CRYPTOGRAPHY

PHONG NGUYEN

http://www.di.ens.fr/~pnguyen

INRIA
INVENTEURS DU MONDE NUMÉRIQUE

ENS
ÉCOLE NORMALE SUPÉRIEURE

CRYPTO EVERYWHERE

Cards

Games and DVDs

Phones and Keys

Passports

Internet

CRYPTO EVERYWHERE


Printers

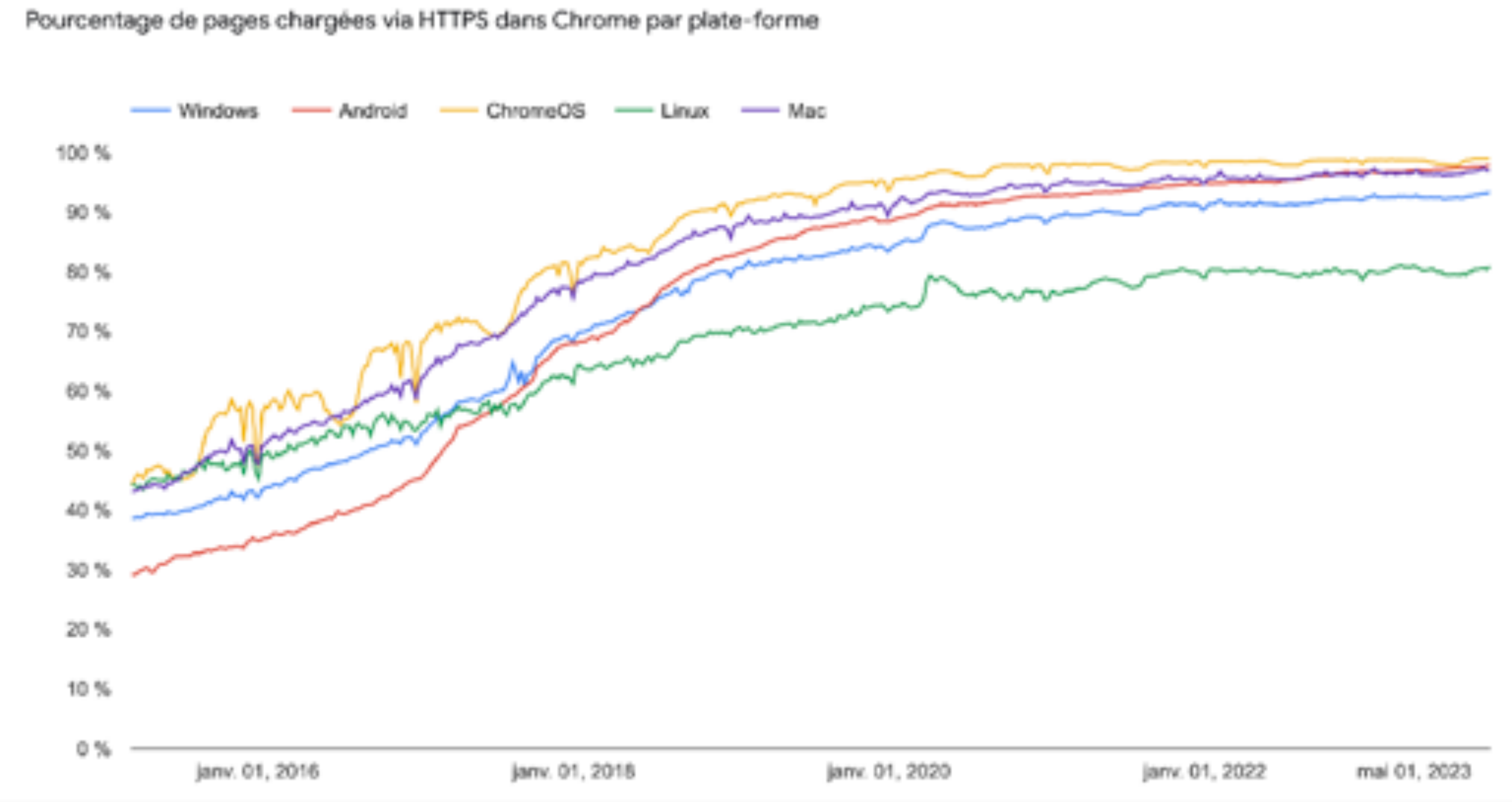
Cameras

Stamps

WHY CRYPTO?

- To defend against enemies

- More and more data
  - ➤ Which data?
  - ➤ Who knows your data?
  - ➤ How secure is data?
    95% of web traffic is encrypted.



- Security problems

SO WHAT?

Many security problems

Crypto helps, but does not solve everything

Trend: more and more crypto,
but many problems remain.

THE TWO WORLDS OF CRYPTOGRAPHY

- **Symmetric Cryptography**
  - ➤ Efficient and widely used
  - ➤ But management/storage of secret keys is tricky

- **Asymmetric (or Public-Key) Cryptography**
  - ➤ No need to share secret keys
  - ➤ But much less efficient and requires larger keys
  - ➤ Public keys must be certified

Uses algebra and number theory

IN PRACTICE

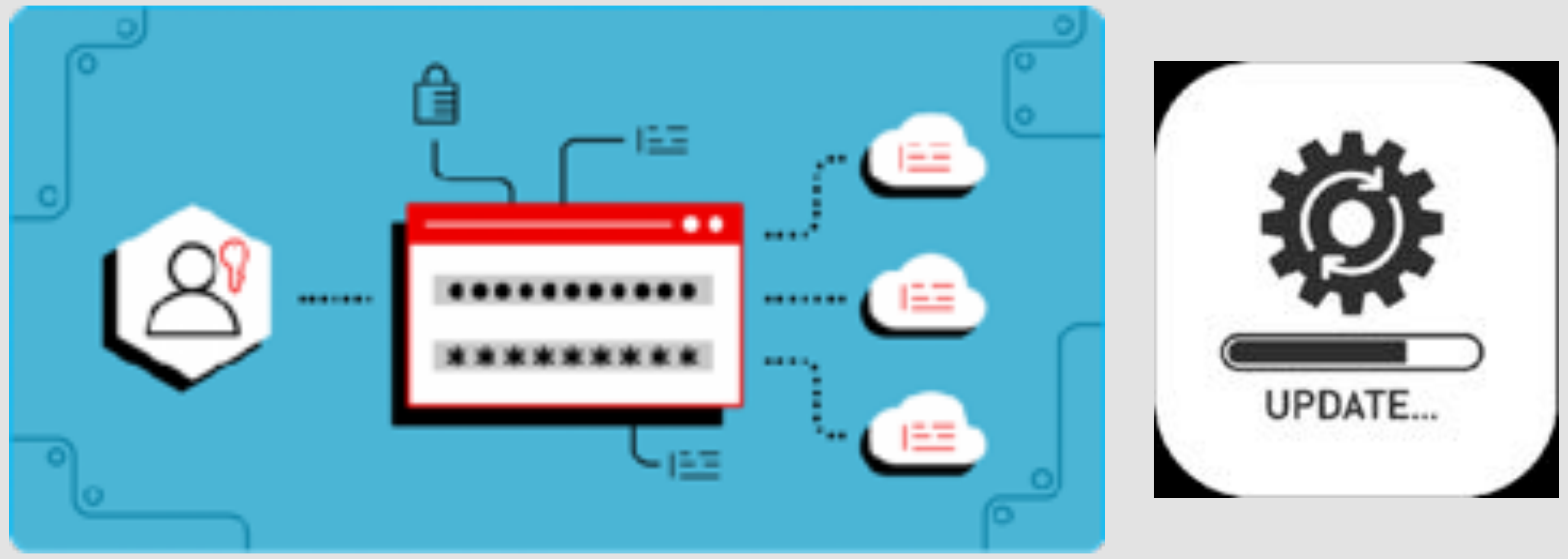
- Public-key cryptography is only used **when really necessary**.

  - Key exchange or asymmetric encryption

  SSL/TLS

  - Signatures, certification of public keys

  X.509 CERTIFICATES

TWO TYPES OF CRYPTOGRAPHY

- **Cheap Cryptography**
  - Goal: efficiency
  - Target: Software (8-bit, 32-bit, 64-bit) or Hardware (LFSR, circuits)
  - Drawback: specialized for an architecture

- **Mathematical Cryptography**
  - Goal: Functionalities
  - Rely on mathematical objects
  - Drawback: difficult to implement

CRYPTOGRAPHIC TRENDS AND CHALLENGES

Security

Efficiency

Functionalities

REQUIREMENTS

- You need to ask questions
- If there's something you don't understand, please raise your hands

- How much mathematics do you know?
    - Permutations
    - Modular arithmetic
    - Groups
    - Polynomials
    - Finite fields

SOFTWARE

- Install
  - Python3 with various libraries: hashlib
  - SageMath
  - Openssl

SUMMARY

- Symmetric Cryptography
  - Keyless cryptography: hash functions and pseudo-random number generators
  - Symmetric encryption: stream ciphers and block ciphers
- Public-Key Cryptography
  - Main algorithms: RSA and DL/ECC
  - Real-world attacks
- Advanced Public-Key Cryptography
  - Post-Quantum Cryptography (to be deployed soon)
  - Homomorphic encryption (ongoing standardization)
  - Zero-Knowledge Proofs

WHAT IS PKC?

PUBLIC-KEY CRYPTOGRAPHY

- Invented by Diffie and Hellman in 1976.

  - « We stand today on the brink of a revolution in cryptography. »
  - « however, it currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such a private courier or registered mail. (…) The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks. »

PUBLIC-KEY CRYPTOGRAPHY

- All the keys do not necessarily have to be secret: some keys can be public!
  - The encryption key becomes public: the decryption key remains secret.
  - The verification key for signatures is public: the signature-generation key remains secret.

ENCRYPTION

ENCRYPTION IN PRACTICE

- Every (video/game) DVD is encrypted.
- The boot codes and operating systems of every modern gaming console are encrypted.

- Most wireless networks are encrypted.

- Credit card numbers are encrypted.
- Mobile phone - Antenna communications are encrypted.
- 95% of web traffic is encrypted.

SYMMETRIC ENCRYPTION

- Encryption and decryption depend on the same (secret) key.

Plaintext m

Encryption

0100011100100101

Ciphertext

$c=E_k(m)$

- Decryption: $m=D_k(c)$

ASYMMETRIC ENCRYPTION [DH76]

- Encryption and decryption use two different keys: only the decryption key is secret. Both keys are related.

Encryption

0100011100100101

Plaintext m

Ciphertext

$c = E_k(m)$

- Decryption: $m = D_{k'}(c)$ where $k' \neq k$. Nobody should be able to compute k' from k.

EXAMPLE: HARD-DISK ENCRYPTION

- Symmetric or asymmetric encryption?
- Who knows the secret key?
- Where is the secret key?

EXAMPLE: MALWARE

- Encrypt all user files with a certain extension.
- Asks for a ransom to decrypt, maybe using bitcoins.

EXAMPLE: MALWARE

- If **gpcode** only used symmetric crypto, the cure would be easy: just retrieve all secret keys by reverse-engineering.
- But with public-key encryption, the program has no secret: where is the secret?

EXAMPLE: MALWARE

- A public key pk is embedded in the malware.
- The malware:
  - selects a random session key **k** for symmetric encryption.
  - **k**-encrypts target files, and destroy the original files.
  - stores $c=E_{pk}(\mathbf{k})$ and destroy **k**.
  - asks for a ransom to disclose **k** from c and recover the files.

EXAMPLE: MALWARE

- To decrypt, pay the ransom and email the challenge **c** to the malware author, who will email back **k**. Only the author knows the secret key.
- This idea appeared in [YoYu96] and was used in the 2000s in late versions of the **gpcode** ransomware.
- Note: this combination of symmetric and asymmetric encryption is well-known. It is called hybrid encryption.

AUTHENTICATION

DIGITAL SIGNATURES [DH76]

- Two different keys: a secret key k’ to sign messages, and a public key k to verify messages.

Message m

Sign

$s=S_{k'}(m)$

- Verification: check V(m,k,s).

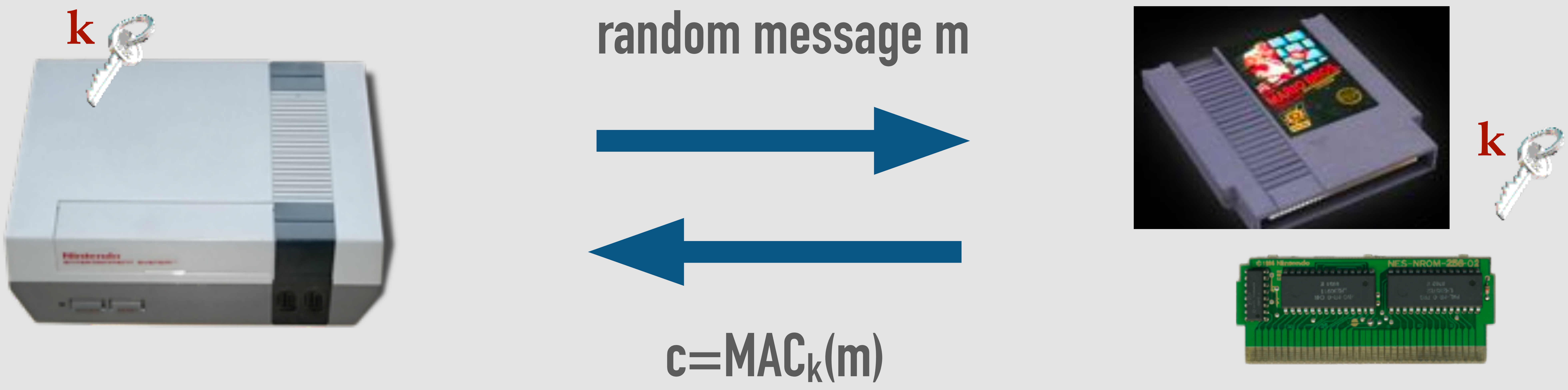
AUTHENTICATION OF VIDEO GAMES

- 1983: The Crash of the Video Game Industry

ATARI 2600

AUTHENTICATION OF VIDEO GAMES

- 1985: Nintendo Entertainment System (NES)



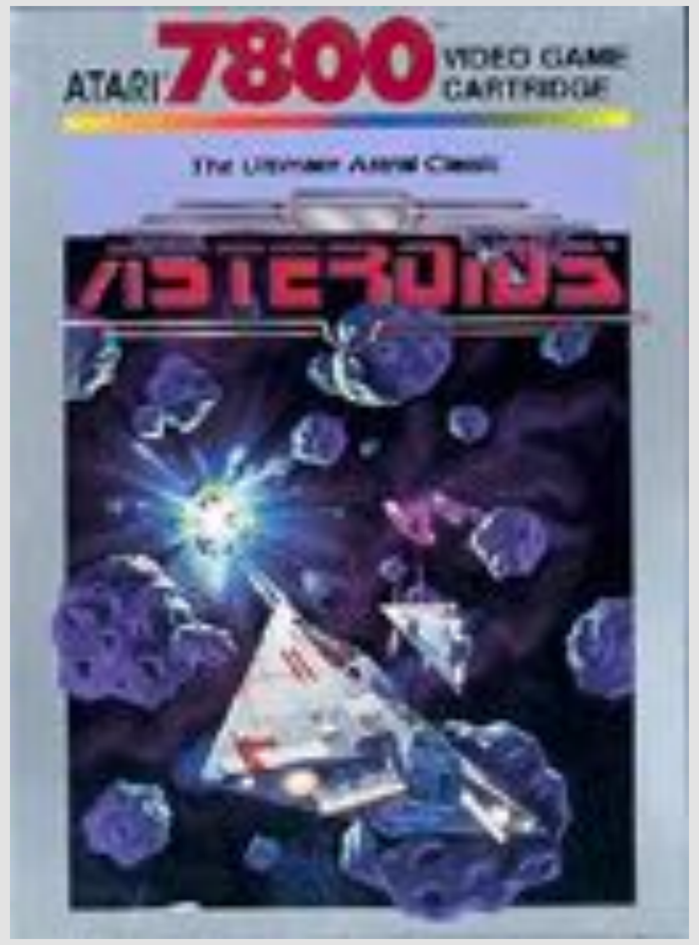
➤ Check that $c=MAC_k(m)$

AUTHENTICATION OF VIDEO GAMES

- 1986: Atari 7800

Program m

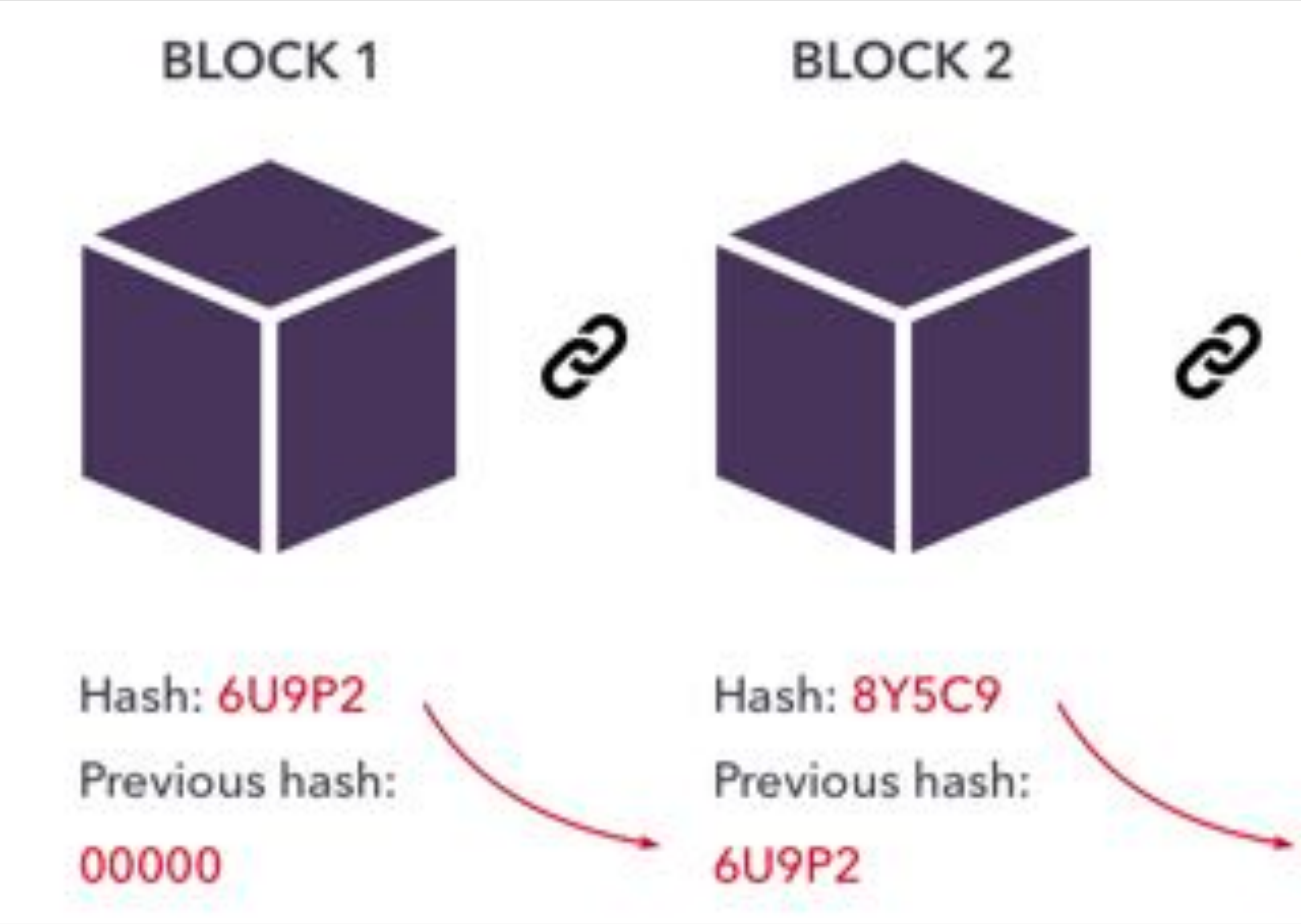
Digital Signature s



➤ Check that s is a valid signature of m, before running m.

EXAMPLE: BITCOIN

- The blockchain:
  - Blocks are chained together by hashing.
  - Each block contains the hash of the previous block.



- A bitcoin block contains statements of the form:
  - « My name is XYZ, and I certify that I give xxx bitcoins to the person ABC » together with a digital signature.

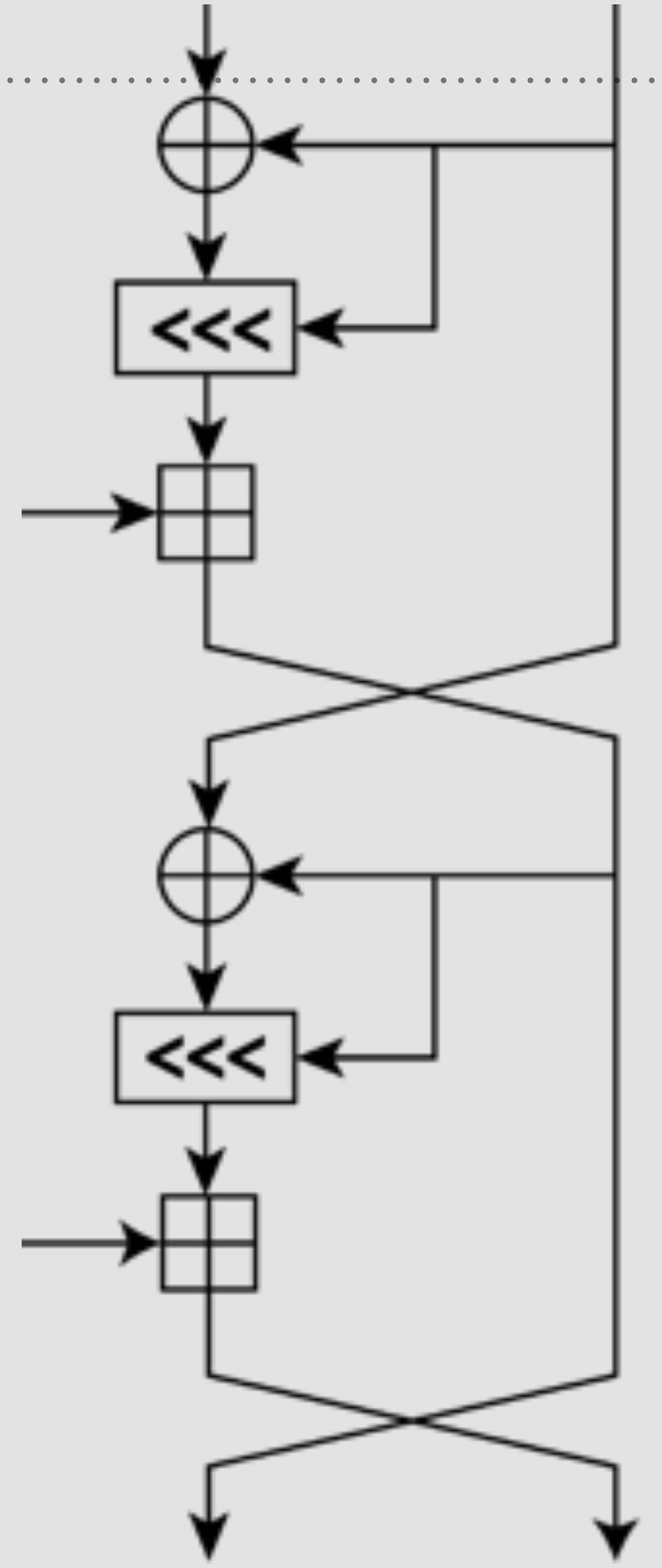
TAKE AWAY

- Public-key cryptography has no unconditional security.
- It requires to make computational assumptions: it is impossible to recover the secret key (or an equivalent key) from the public key.
- But then what is <u>impossible</u>?

WHAT IS THE LARGEST COMPUTATION EVER?

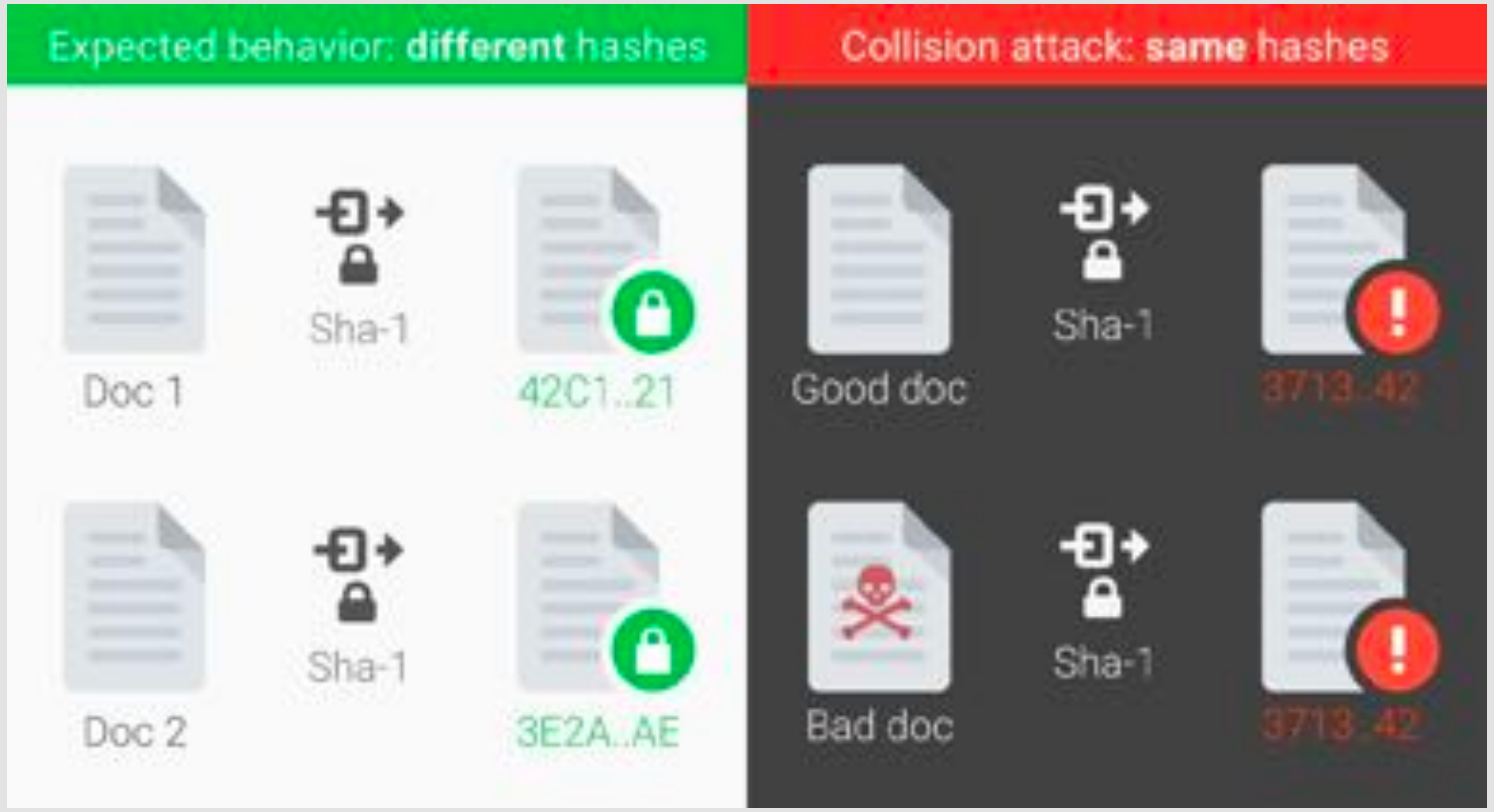
THE LARGEST (PUBLIC) COMPUTATION EVER

- Duration: 4 years, from 1998 to 2002.
- 2^{64} RC5 encryptions ≈ 2^{74} clock cycles.
- Up to 300,000 PCs used on the Internet.
- But this is much less than Bitcoin computations.



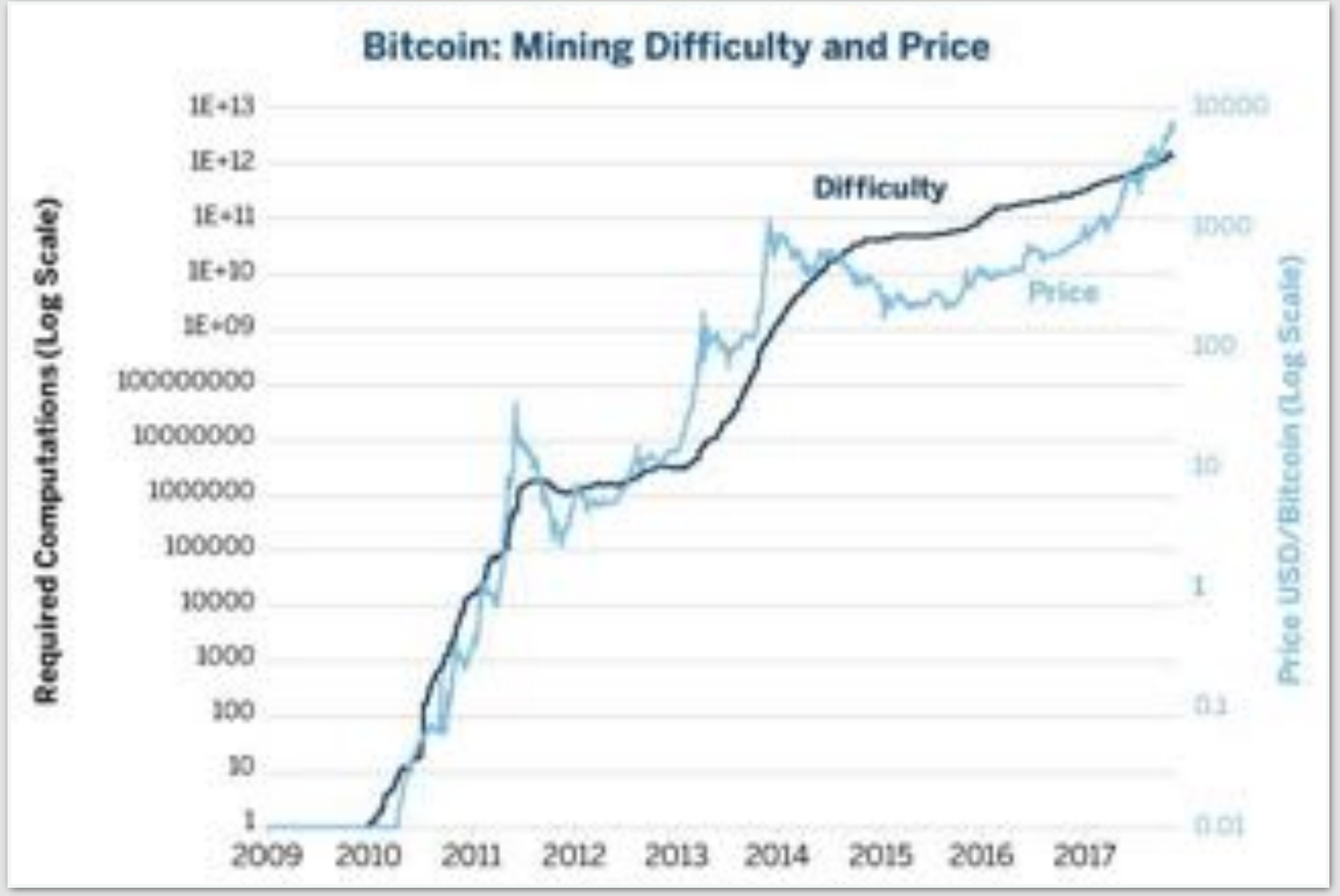
MORE RECENTLY

- In Feb. 2017, an international team (Stevens et al.) found the first SHA-1 collision using approximately 2^{63} hash computations.
- It took 6500 CPU years and 100 GPU years.



BITCOIN POWER

- If D is the current Bitcoin difficulty, mining requires to calculate D x $2^{32}/600$ hashes per second.
    - In Nov 2023, D≈62,4*10^{12} so #hash/sec≈2^{68} and #hash/year≈2^{93}
- D is updated so that mining takes about 10 minutes.



2^{64}

- Any large organization can perform 2^{64} cryptographic operations secretly:
  - Bitcoin does it every 0.0625 second.
  - 20 years ago, it was a record computation.
- That is an order of magnitude.

COMPARISONS

- Number of milliseconds in 100 years = 100x365x24x60x60x1000 ≈ 2^{42}
- World population ≈ 7 billions = 7 x 10^9 ≈ 2^{33}
- Earth-Sun distance = 150 x 10^6 km ≈ 2^{47} mm
- Age of earth = 4,5 x 10^9 years ≈ 2^{57} seconds

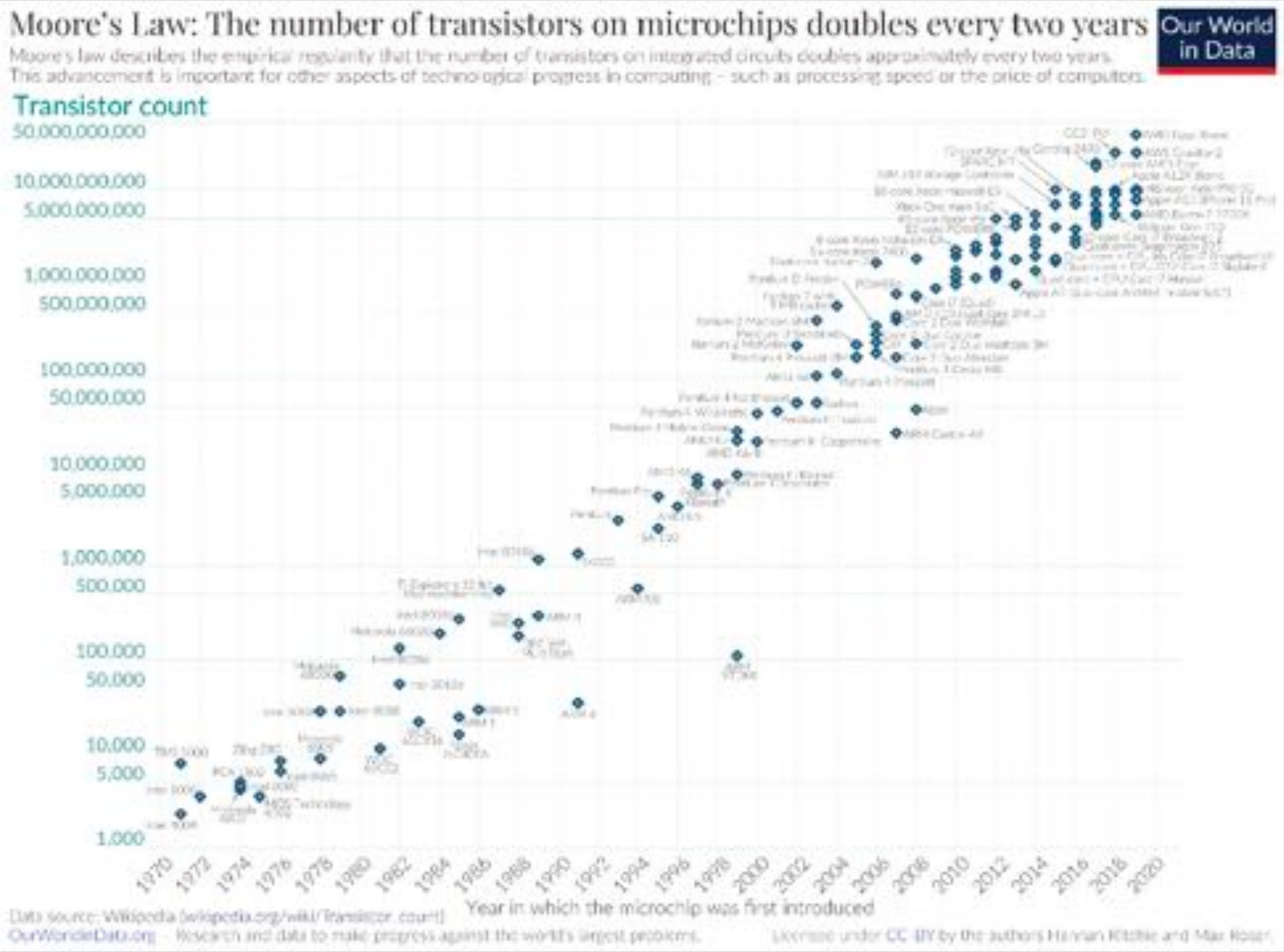
WHAT IS « IMPOSSIBLE »?

- The PC computing power sold per year is roughly 2^{86} clock cycles.
  - 2-GHz core = $2 \times 10^9 = 2^{31}$ clock cycles / sec.
  - In 2022, a typical CPU has 4 cores.
  - 1 year = $60 \times 60 \times 24 \times 365 = 2^{25}$ seconds.
  - About 2^{28} PCs sold per year.
- Total number of computers ever sold: $4 \times 10^9 \approx 2^{32}$?
- Total number of cycles per year: about 2^{90}

➤ Impossible := 2^{128} operations

MOORE'S LAW

- Computing power doubles roughly every 18 months.
- 2011's QuadCore has 2200 times more cycles/s than 1981's IBM PC.



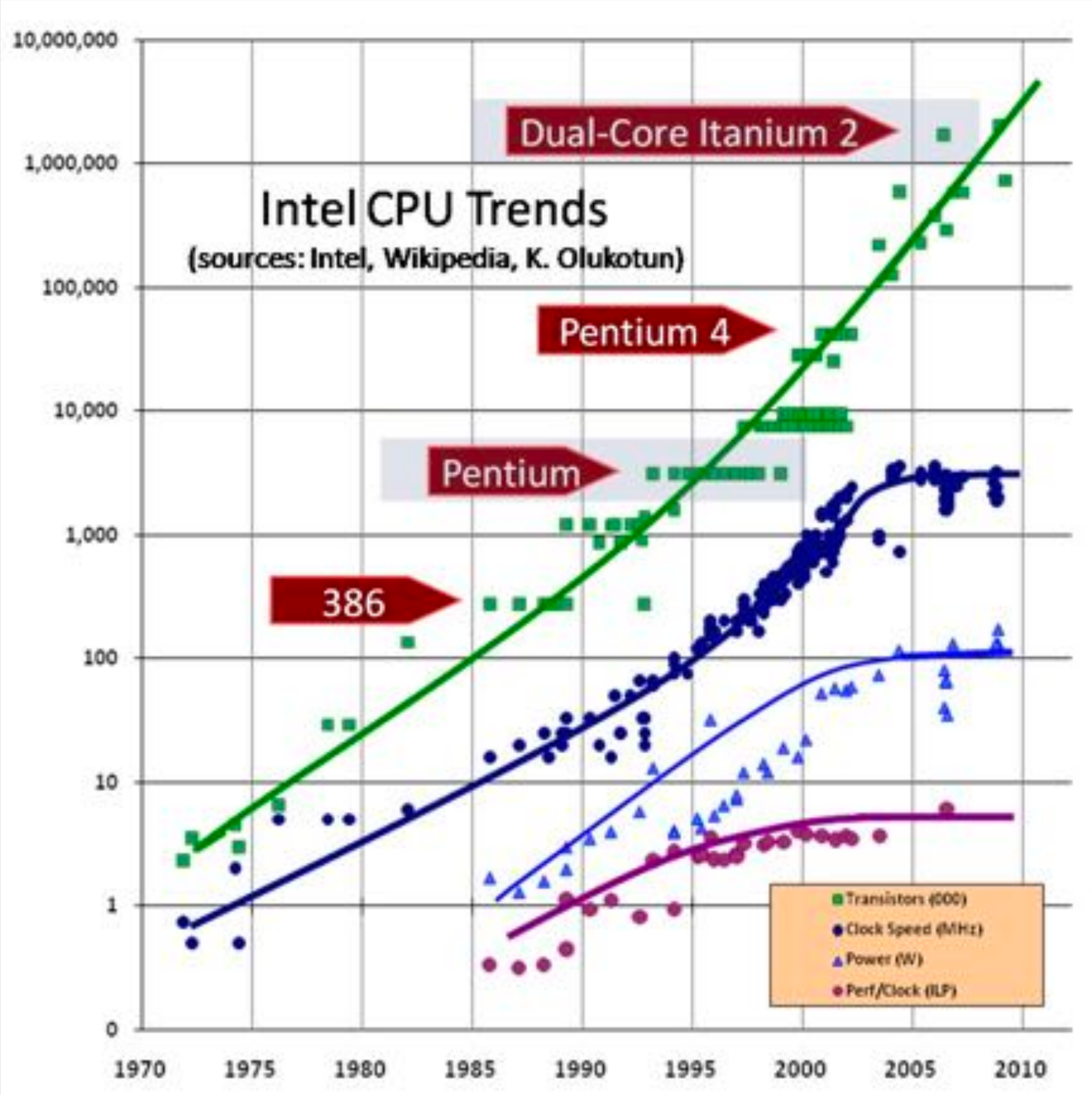
MOORE'S LAW LIMITS

- Frequency stalls

- Number of cores increases

Multicore

GPU



MASSIVE COMPUTATIONAL POWER

- In 2023:
  - The top supercomputer (Frontier, cost 600 million \$) performs

    1,000 PetaFlop/s = 2^{84} floating-point operations/year.
  - BitCoin performs 2^{93} hash/year.

    The hardware cost is ≤ 400 million \$.

SECURITY LEVELS

- Weak: 40 bits.
- Average: 64-80 bits.
- Strong: 128 bits
- High: 256 bits

➤ No precise definition of operation: it is an order of magnitude.

➤ For instance, AES has 3 key-lengths: 128, 192 and 256 bits.

COMPARISON WITH PASSWORDS

Number of characters	Lower case (26)	Lower + Digits (36)	Lower + Upper + Digits (62)	Keyboard (95)
5	$2^{23.5}$	$2^{25.9}$	$2^{29.8}$	$2^{32.9}$
8	$2^{37.6}$	$2^{41.4}$	$2^{47.6}$	$2^{52.6}$

STORAGE POWER

- A typical PC hard-disk has 1Tb = 8×10^{12} bits $\approx 2^{43}$ bits.
- The number of HDD shipped per year is roughly 560 millions, i.e. 2^{72} bits in total.
- The NSA Utah data center: rumors of 3-12 exabytes (10^{18} bytes) = about 2^{62} bits.
- Internet traffic in 2023: 4.1 zettabyte $\approx 10^{23}$ bytes = 2^{75} bits