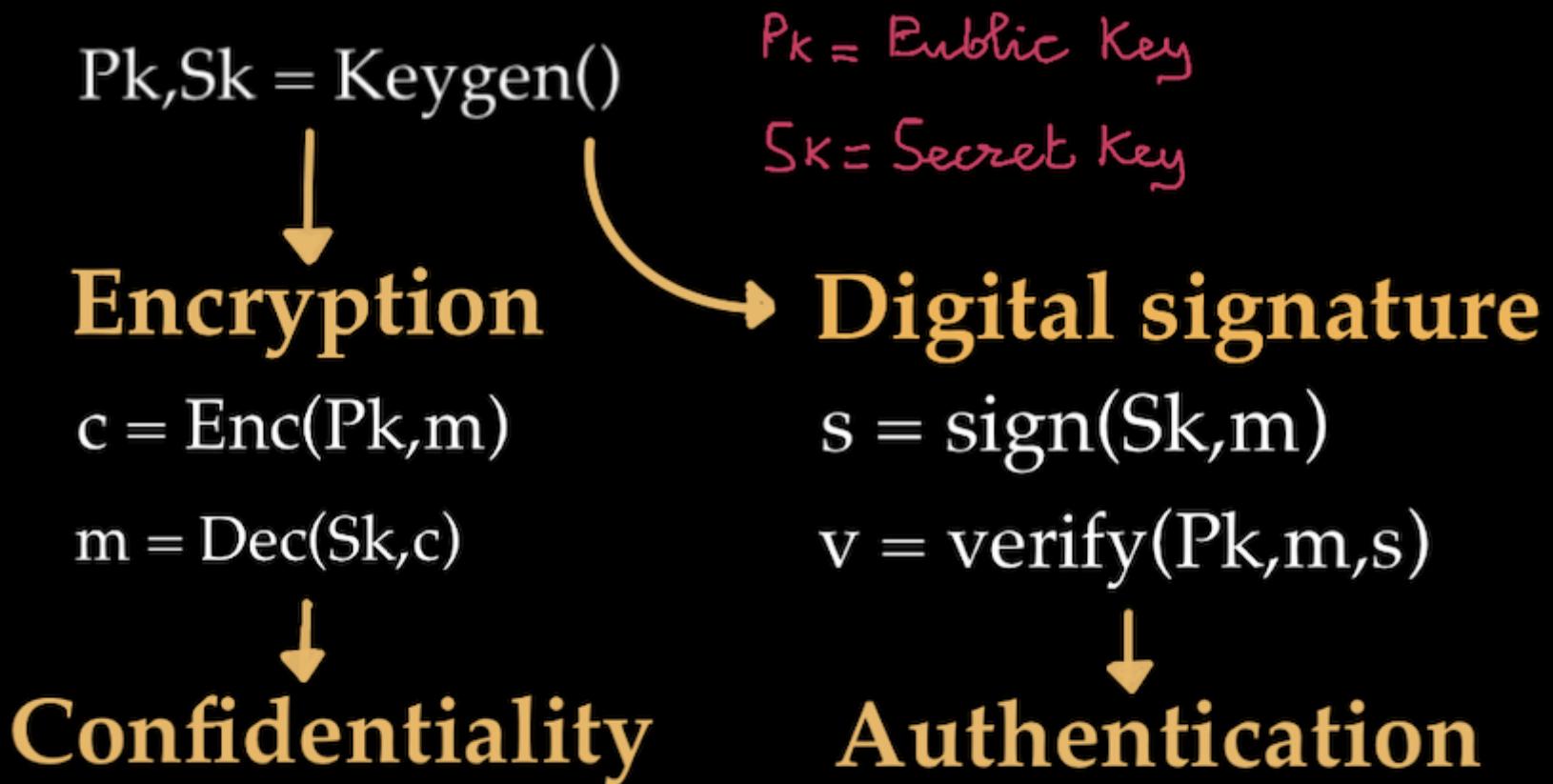


Confidentiality

What is a public key cryptography scheme ?



What is a public key cryptography scheme ?



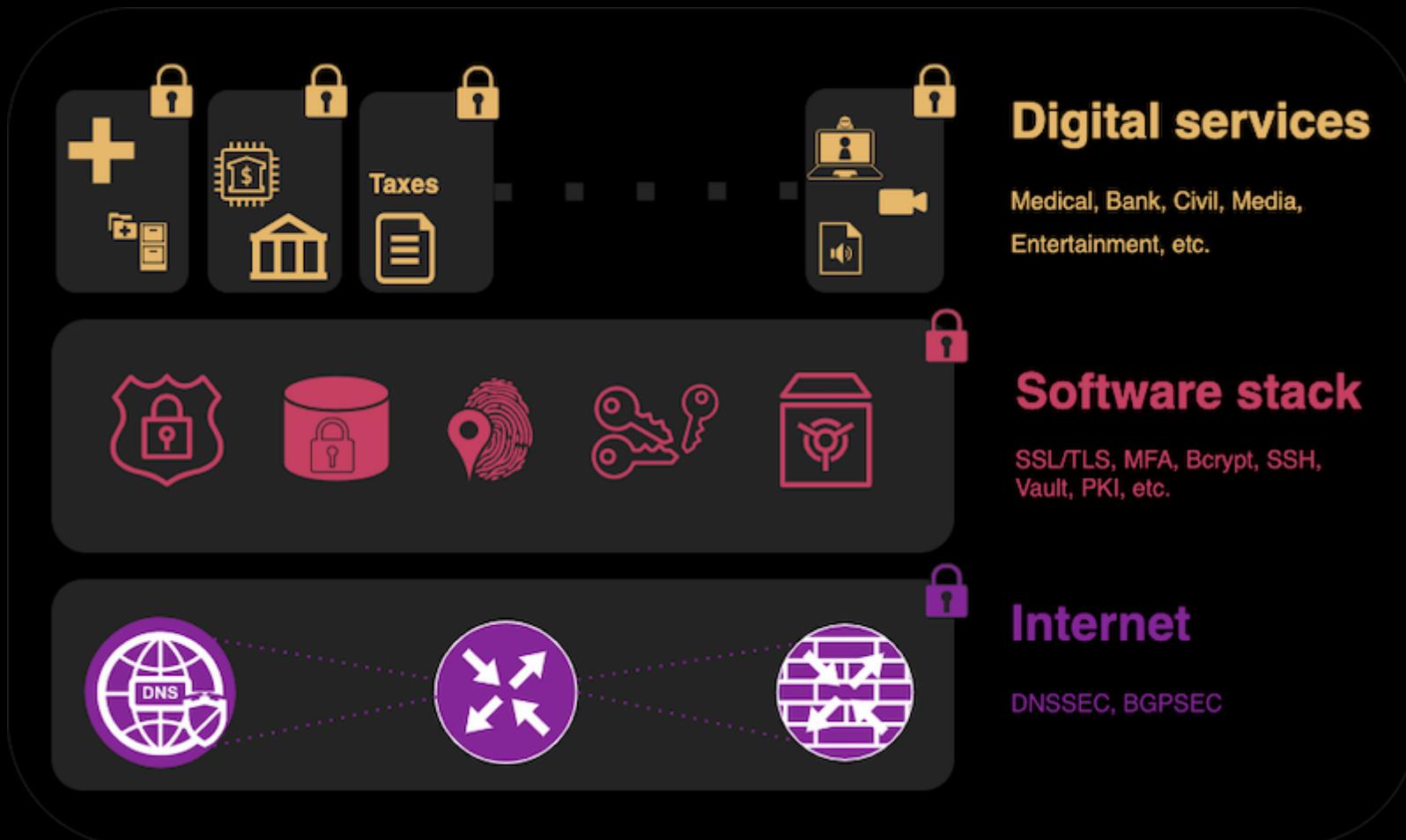
What's the point ?



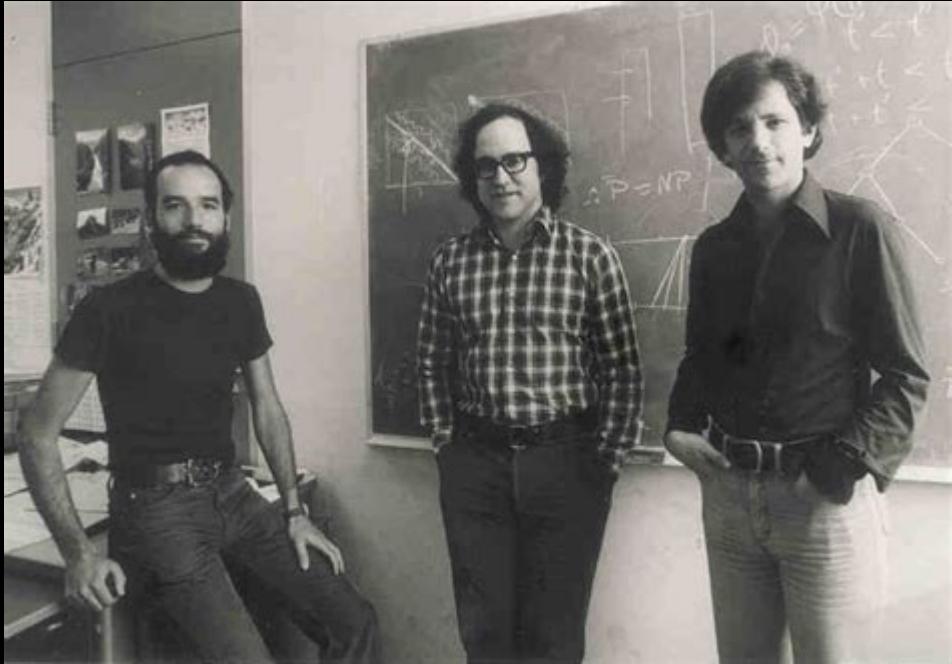
What's the point ?



What's the point ?

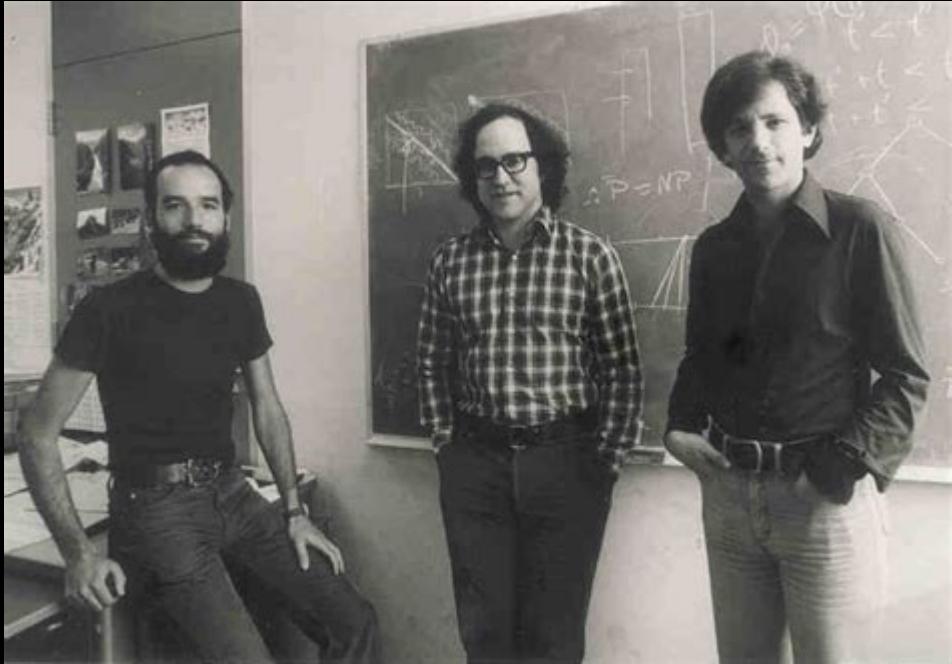


How to implement public key cryptography ?



Ronald L. Rivest, Adi Shamir, Leonard M. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.
Commun. ACM 21(2): 120-126 (1978)

How to implement public key cryptography ?



Ronald L. Rivest, Adi Shamir, Leonard M. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.
Commun. ACM 21(2): 120-126 (1978)

Prime number factorization

$N = ? \times ?$

Hard!

Prime number factorization

$$N = ? \times ? \rightarrow 15 \sim 0,01\text{s}$$

Hard!

Prime number factorization

$N = ? \times ? \rightarrow 15 \sim 0,01s$

Hard!

542334806886579084945801229
632589528976540003506920
0613911119134831511204958711

~ 9 minutes

Prime number factorization

$$N = ? \times ? \rightarrow 15 \sim 0,01s$$

Hard!

542334806886579084945801229
632589528976540003506920
0613911119134831511204958711

~ 9 minutes

123018668453011775513049495838496272077
6765434567898654244568976787643245679998
7654322245678999865443322234678998765432
3446789997654433223445789087655433222234
556778986543223456799086554332223346788
8776554323578008765432-23344555567888995

~ 2000 years (1 CPU)

~ 30 days (24000 CPU)

RSA cryptosystem (simplified)

KeyGen()

$$n = p * q$$

$$Pk = (c(p, q), n)$$

$$Sk = (\text{mmi}(p, q), n)$$

Enc(pk, m)

$$\backslash (c = m^{\wedge} \{ Pk \} \backslash \bmod \backslash n \backslash)$$

Dec(sk, c)

$$\backslash (m = c^{\wedge} \{ sk \} \backslash \bmod \backslash n \backslash)$$

RSA cryptosystem (simplified)

KeyGen()

$n = p * q$

$Pk = (c(p, q), n)$

$Sk = (mmi(p, q), n)$

Enc(pk, m)

$\backslash (c = m^{\wedge} \{ Pk \} \backslash \bmod \n \backslash)$

Dec(sk, c)

$\backslash (m = c^{\wedge} \{ sk \} \backslash \bmod \n \backslash)$

Sign(sk, m)

$\backslash (s = \text{hash}(m) ^{\wedge} \{ sk \} \backslash \bmod \n \backslash)$

Verify(pk, m)

$\backslash (h = s ^{\wedge} \{ Pk \} \backslash \bmod \n \backslash)$

OK if $\backslash (\text{hash}(m) = h \backslash)$

RSA cryptosystem (simplified)

KeyGen()

$$n = p * q$$

$$Pk = (c(p, q), n)$$

$$Sk = (\text{mmi}(p, q), n)$$

Enc(pk, m)

$$\backslash (c = m^{\wedge} \{ Pk \} \backslash \bmod \backslash n \backslash)$$

Dec(sk, c)

$$\backslash (m = c^{\wedge} \{ sk \} \backslash \bmod \backslash n \backslash)$$

Sign(sk, m)

$$\backslash (s = \text{hash}(m) ^{\wedge} \{ sk \} \backslash \bmod \backslash n \backslash)$$

Verify(pk, m)

$$\backslash (h = s^{\wedge} \{ Pk \} \backslash \bmod \backslash n \backslash)$$

OK if $\backslash (\text{hash}(m) = h \backslash)$

RSA is working great !

The RSA problem/challenge

If an attacker can retrieve p and q from n , then she can retrieve S_k

It takes around 1000 core-years to factorize RSA-768 (record), with a classical computer.

Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thomé, Paul Zimmermann: Comparing the difficulty of factorization and discrete logarithm: a 240-digit experiment. IACR Cryptol. ePrint Arch. 2020: 697 (2020)

RSA is working great !

The RSA problem/challenge

If an attacker can retrieve p and q from n , then she can retrieve S_k

It takes around 1000 core-years to factorize RSA-768 (record), with a classical computer.

Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thomé, Paul Zimmermann: Comparing the difficulty of factorization and discrete logarithm: a 240-digit experiment. IACR Cryptol. ePrint Arch. 2020: 697 (2020)

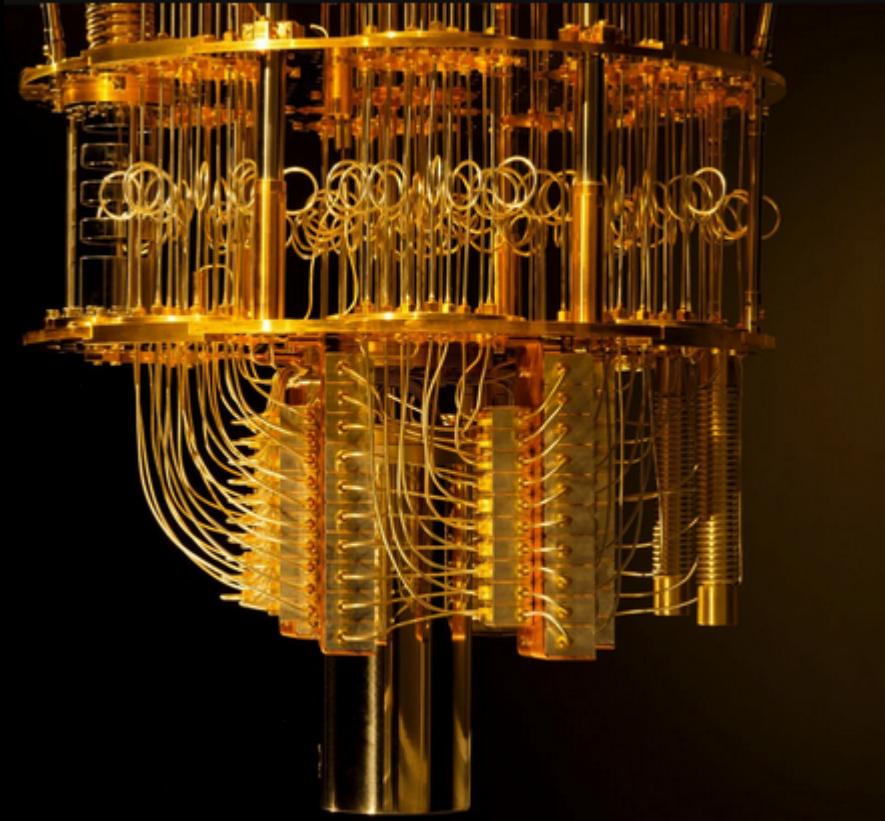
NIST standardization

Definitions:

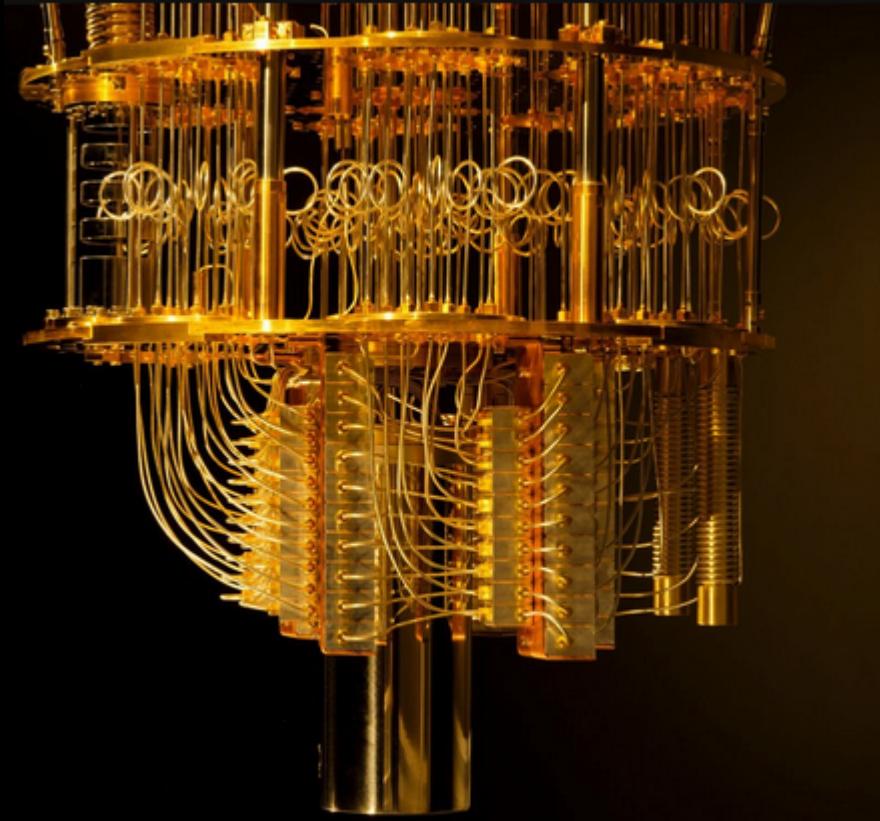
Algorithm developed by Rivest, Shamir and Adleman

But... this stands only for classical
computers

Quantum computers is threatening !



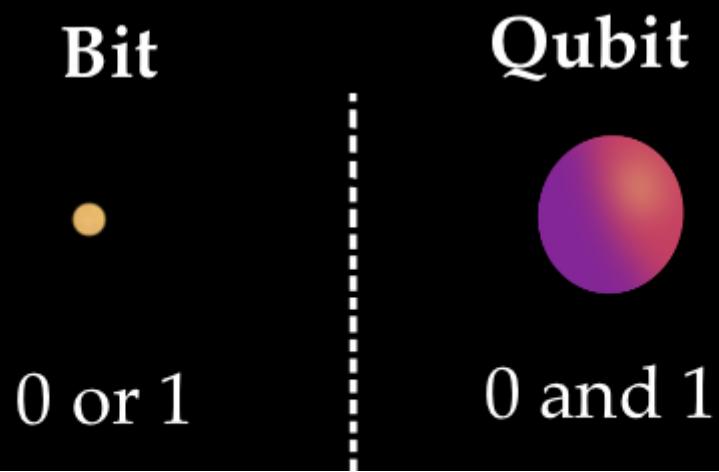
Quantum computers is threatening !



Orbit

Why is the qubit so
powerfull ?

Why is the qubit so powerfull ?



Quantum supremacy

Is the ability of quantum computing to outperform classical computing.

Typically resolving exponential time problems in polynomial time.

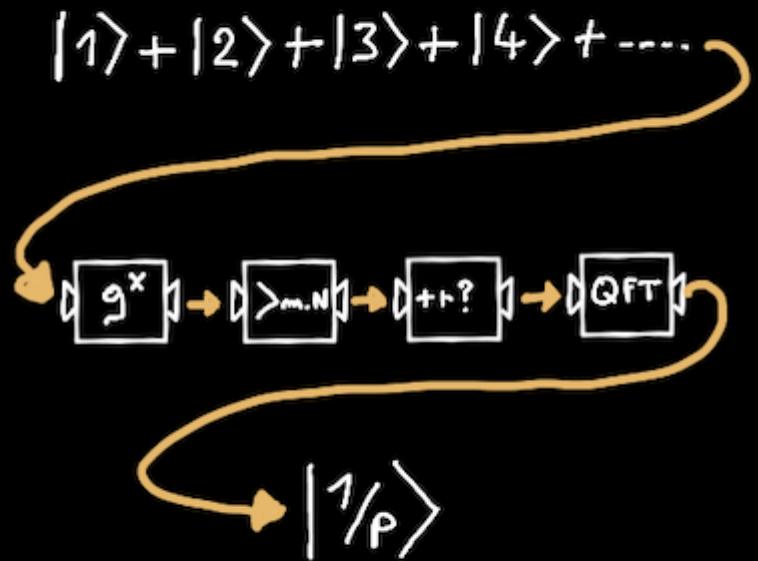
Strong at resolving high combinatory Problems:

Pharma, Nuclear, Meteo

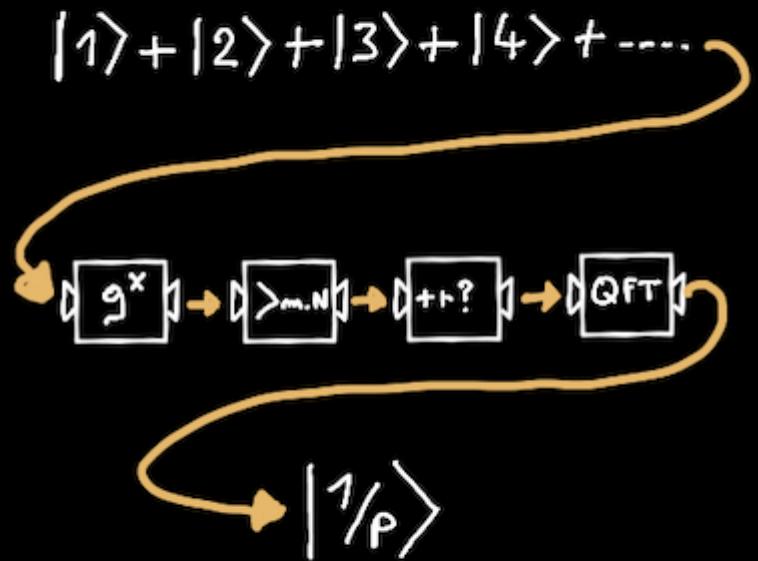
Artificial intelligence

Breaking RSA !

The Famous Shor algorithm



The Famous Shor algorithm

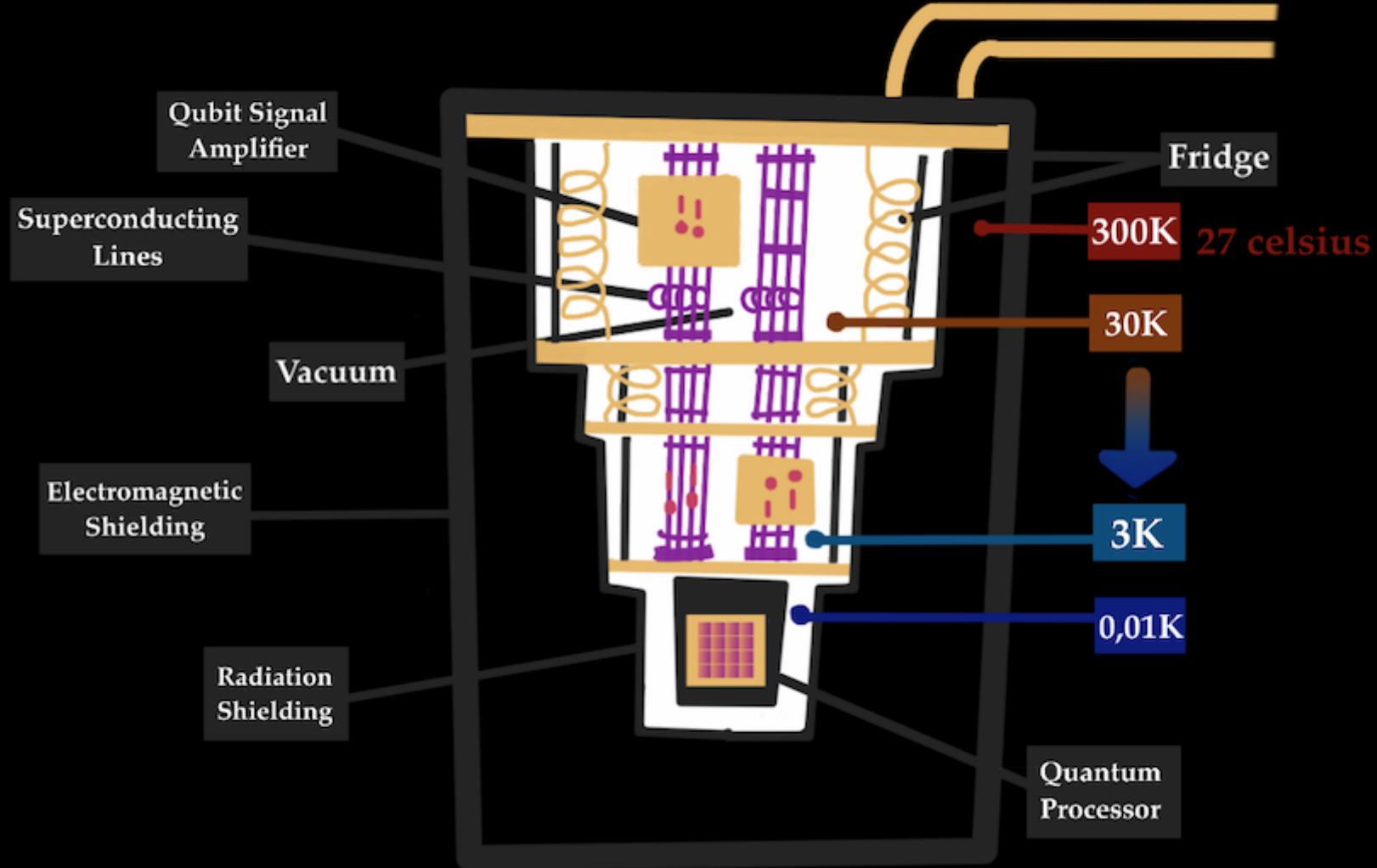


Should we be afraid of the monster ?



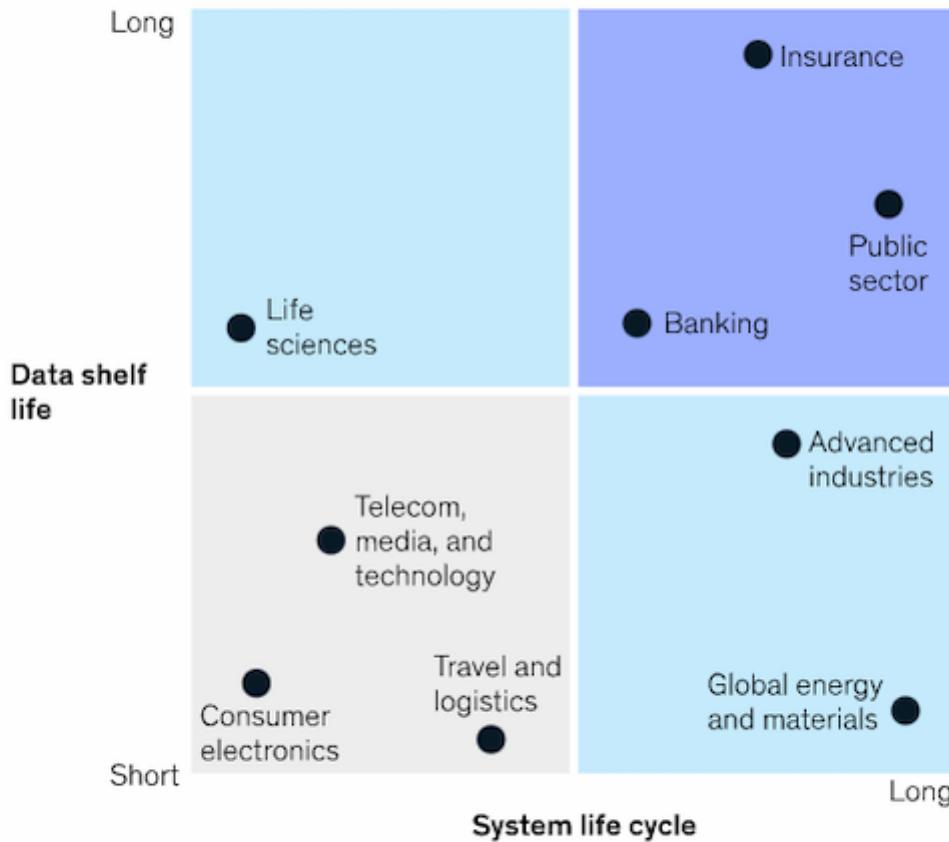
Should we be afraid of the monster ?





Risk of quantum-powered attack by industry

■ At risk before ~2025 ■ At risk between ~2025 and ~2030 ■ At risk after ~2030



Quantum Cryptography

Using quantum channels to exchange private keys.

C. H. Bennett and G. Brassard. "Quantum cryptography: Public key distribution and coin tossing". In Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, volume 175, page 8. New York, 1984

Quantum Cryptography

Using quantum channels to exchange private keys.

C. H. Bennett and G. Brassard. "Quantum cryptography: Public key distribution and coin tossing". In Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, volume 175, page 8. New York, 1984

Post-quantum Cryptography

Use traditionnal computers to use cryptography schemes robust to quantum attacks.

NIST PQC challenge

 NIST PQC standardization page screenshot

Selected protocols

Public Key encryption and key-establishment

Kyber: an IND-CCA2-secure key-encapsulation mechanism (KEM)

Digital signature

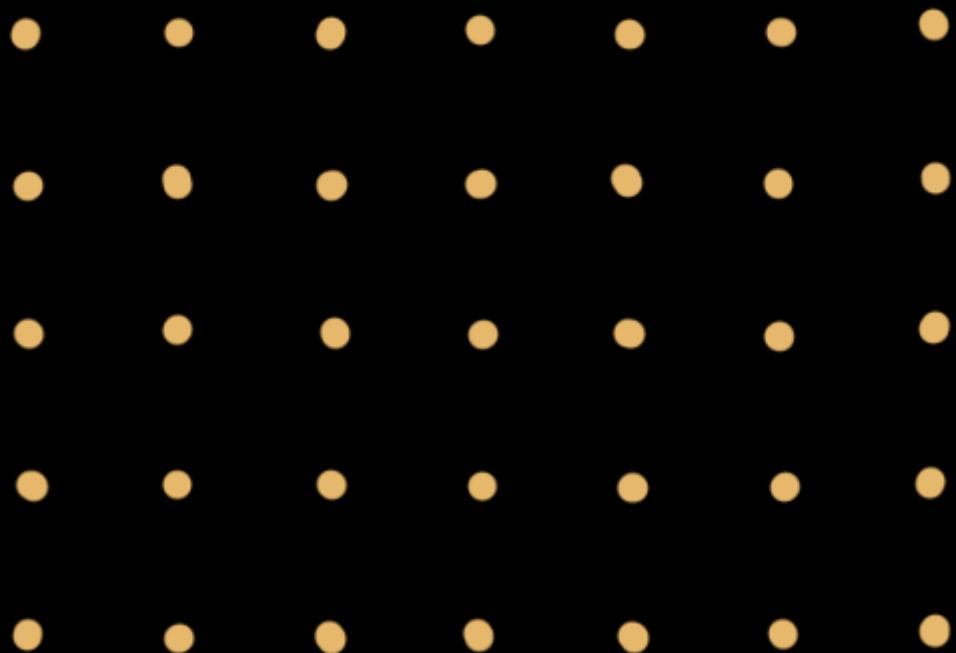
Dilithium: a strongly EUF-CMA-secure digital signature algorithm

Falcon: Fast-Fourier lattice-based compact signatures over NTRU

SPHINCS+: is a stateless hash-based signature scheme.

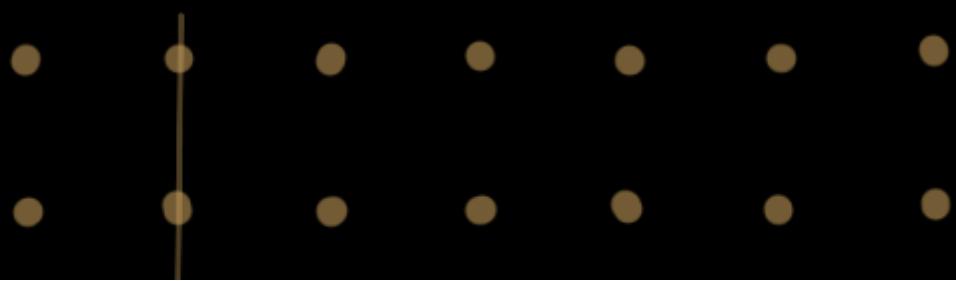
Introduction to lattice

- Infinite set of points in some N dimensional space



Introduction to lattice

- Infinite set of points in some N dimensional space
- Linear combination of vectors from a basis $\{b_1, b_2, \dots, b_N\}$ of (\mathbb{R}^N)
- $L = \left\{ \sum a_i b_i : a_i \in \mathbb{Z} \right\}$



Introduction to lattice

- Infinite set of points in some N dimensional space
- Linear combination of vectors from a basis $\{b_1, b_2, \dots, b_N\}$ of (\mathbb{R}^N)
- $L = \left\{ \sum_{i=1}^n a_i b_i : a_i \in \mathbb{Z} \right\}$

Fundamental hard problem

- Closest Vector Problem (CVP)
- Shortest Vector Problem (SVP)

Introduction to lattice

- Infinite set of points in some N dimensional space
- Linear combination of vectors from a basis $\{b_1, b_2, \dots, b_N\}$ of (\mathbb{R}^N)
- $L = \left\{ \sum_{i=1}^n a_i b_i : a_i \in \mathbb{Z} \right\}$

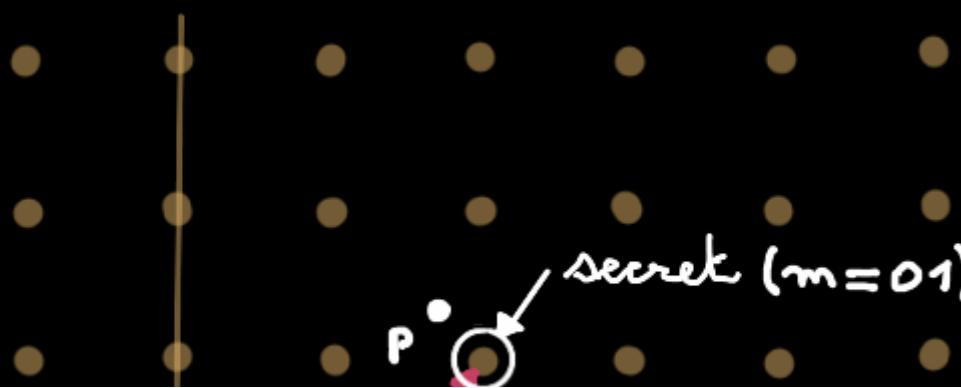
Fundamental hard problem

- Closest Vector Problem (CVP)
- Shortest Vector Problem (SVP)

Learning With Error (LWE)

Encrypt with Lattice

- Good basis $\langle (b_1, b_2) \rangle$ is the private key
- Bad basis $\langle (b_1', b_2') \rangle$ is the public key
- Encode bit with coordinates of a lattice point (according to public key, by instance)



Encrypt with Lattice

- Good basis $\langle (b_1, b_2) \rangle$ is the private key
- Bad basis $\langle (b_1', b_2') \rangle$ is the public key
- Encode bit with coordinates of a lattice point (according to public key, by instance)

Decrypt with Lattice

- Try to surround the point

Encrypt with Lattice

- Good basis $\langle (b_1, b_2) \rangle$ is the private key
- Bad basis $\langle (b_1', b_2') \rangle$ is the public key
- Encode bit with coordinates of a lattice point (according to public key, by instance)

Decrypt with Lattice

- Try to surround the point
- Easy to find with good basis

Encrypt with Lattice

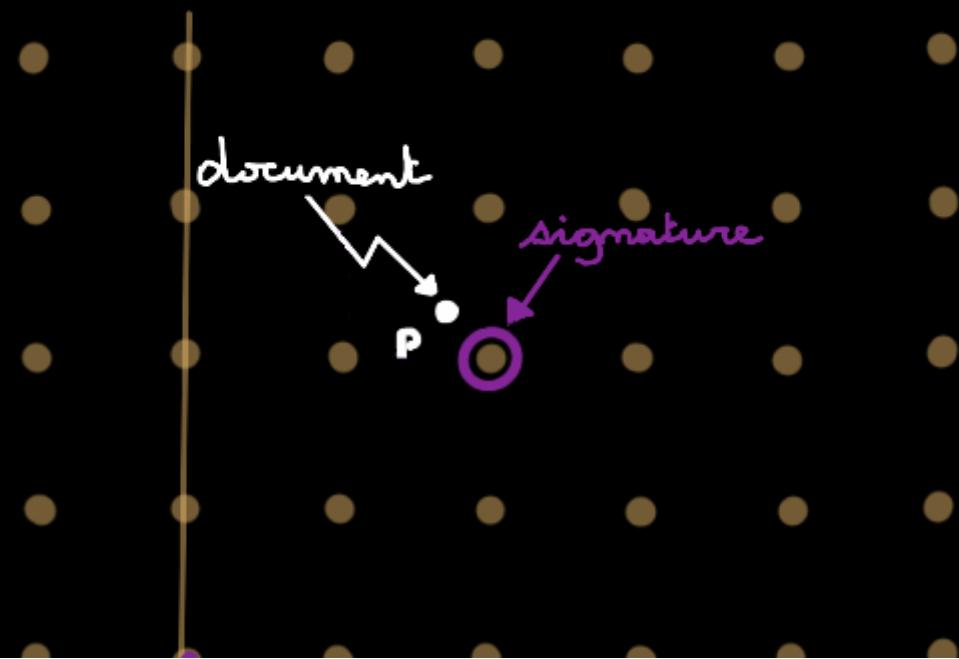
- Good basis $\langle (b_1, b_2) \rangle$ is the private key
- Bad basis $\langle (b_1', b_2') \rangle$ is the public key
- Encode bit with coordinates of a lattice point (according to public key, by instance)

Decrypt with Lattice

- Try to surround the point
- Easy to find with good basis
- Hard to find with bad basis (or no basis)

Lattice digital signature

- Encode document with point in space
- Closest Vector is the signature

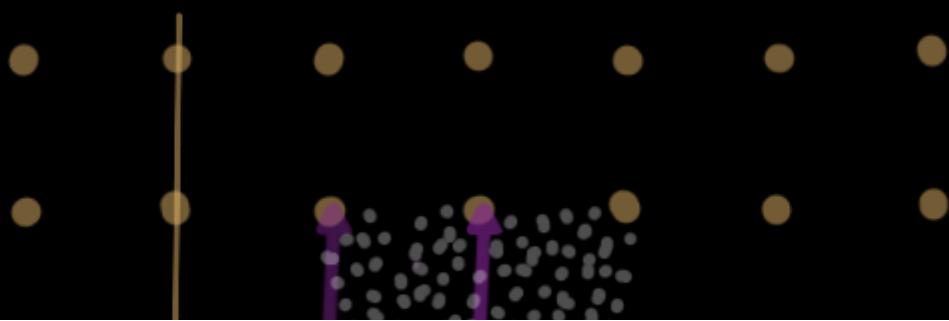


Lattice digital signature

- Encode document with point in space
- Closest Vector is the signature

Disadvantages

- Informations about private key can leak with a great number of signatures.



From lattice encryption to Kyber

From lattice encryption to Kyber

- Use polynomial rings rather than integers. Ring-LWE, then even Module-LWE (NTRU)

$$\begin{array}{r} 43 \quad 67 \\ \times 163 \end{array} \left| \begin{array}{c} x^3 + 2x + 6 \\ x^2 + x - 7 \end{array} \right. \begin{array}{l} x-1 \\ \hline \end{array}$$

From lattice encryption to Kyber

- Use polynomial rings rather than integers. Ring-LWE, then even Module-LWE (NTRU)

$$\begin{array}{r} 43 \quad 67 \\ \times \quad 163 \\ \hline 163 \quad | \quad x^3 + 2x + 6 \\ \quad \quad x^2 + x - 7 \end{array}$$

We can add and multiply

$$\begin{array}{r} 43 \times 67 \\ = 2881 \\ \hline | \quad (x^2 + x - 7)(x - 1) \\ \quad \quad \quad = x^3 - 8x + 7 \end{array}$$

From lattice encryption to Kyber

- Use polynimial rings rather than integers. Ring-LWE, then even Module-LWE (NTRU)

$$\begin{array}{r|l} 43 & \begin{array}{c} 67 \\ 163 \end{array} \\ \hline & \begin{array}{c} x^3 + 2x + 6 \\ x^2 + x - 7 \end{array} \end{array}$$

We can add and multiple

$$\begin{array}{r|l} 43 \times 67 & (x^2 + x - 7)(x - 1) \\ \hline = 2881 & = x^3 - 8x + 7 \end{array}$$

From lattice encryption to Kyber

- Use polynomial rings rather than integers. Ring-LWE, then even Module-LWE (NTRU)
- Public Key can be built using a square Matrix

$$\begin{pmatrix} 5 & -2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 4 \end{pmatrix} + (2, 7) = (9, 11)$$



$$(m_1, \dots, m_{n-1}, m_n + \sqrt{z^2 + s})$$

Kyber matrix

$$\begin{pmatrix} x+1 & 3x^2-4 \\ x^2-2 & 2x^2-2x \end{pmatrix} \begin{pmatrix} 3x^2+2x \\ -x-4 \end{pmatrix} + (-5x-10, 12x^2+3)$$

\uparrow
Public secret noise

$$= (-7x^2+6, 4x)$$

\uparrow
Public (MLWE)

From lattice encryption to Kyber

- Use polynomial rings rather than integers. Ring-LWE, then even Module-LWE (NTRU)
- Public Key can be built using a square Matrix

small coefficients

$$A_s + \tilde{e} = t \rightarrow P_k = (A, t)$$

From lattice encryption to Kyber

- Use polynomial rings rather than integers. Ring-LWE, then even Module-LWE (NTRU)
- Public Key can be built using a square Matrix
- **Secret and noise can come from the same distribution**

small coefficients

0 1 2 3 4 5 6 7 8 9

Kyber encryption

C

↓ ASCII to Binary

1000011

↓ Binary to Polynomial

$$m = 1x^6 + 0x^5 + 0x^4 + 0x^3 + 0x^2 + 1x^1 + 1x^0$$

$$\underbrace{m = x^6 + x + 1}_{\text{Plaintext}}$$

Kyber encryption

$$P_k = (A, t) \quad m = x^6 + x + 1 \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{Plaintext}$$

↓ Scale with large factor

$$\underbrace{m}_{\substack{\text{Scaled} \\ \text{Plaintext}}} = 1337 \quad m = 1337x^2 + 1337x + 1337$$

↓ Add small errors *

$$\begin{array}{l} v = t * + * + m \\ n = A * + * \end{array} \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{Ciphertext}$$

Kyber decryption

$$\left. \begin{array}{l} P_k = (A, t) \\ S_k = s \end{array} \right\} \begin{array}{l} v = t * + * + m \\ m = A * + * \end{array} \text{ Ciphertext}$$

Kyber decryption

$$\begin{array}{ll} P_k = (A, t) & v = t * + * + m \\ S_k = \Delta & u = A * + * \end{array} \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{Ciphertext}$$

↓ remove P_k

$$d = v - \Delta u = t * + * + m - \Delta (A * + *)$$

$$d = A_{\Delta} * + * * + * + m - A_{\Delta} * - \Delta *$$

$$d = * * + * + m - \Delta * \quad \text{since } A_{\Delta} * + * = t$$

Kyber decryption

$$\begin{array}{ll} P_k = (A, t) & v = t * + * + m \\ S_k = \Delta & m = A * + * \end{array} \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{Ciphertext}$$

↓ remove P_k

$$d = v - \Delta m = t * + * + m - \Delta (A * + *)$$

$$d = A_{\Delta} * + * * + * + m - A_{\Delta} * - \Delta *$$

$$d = * * + * + m - \Delta *$$
 since $A_{\Delta} * + * = t$

↓

$$d = \underbrace{m}_{\text{large}} + \underbrace{* * + * - \Delta *}_{\text{small}}$$

Kyber decryption

$$\begin{aligned} P_K &= (A, t) & v &= t * + * + m \\ S_K &= \Delta & m &= A * + * \end{aligned} \quad \left. \begin{array}{l} v = t * + * + m \\ m = A * + * \end{array} \right\} \text{Ciphertext}$$

↓ remove P_K

$$d = v - \Delta m = t * + * + m - \Delta (A * + *)$$

$$d = A_{\Delta} * + * * + * + m - A_{\Delta} * - \Delta *$$

$$d = * * + * + m - \Delta *$$
 since $A_{\Delta} * + * = t$

↓

$$d = \underbrace{m}_{\text{large}} + \underbrace{* * + * - \Delta *}_{\text{small}} = 1337x^6 + 1337x + 1337$$

↑ rounding

Kyber decryption

C

↑ ASCII

1000011

↑ Binary

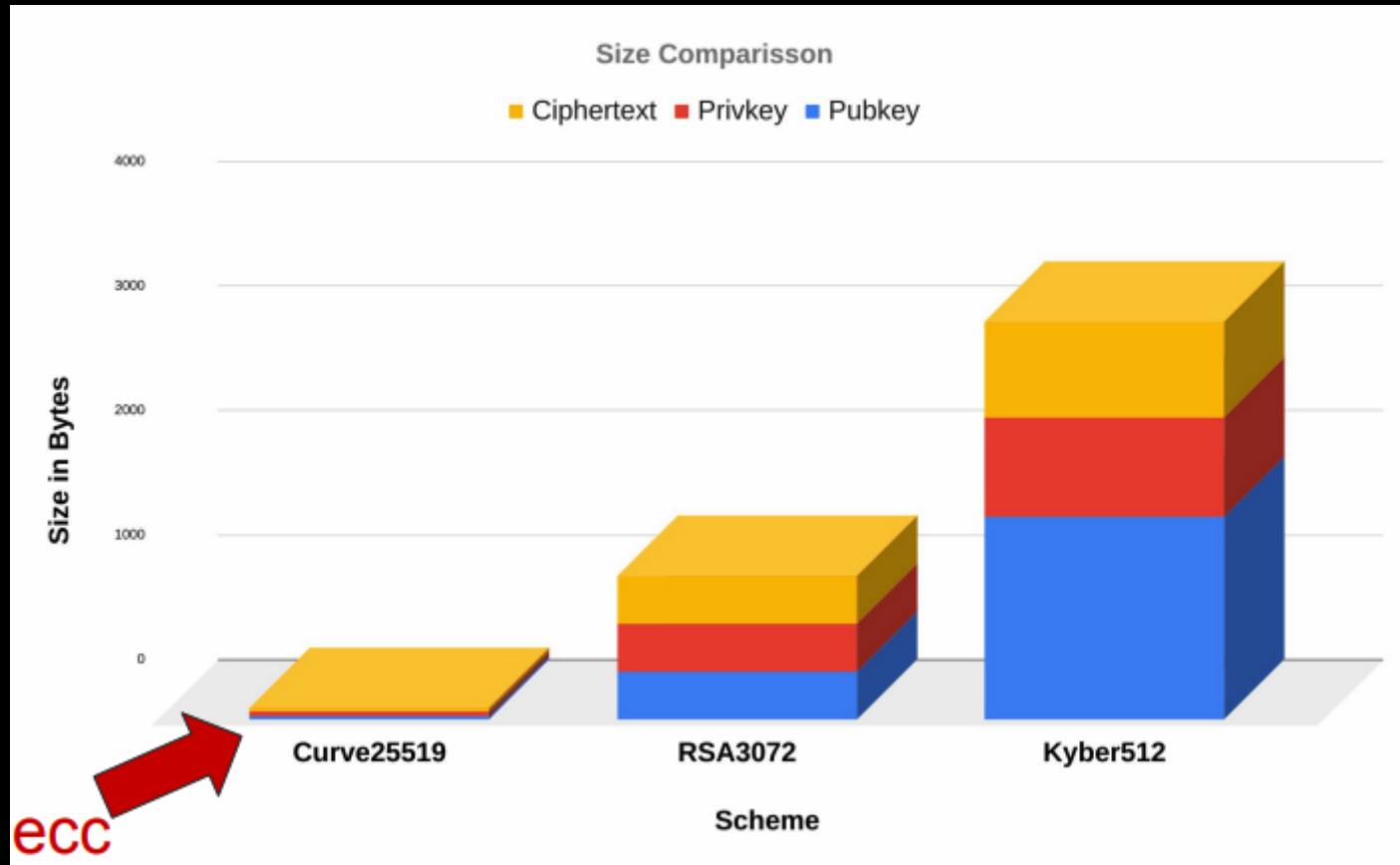
$$m = 1x^6 + 0x^5 + 0x^4 + 0x^3 + 0x^2 + 1x^1 + 1x^0$$

↑ Scale down

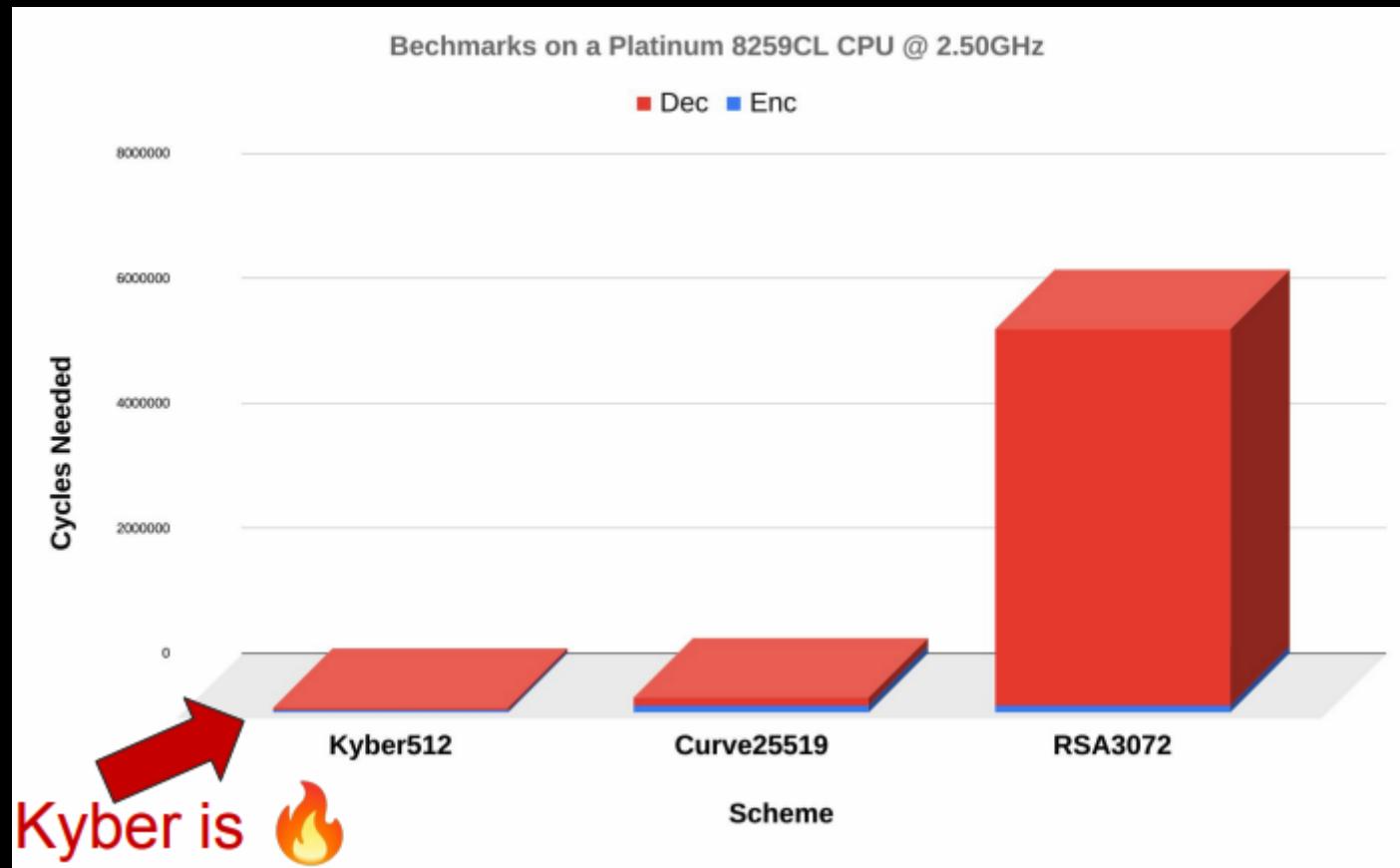
$$1337x^6 + 1337x + 1337$$

Kyber performances

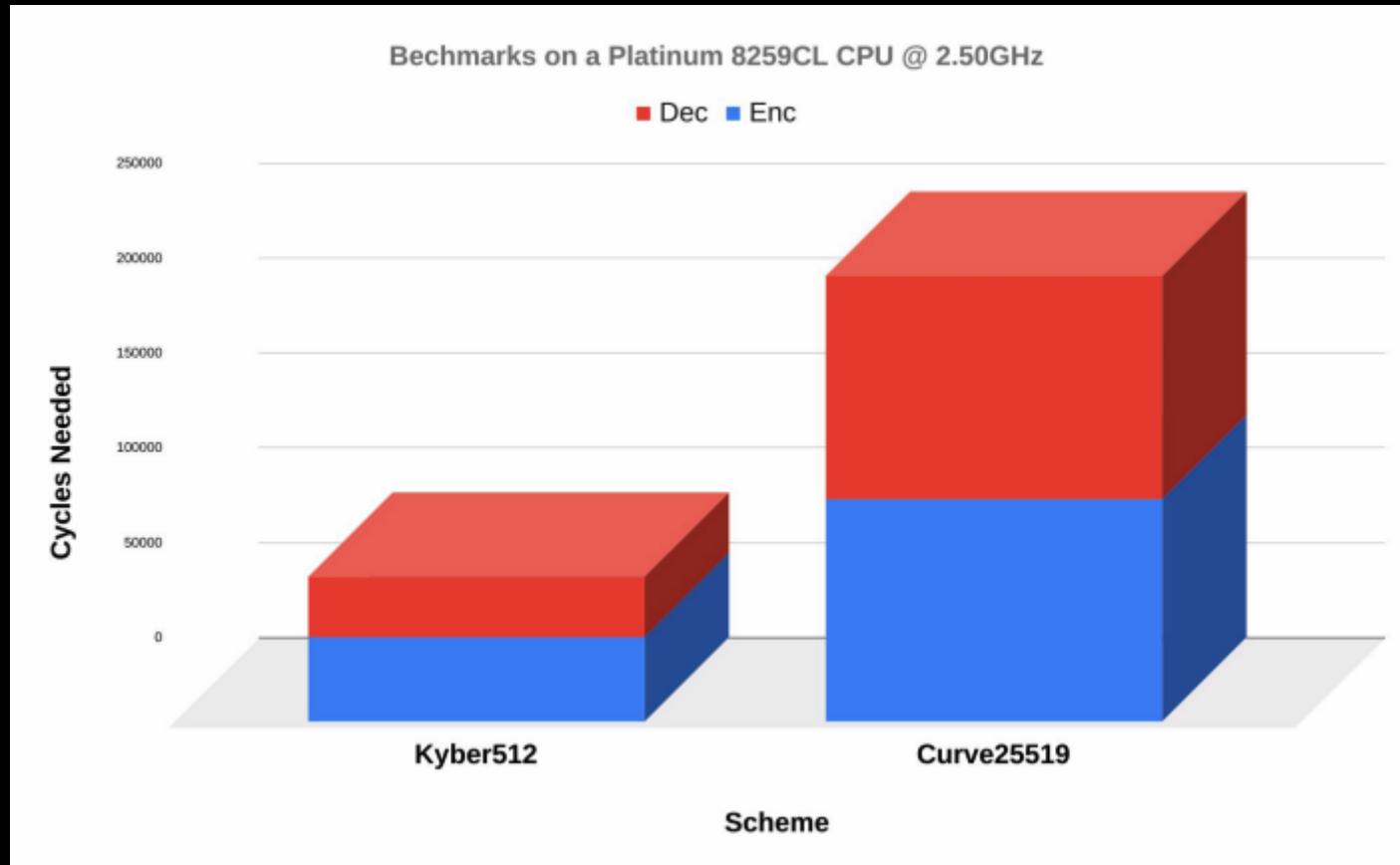
Kyber performances



Kyber performances



Kyber performances



Conclusion

Conclusion

Quantum computing

- Is a real threat for Cryptography and digital industry
- Is complex
 - Error management
 - Cold environment

Conclusion

Quantum computing

- Is a real threat for Cryptography and digital industry
- Is complex
 - Error management
 - Cold environment

Lattice based protocol

- Are secure according to CVP and SVP
- Are just polynomial manipulation
- Provide efficient cryptography schemes
- Are being standardized anyway

Resources

RSA

RSA nist standardization

Quantum Computing

- IBM Composer Guide
- Blog (IBM) on factorizing 15 with Shor in 2021
- Are we doomed with Shor ?



Resources

Post quantum Cryptography

- Article de Stéphane Bortzmeyer sur l'annonce su NIST
- NIST PQC standardization page
- McKinsey - when-and-how-to-prepare-for-post-quantum-cryptography

Resources

Lattice based encryption explained

- [Video on lattice based crypto](#)
- [Best video on Kyber](#)
- [Learning with Error explanation video](#)
- [Dilithium](#)
- [Kyber source code](#)
- [Réseaux euclidiens](#)
- [Course on SIS and LWE](#)
- [Course on ring LWE](#)
- [Kyber inspiration video](#)

Learning with errors (LWE)

| Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. pages 84–93, 2005.

Thank you !

Tweet me at

 -> @malvaultw

Mail me at

 -> willy@sogilis.com

Get slides at

 ->
<https://github.com/vlamy/talks>