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

ABYLSEN

Microsoft

salesforce

CGI

OVHcloud

« Flocon »

AVISTO

diabeloop

LIGHT UP
YOUR FUTURE_
HARDIS
GROUP

KLSGROUP
La French Logistique

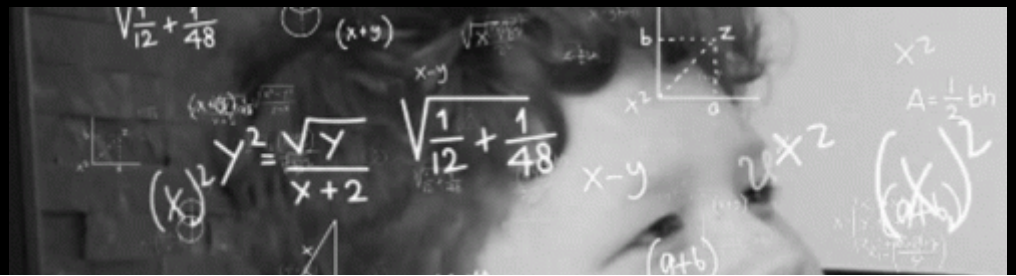
kelkoogroup

MOODY'S

It all begins with

My Approach

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



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 = Public Key

Sk = Secret Key

What is a public key cryptography scheme ?

Pk,Sk = Keygen()

Encryption

c = Enc(Pk,m)

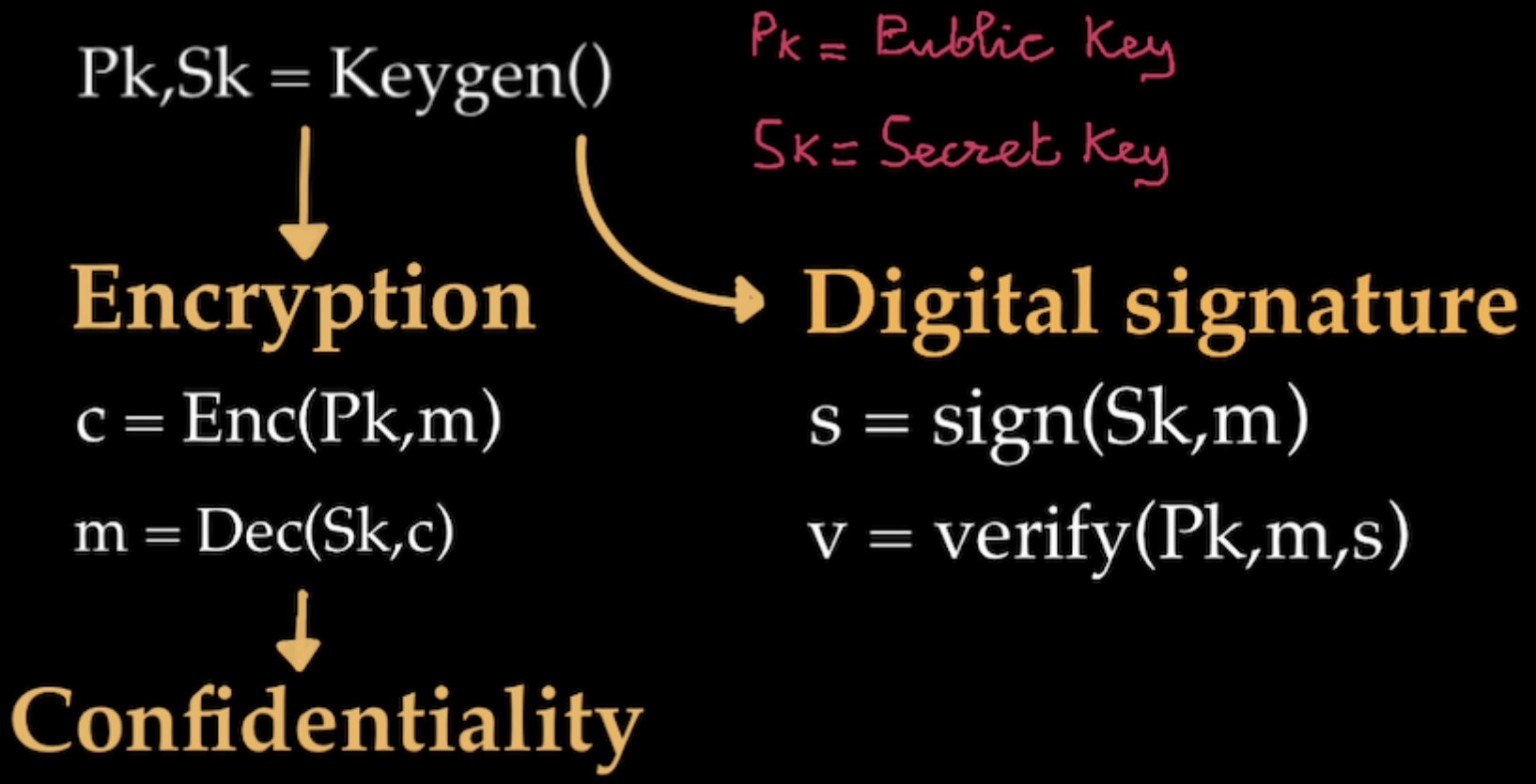
m = Dec(Sk,c)

Confidentiality

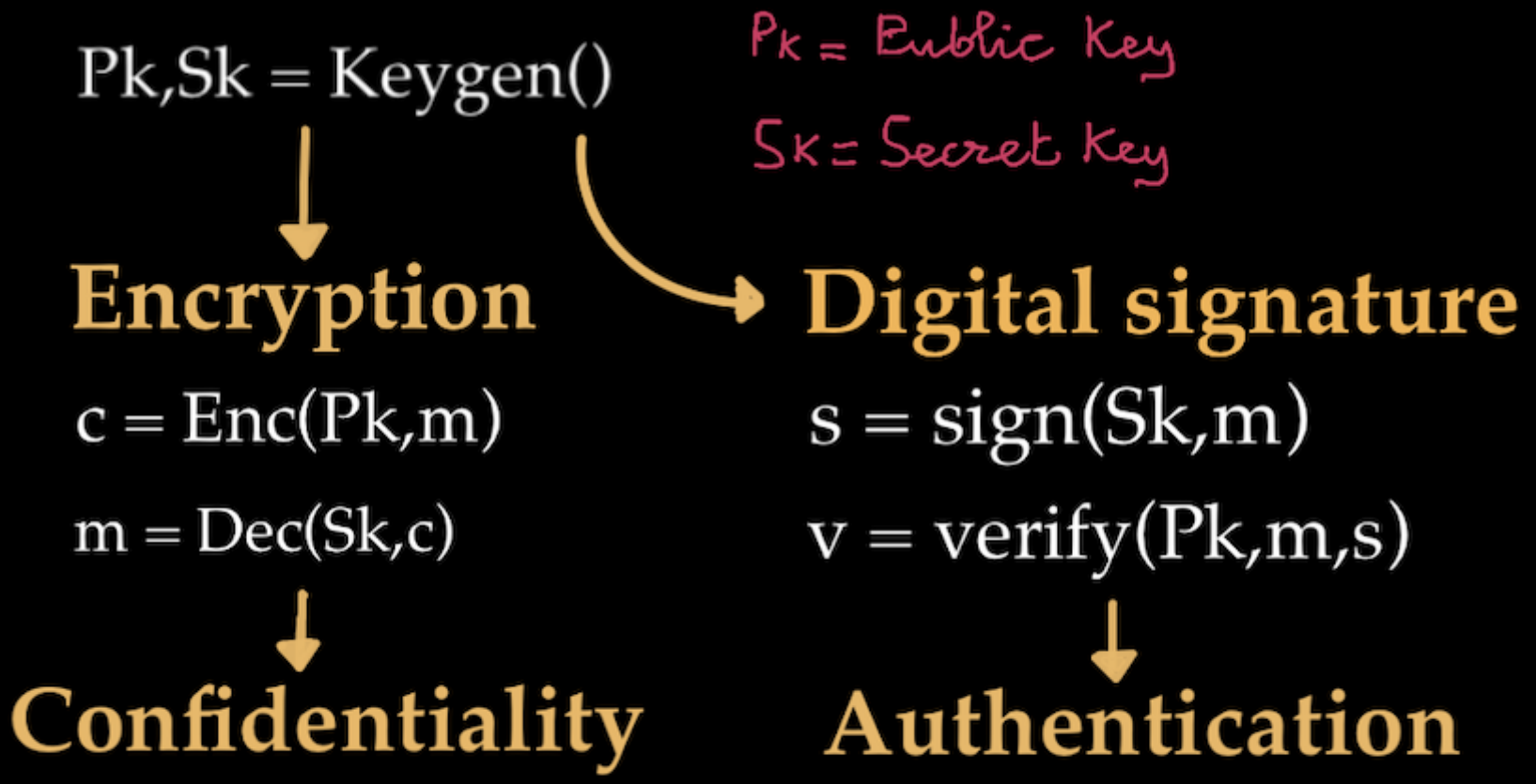
Pk = Public Key

Sk = Secret Key

What is a public key cryptography scheme ?



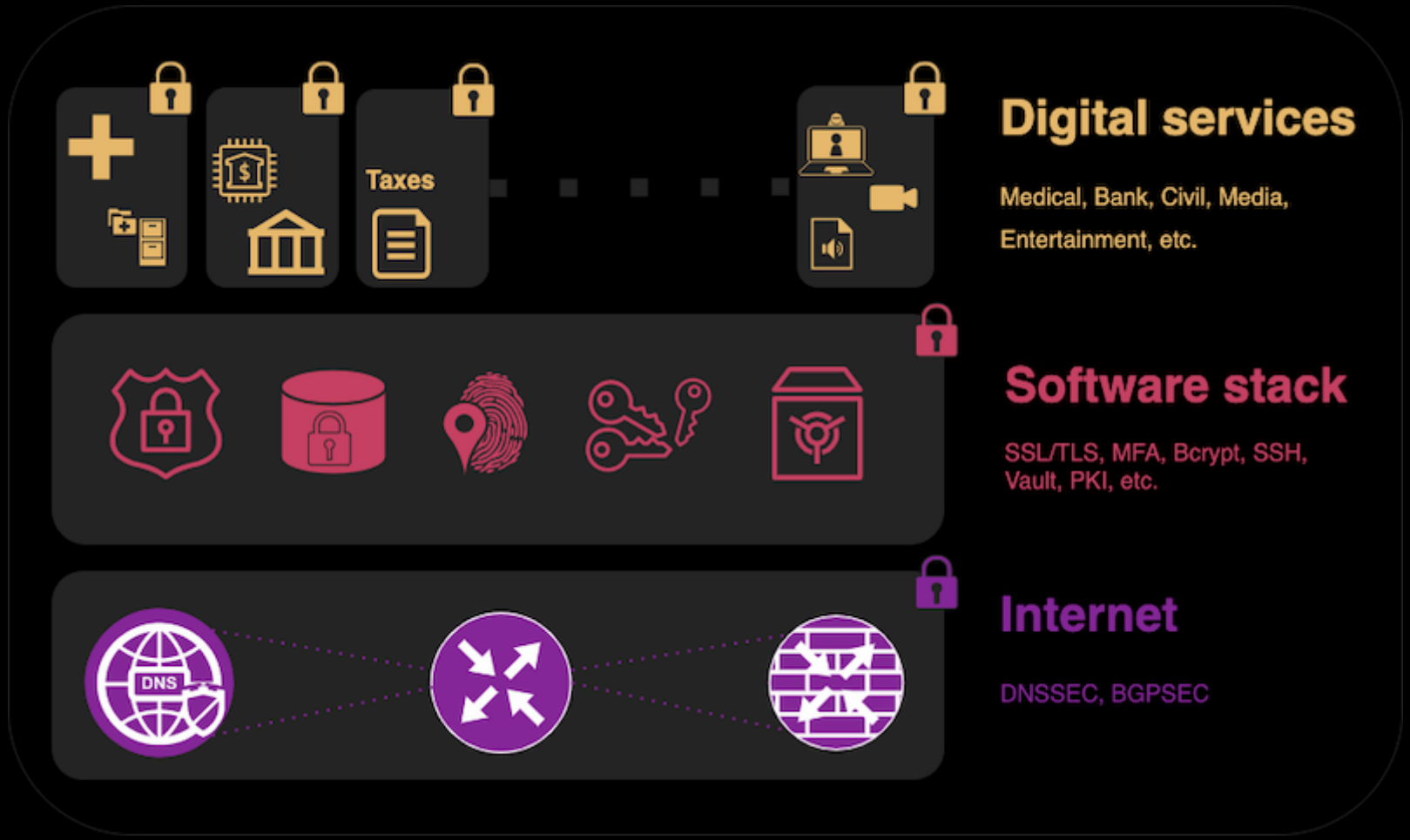
What is a public key cryptography scheme ?



What's the point ?

What's the point ?

What's the point ?



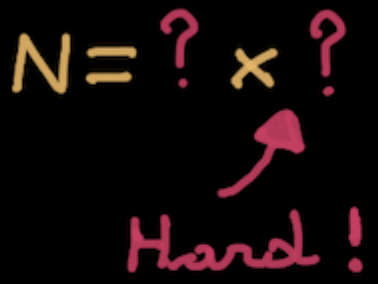
How to implement public key cryptography ?

> 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 ?

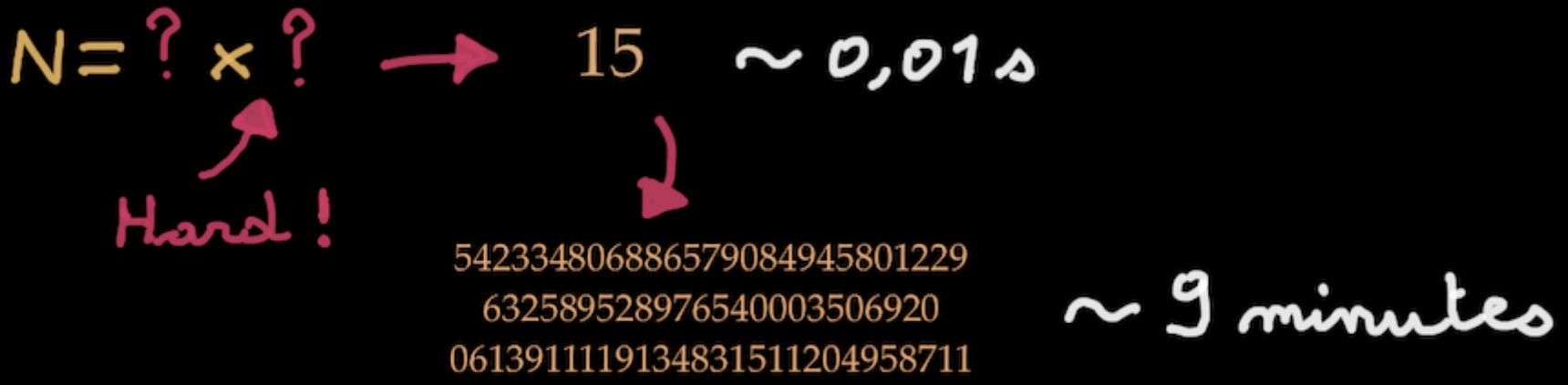
> Ronald L. Rivest, Adi Shamir, Leonard M. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM 21(2): 120-126 (1978)

Prime number factorization

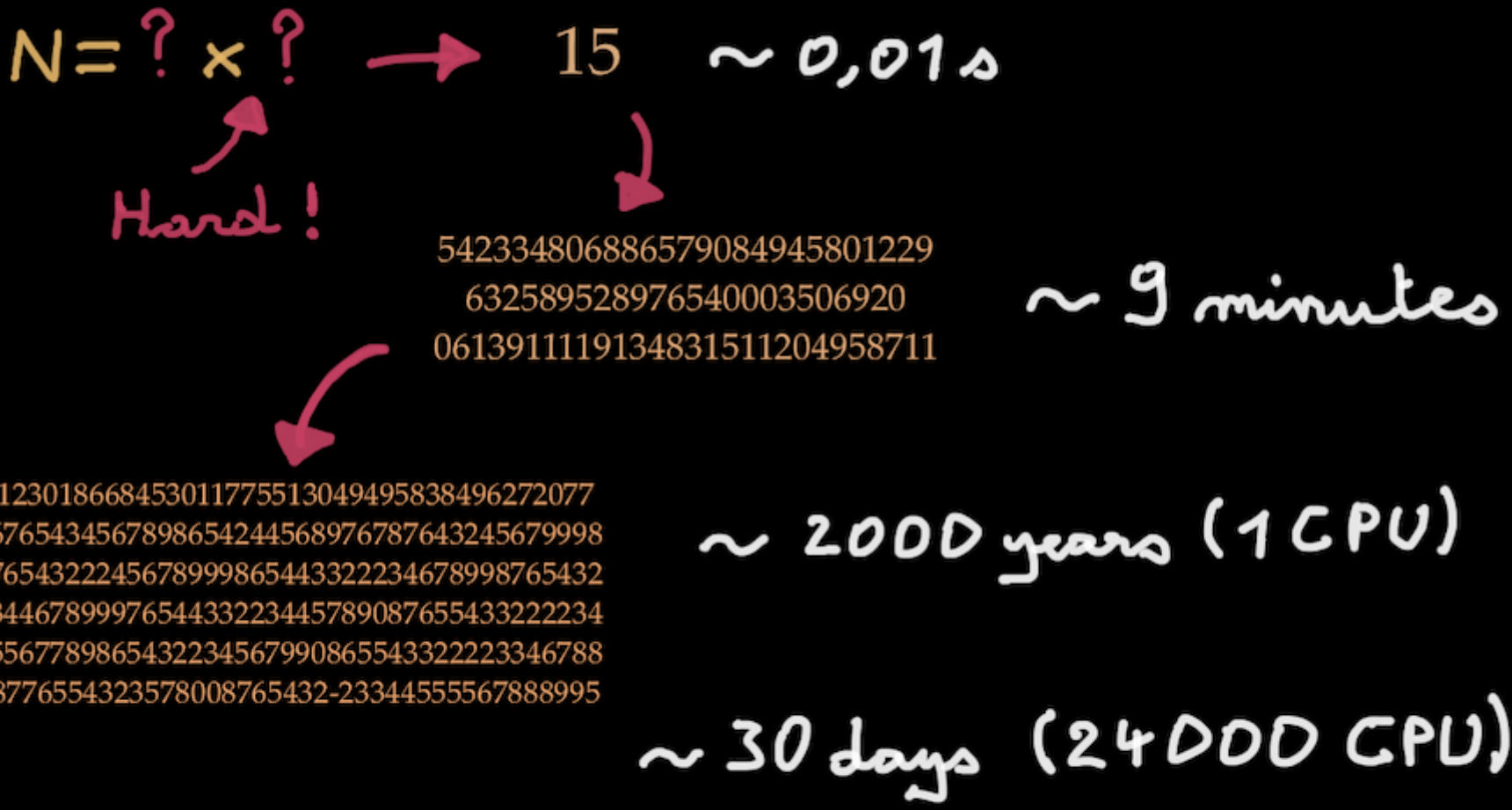


Prime number factorization

Prime number factorization



Prime number factorization



RSA cryptosystem (simplified)

KeyGen()

```
n = p * q

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

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

Enc(pk, m)

```
\(c = m^{Pk}\ mod\ n\)
```

Dec(sk, c)

```
\(m = c^{sk}\ mod\ n\)
```

RSA cryptosystem (simplified)

KeyGen()

$$n = p * q$$

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

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

Enc(pk, m)

$$c = m^{Pk} \mod n$$

Dec(sk, c)

$$m = c^{sk} \mod n$$

Sign(sk, m)

$$s = hash(m)^{sk} \mod n$$

Verify(pk, m)

$$h = s^{Pk} \mod n$$

OK if $hash(m)=h$

RSA cryptosystem (simplified)

KeyGen()

$$n = p * q$$

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

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

Enc(pk, m)

$$c = m^{Pk} \mod n$$

Dec(sk, c)

$$m = c^{sk} \mod n$$

Sign(sk, m)

$$s = hash(m)^{sk} \mod n$$

Verify(pk, m)

$$h = s^{Pk} \mod n$$

OK if $hash(m)=h$

RSA is working great !

The RSA problem/challenge

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

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 `Sk`

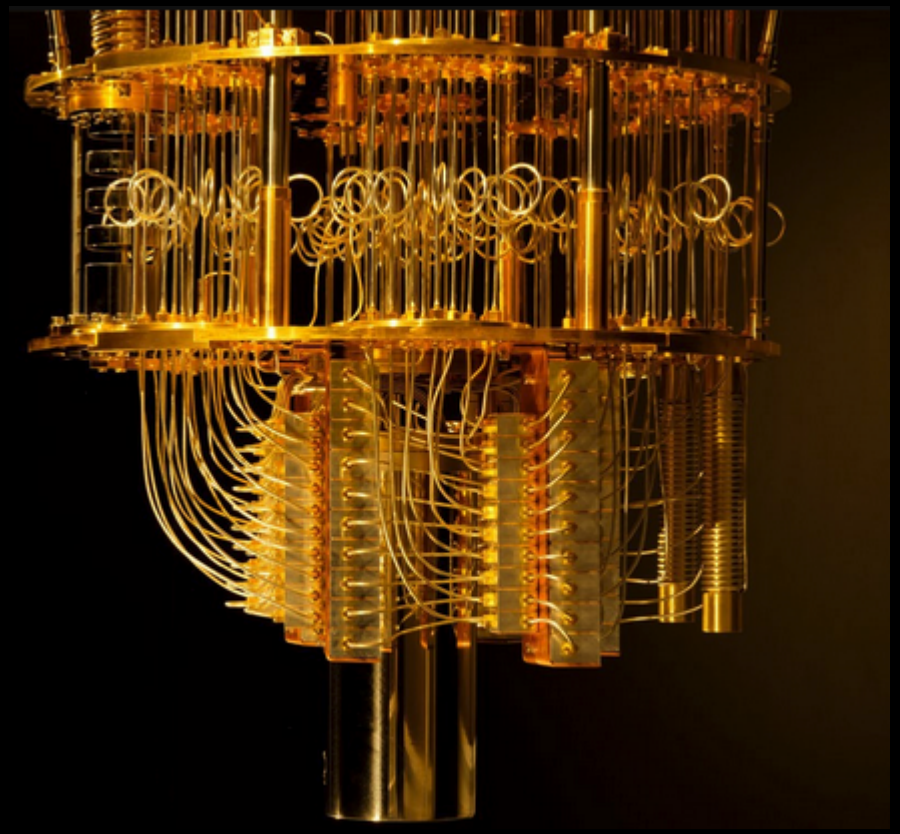
It takes around 1000 core-years to factorize RSA-768 (record), with a classical computer.

> 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)

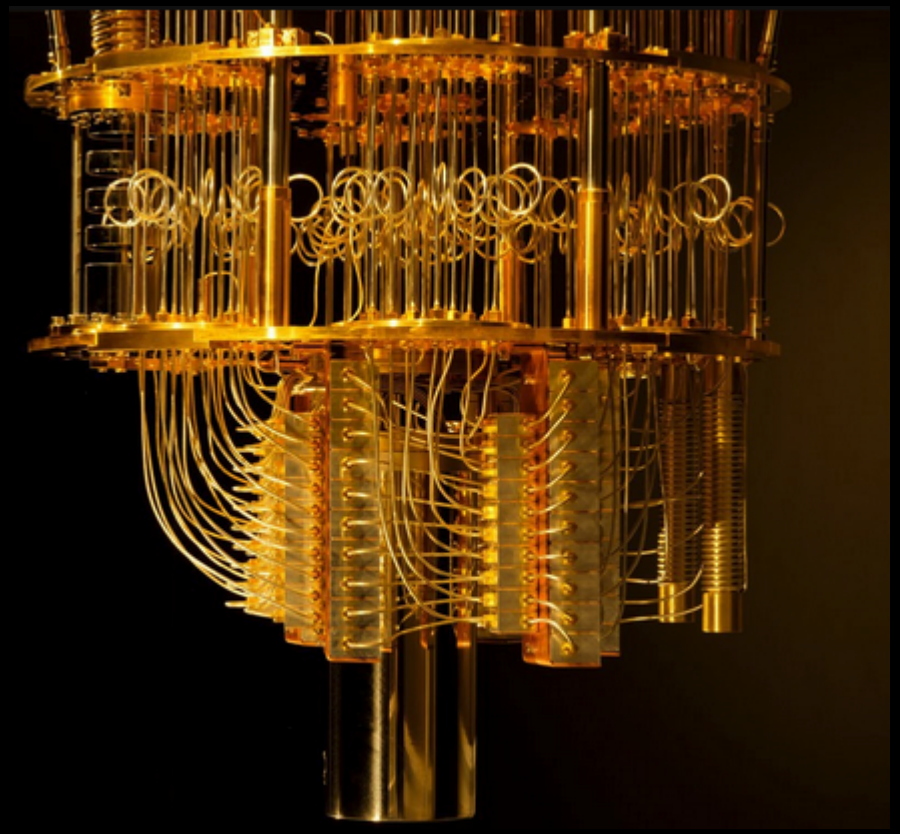
NIST standardization

But... this stands only for classical computers

Quantum computers is threatening !



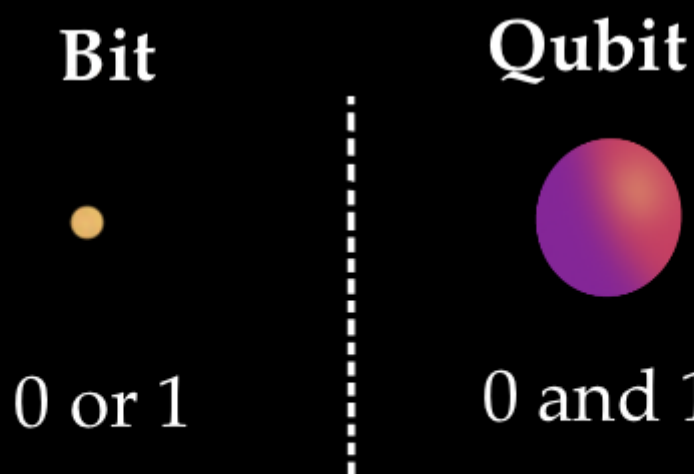
Quantum computers is threatening !



Qubit

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.

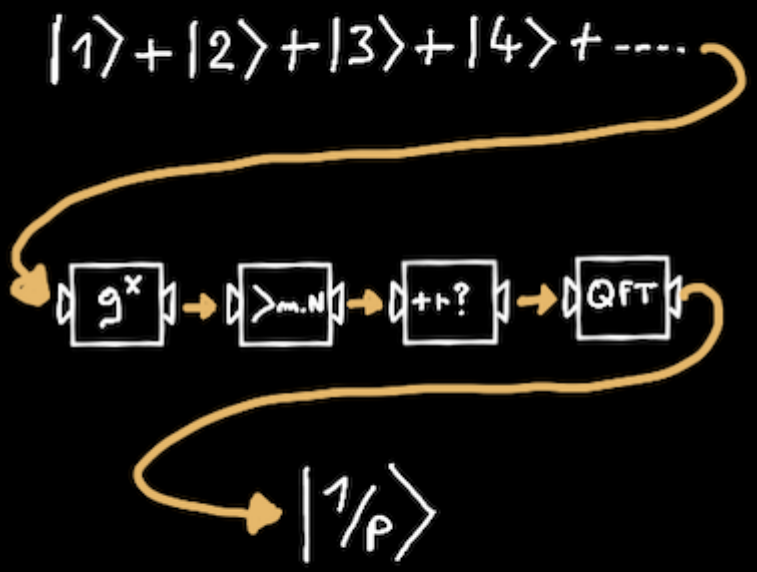
Strong at resolving high combinatory Problems:

Pharma, Nuclear, Meteo

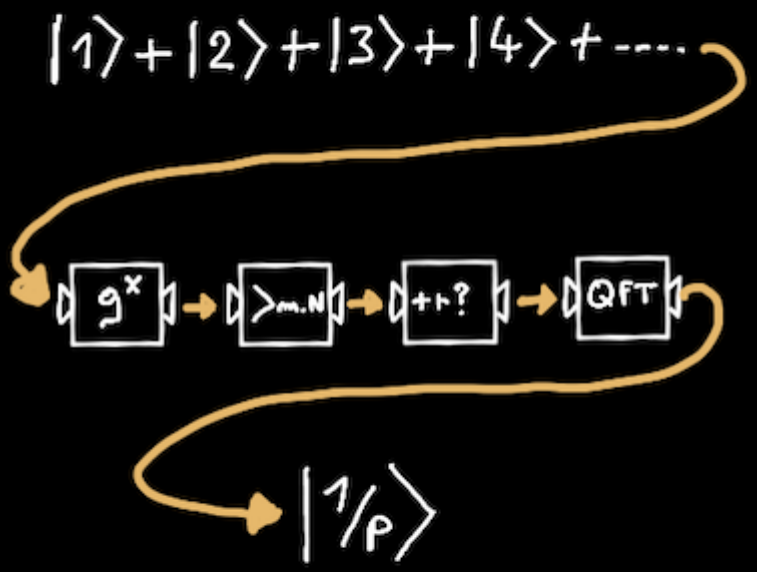
Articficial intelligence

Breaking RSA !

The Famous Shor algorithm

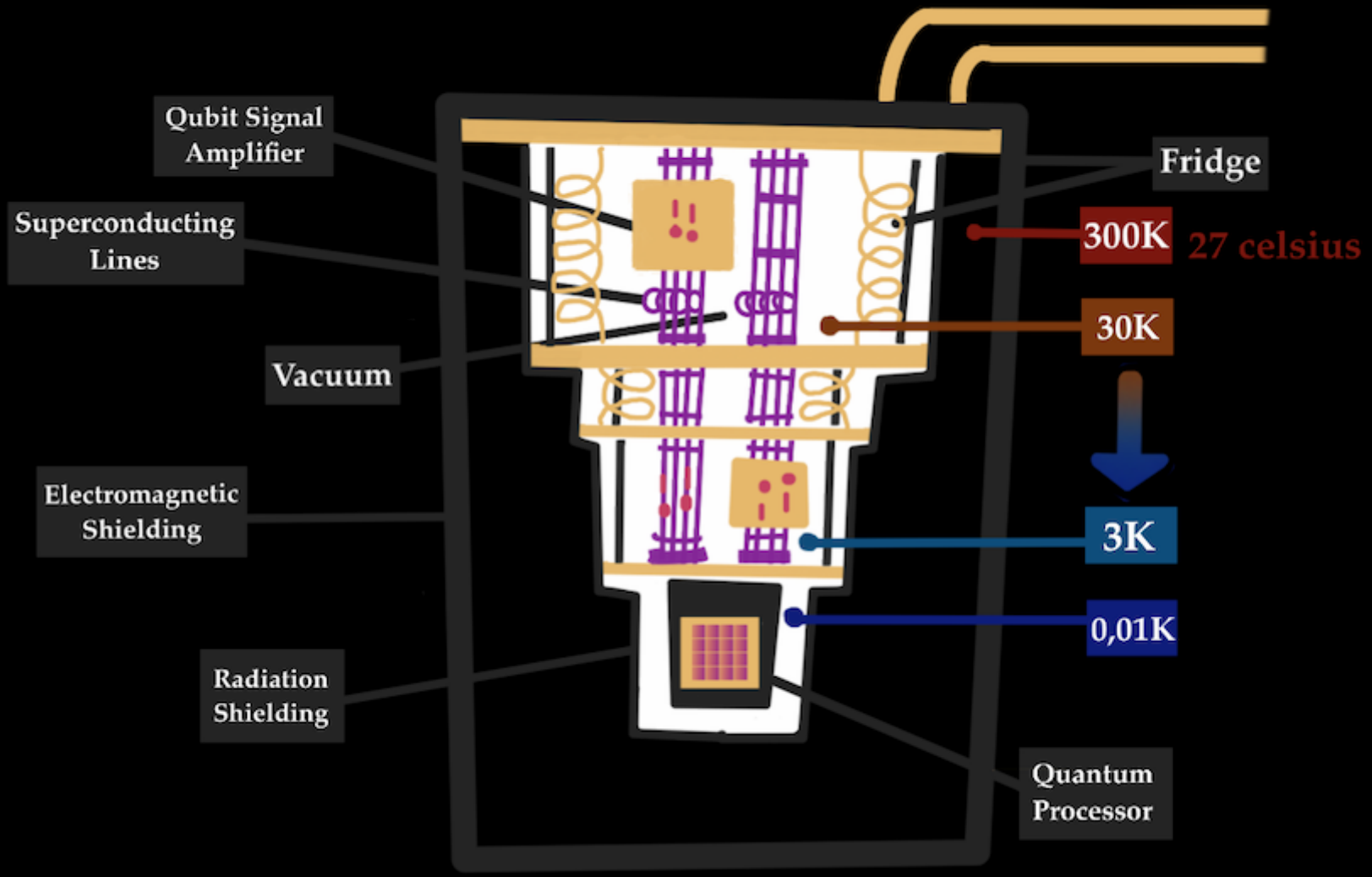


The Famous Shor algorithm

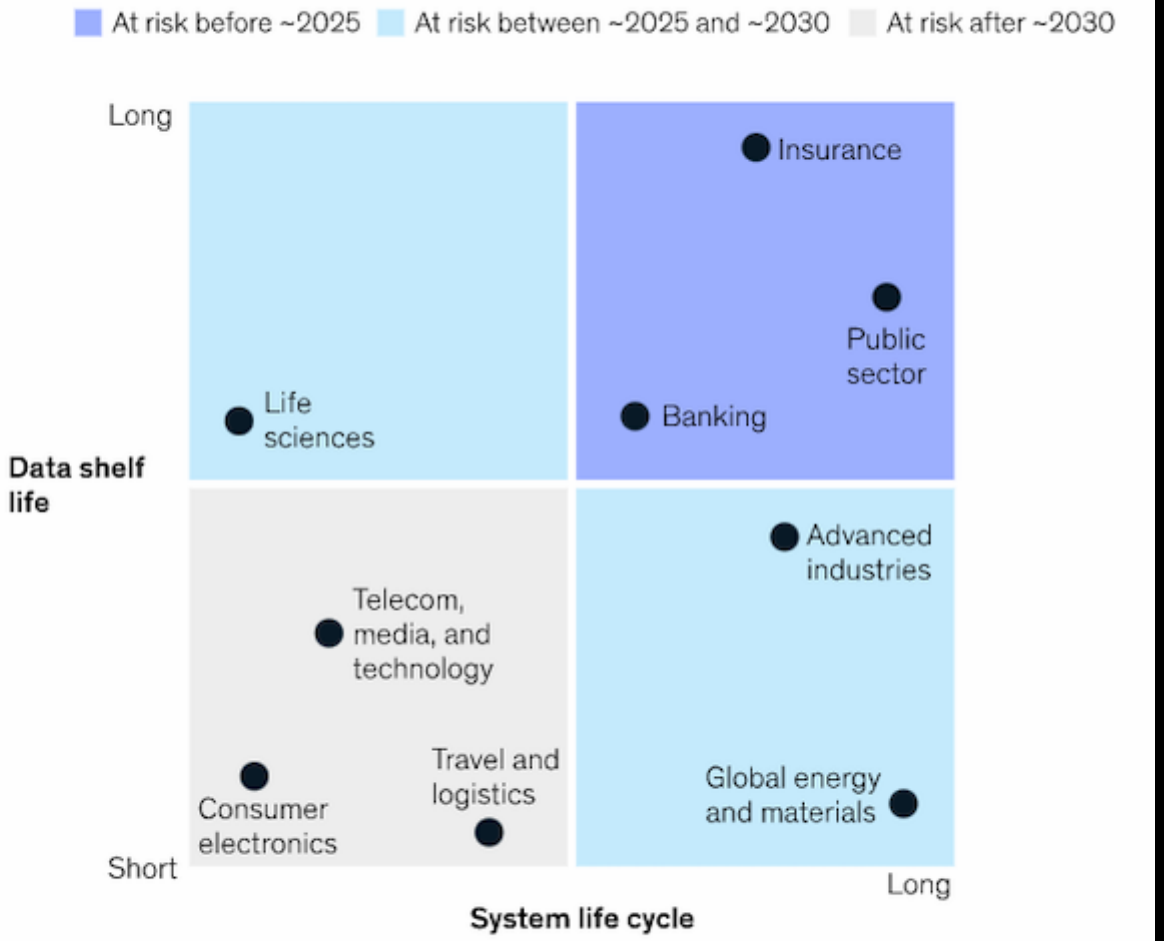


Should we be afraid of the monster ?

Should we be afraid of the monster ?


Qubit Signal Amplifier
Superconducting Lines
Vacuum
Electromagnetic Shielding
Radiation Shielding
Fridge
300K
27 celsius
30K
3K
0,01K
Quantum Processor

Risk of quantum-powered attack by industry



McKinsey & Company

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

Selected protocols

Public Key encryption and key-establishment

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

Digital signature

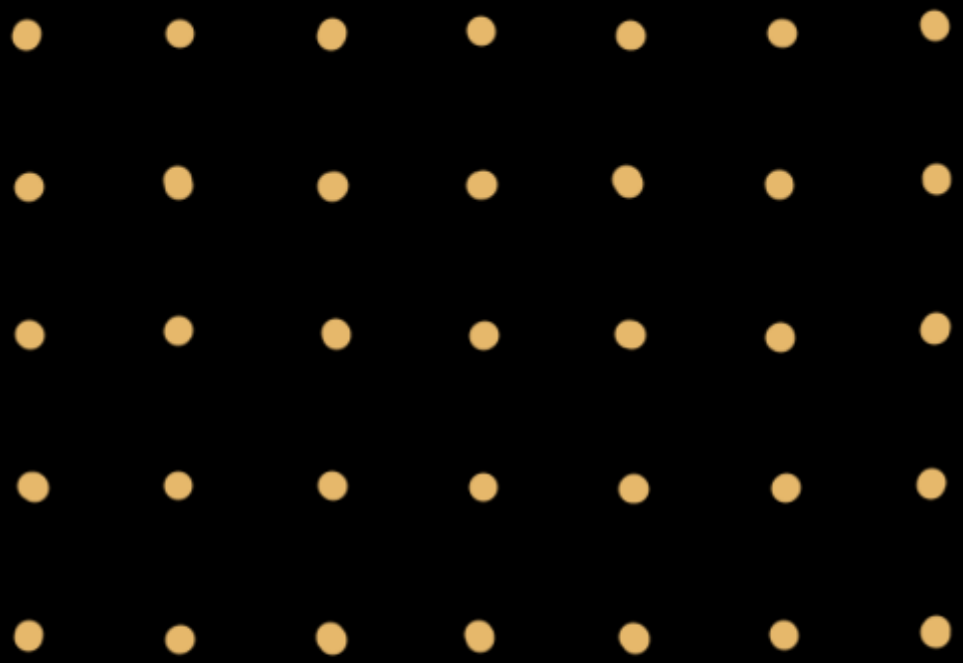
Dilithium: a strongly EUF-CMA-secure digital signature algorithm

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

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

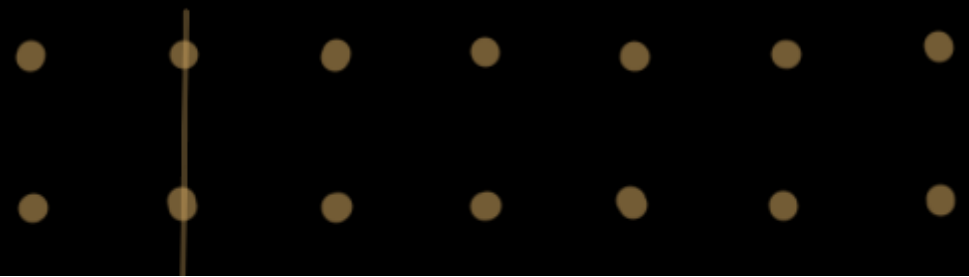
Introduction to lattice

- 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 \({b_1, b_2, ...b_N}\) 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 \({b_1, b_2, ...b_N}\) of \(\mathbb{R}^N\)
- \(L= \left\{ \sum a_{i} {b}_{i}\ :\ a_{i}\in \mathbb {Z} \right\} \)

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 $\{b_1, b_2, ...b_N\}$ of $\mathbb{R}^N$
- $L= \left\{ \sum a_{i} {b}_{i}\ :\ a_{i}\in \mathbb {Z} \right\}$

Fundamental hard problem

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

Learning With Error (LWE)

> Oded Regev, On lattices, learning with errors, random linear

Encrypt with Lattice

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

Encrypt with Lattice

- Good basis (b_1, b_2) is the private key
- Bad basis (b_1', b_2') 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 (b_1, b_2) is the private key
- Bad basis (b_1', b_2') 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

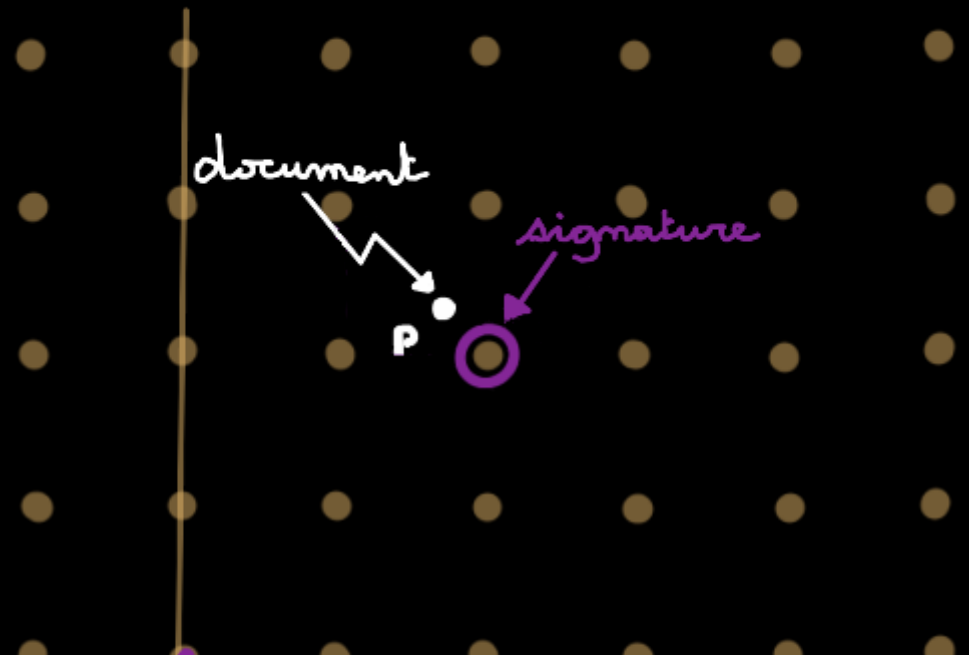
- Good basis (b_1, b_2) is the private key
- Bad basis (b_1', b_2') is the public key
- Encode bit with coordinates of a lattice point (according to public key, by instance)

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

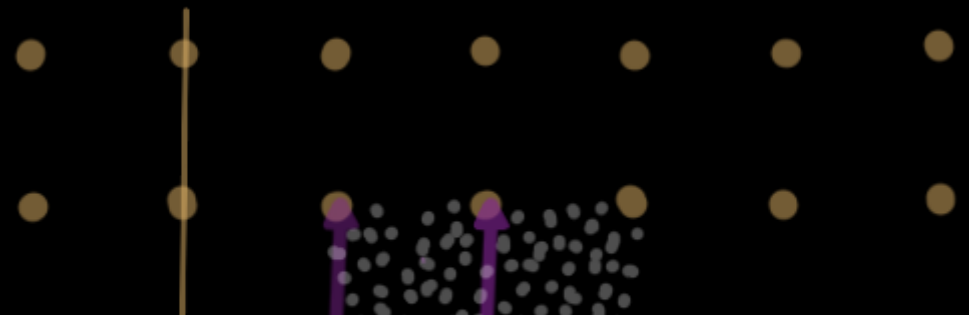


Lattice digital signature

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

Disadvantages

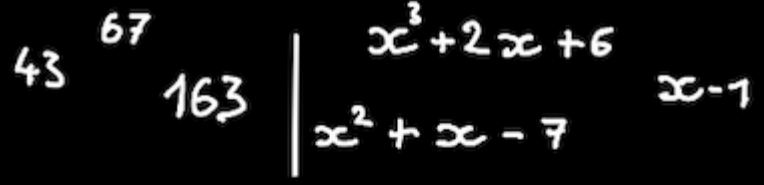
- Informations about private key can leak with a great number of signatures.



From lattice encryption to Kyber

From lattice encryption to Kyber

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



From lattice encryption to Kyber

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

$43 \quad 67 \quad 163 \;\Big|\; x^3+2x+6 \quad x-1$

x^2+x-7

We can add and multiple

$43 \times 67 = 2881 \;\Big|\; (x^2+x-7)(x-1) = x^3-8x+7$

From lattice encryption to Kyber

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

$43 \quad 67 \quad 163 \;\Big|\; x^3+2x+6 \quad x-1$

x^2+x-7

We can add and multiple

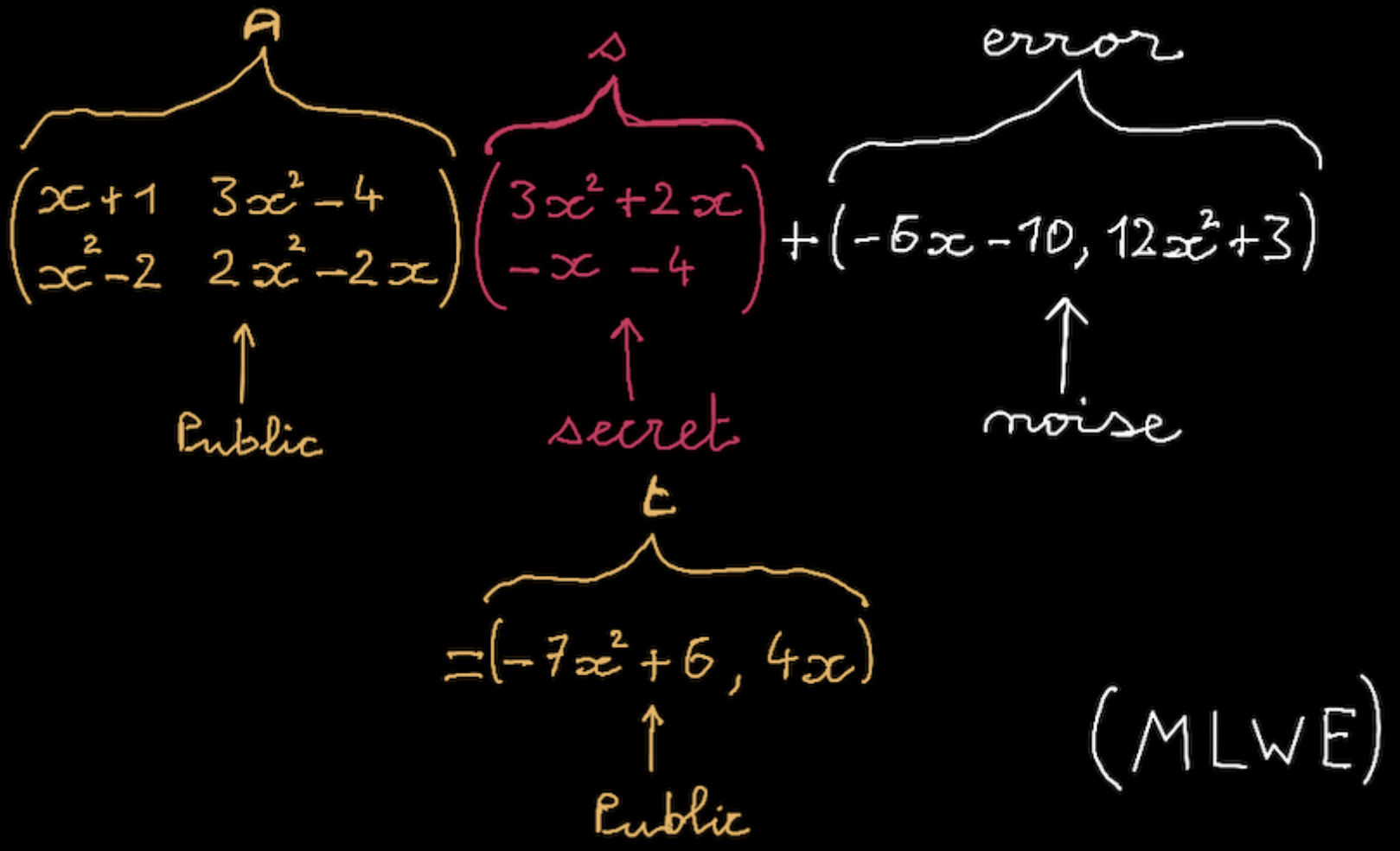
$43 \times 67 = 2881 \;\Big|\; (x^2+x-7)(x-1) = x^3-8x+7$

From lattice encryption to Kyber

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

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

Kyber matrix



From lattice encryption to Kyber

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

From lattice encryption to Kyber

- Use polynimial 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**

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

$P_K = (A, t)$

$m = x^6 + x + 1$ } Plaintext

↓ Scale with large factor

$m = 1337\,m = 1337x^2 + 1337x + 1337$

m: Scaled Plaintext

↓ Add small errors *

$v = t * + * + m$

$u = A * + *$ } Ciphertext

Kyber decryption

$pk = (A, t)$

$sk = s$

$v = t * r + e_2 + m$

$u = A * r + e_1$

} Ciphertext

Kyber decryption

$pk = (A, t)$

$sk = s$

$v = t * + * + m$

$u = A * + *$

} Ciphertext

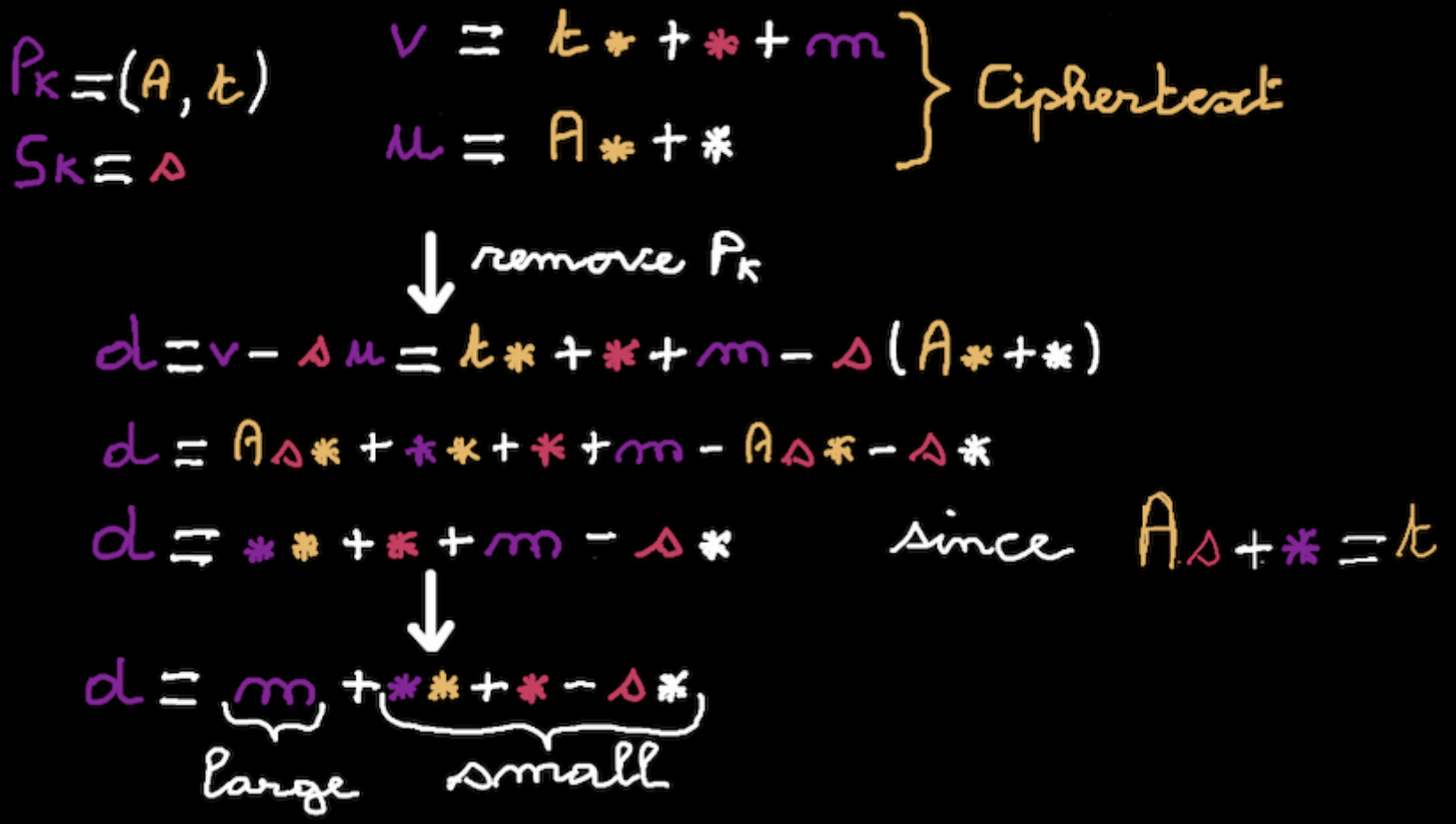
↓ remove pk

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

$d = A s * + * * + * + m - A s * - s *$

$d = * * + * + m - s *$ since $A s + * = t$

Kyber decryption



Kyber decryption

$pk = (A, t)$

$sk = s$

$v = t*r + *e + m$

$u = A*r + *e$

} Ciphertext

↓ remove pk

$d = v - su = t*r + *e + m - s(A*r + *e)$

$d = A*s*r + *e*r + *e + m - A*s*r - s*e$

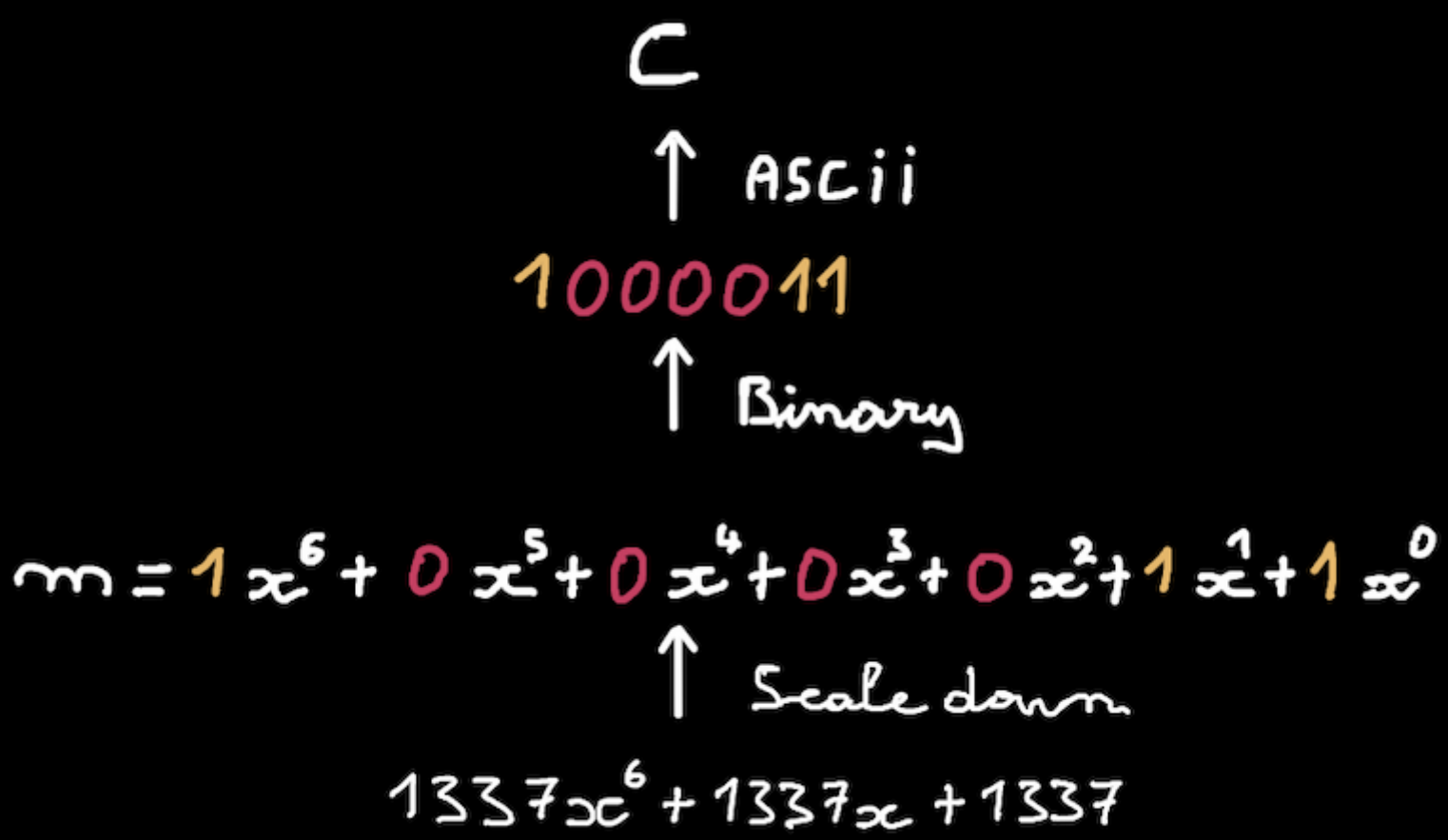
$d = *e*r + *e + m - s*e$ since $A*s + *e = t$

↓

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

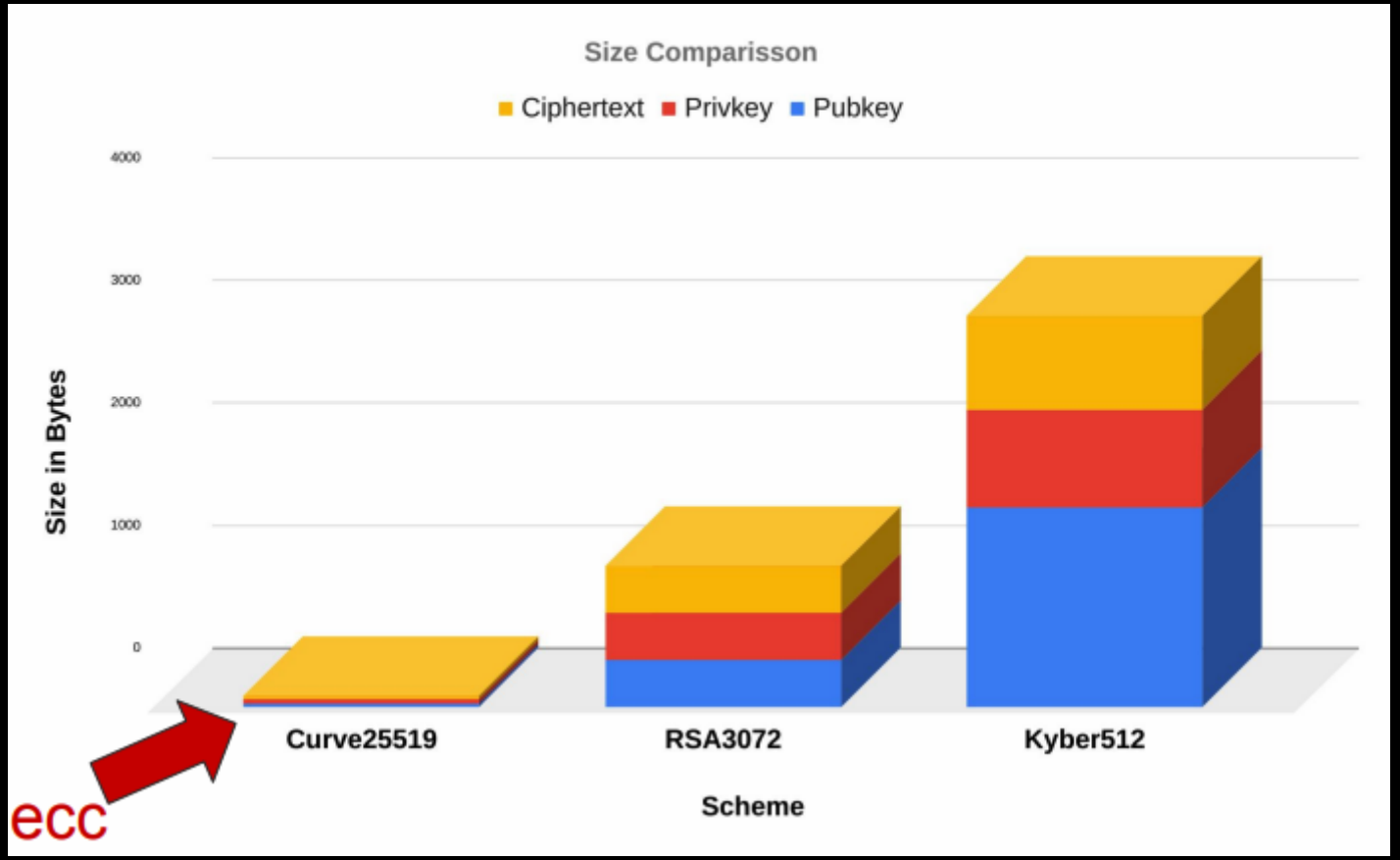
↑ rounding

Kyber decryption

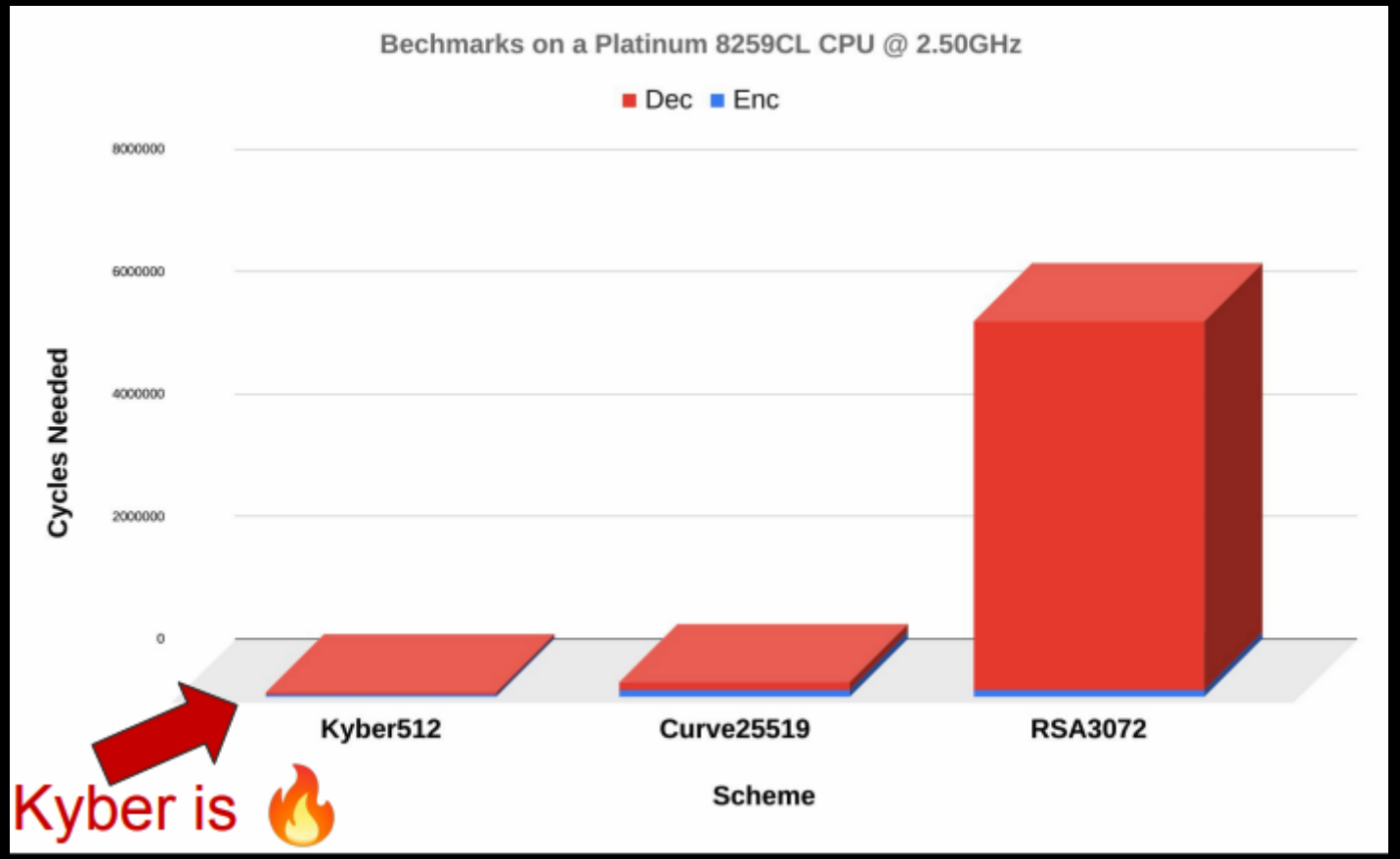


Kyber performances

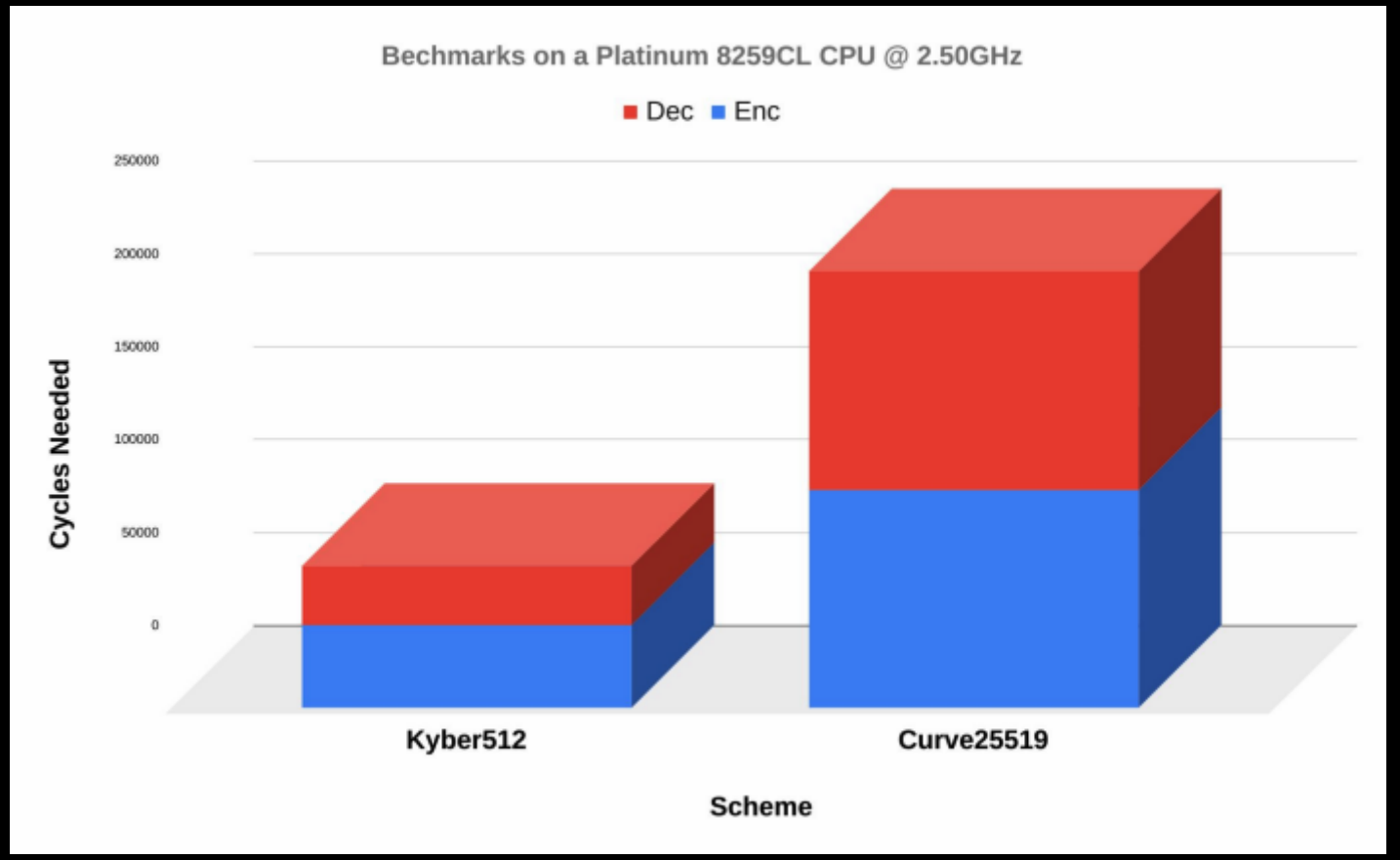
Kyber performances



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`