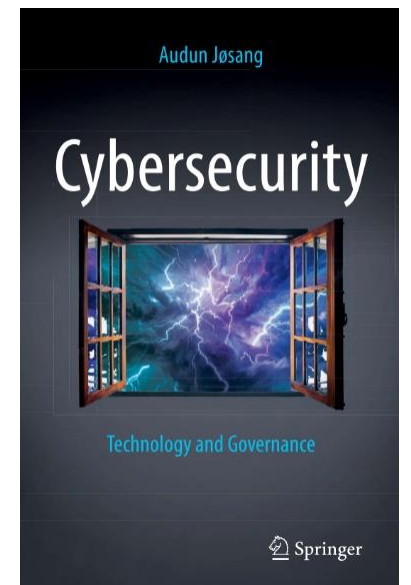


Chapter 4: Cryptography

Cybersecurity: Technology and Governance

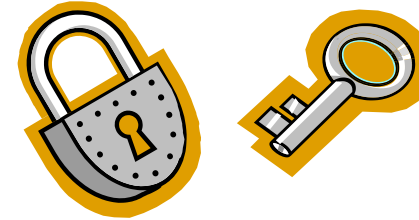
Audun Jøsang,

1st Edition, Springer Nature, 2024.

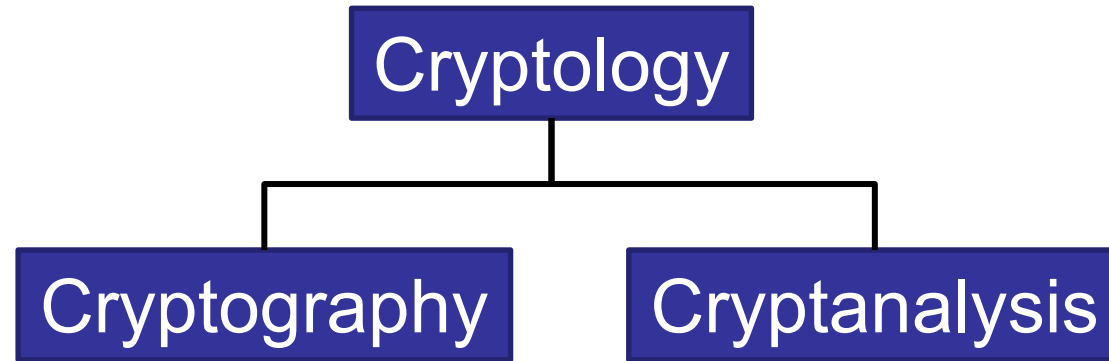


Overview

- What is cryptography?
- Symmetric cryptography
 - Brief crypto history
 - Power cipher
 - Block cipher
 - Hash functions
- Asymmetric cryptography
 - Asymmetric encryption in general
 - Diffie-Hellman key exchange
 - Digital signatures
- Post-Quantum Crypto

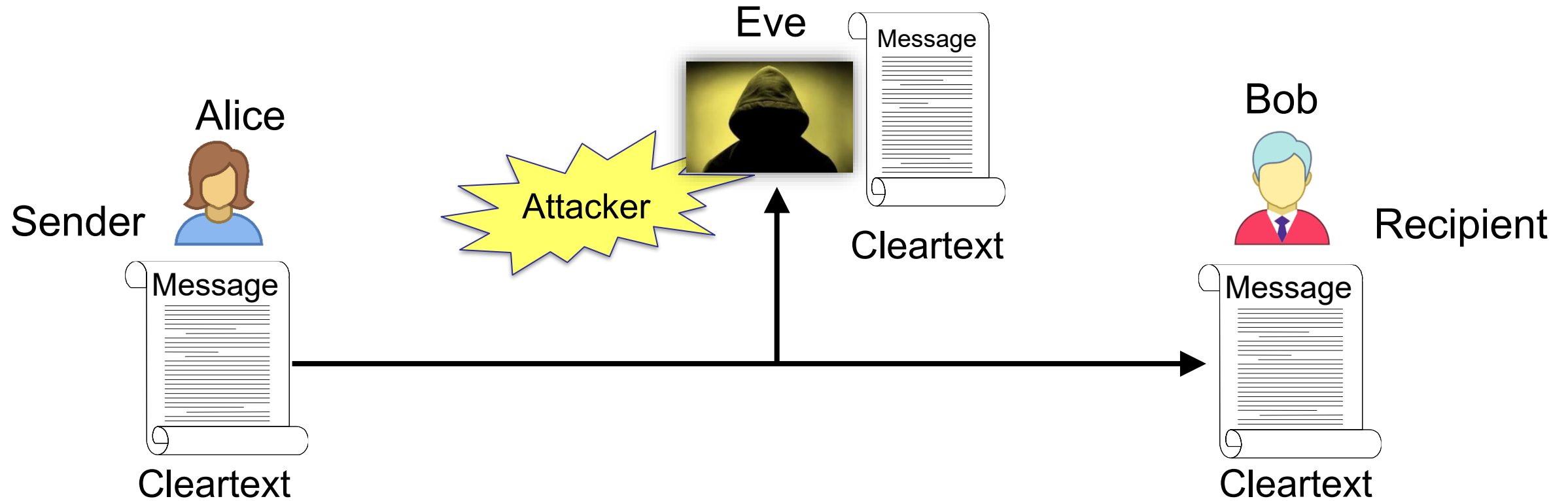


Terminology



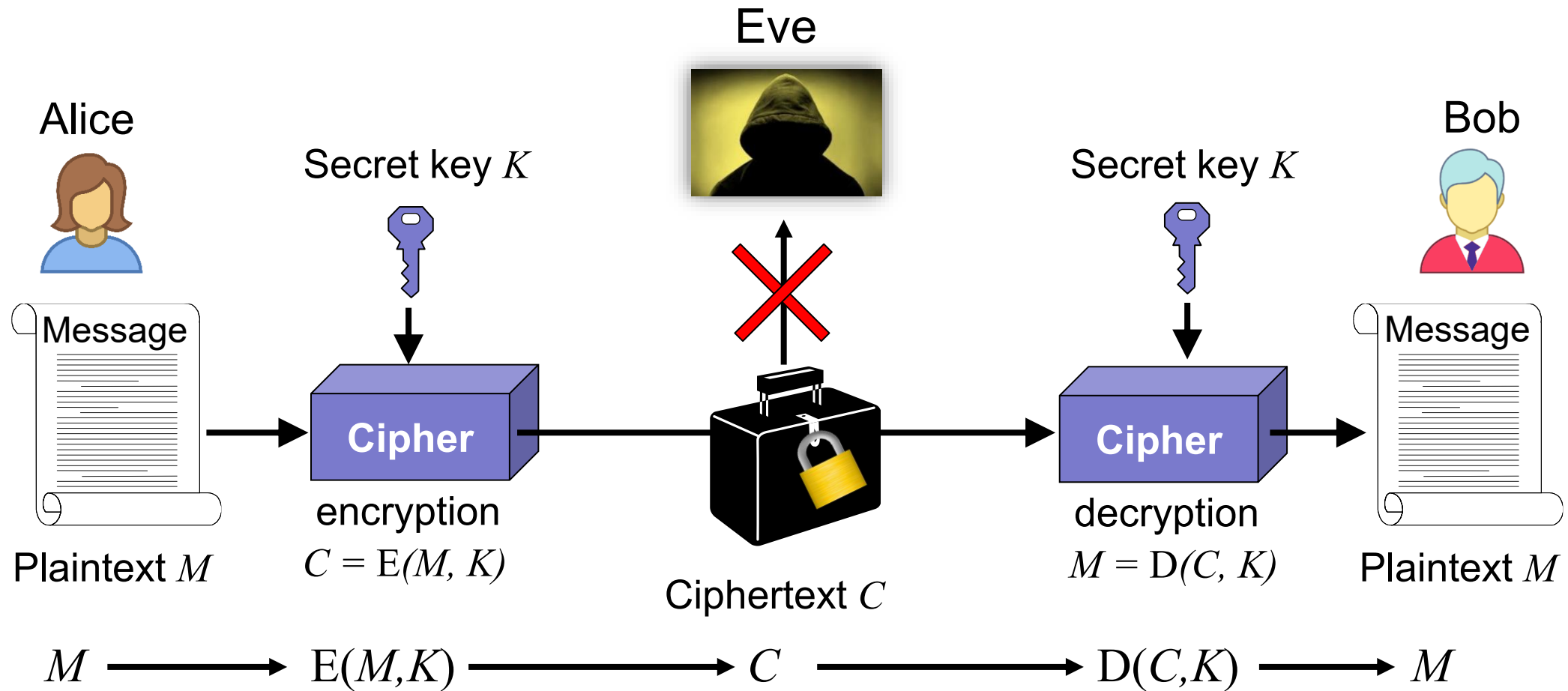
- **Cryptography** is the science of secret writing for the purpose of concealing the meaning of a message.
- **Cryptanalysis** is the science of cracking cryptography.
- **Cryptology** covers both cryptography and cryptanalysis.

Need for cryptography at a glance



- Cleartext is text that can be directly interpreted and is not intended for encryption
- Plaintext is the unencrypted form of text intended for encryption

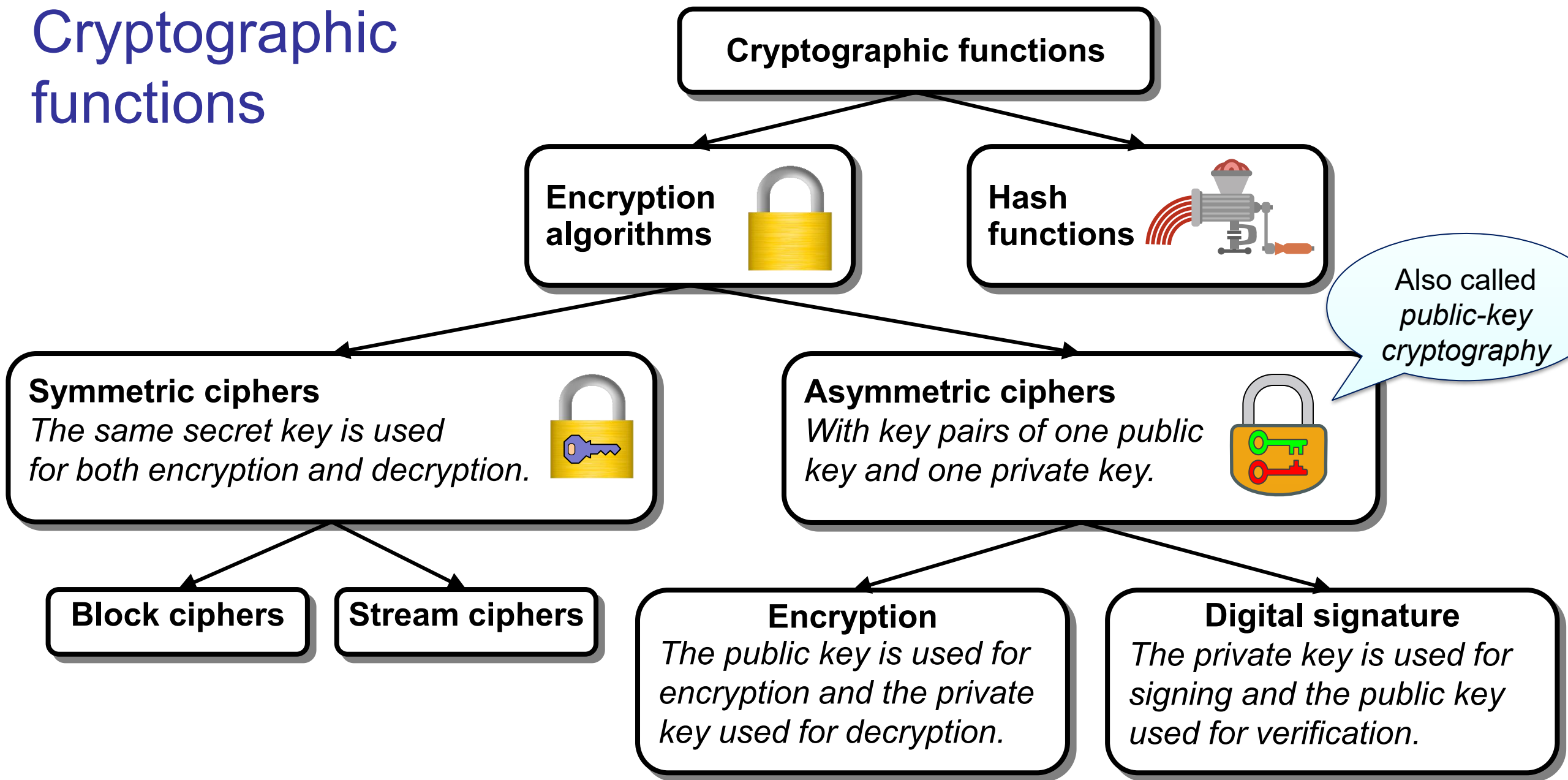
Application of cryptography at a glance



What can cryptography be used for?

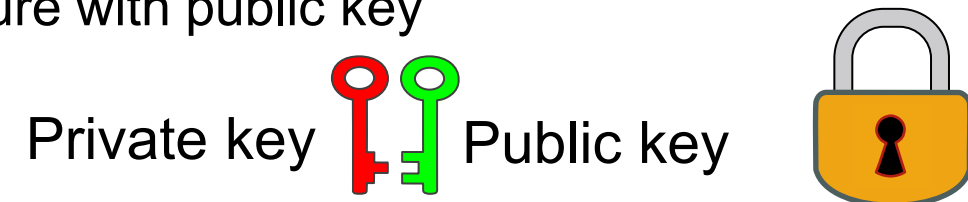
- Cryptography supports the following security objectives:
 - **Confidentiality:**
 - Make data unreadable to entities that don't have the correct cryptographic keys, even if they do have the data.
 - **Data Integrity:**
 - Entities with the correct cryptographic keys can verify that data is correct and has not been altered by unauthorized persons.
 - **Authentication:**
 - Entities that communicate can gain certainty that the identity of the other user/entity or sender of a message is what it claims to be.
 - **Digital Signature and PKI (Public-Key Infrastructure):**
 - Strong proof of data authenticity that can be verified by third parties.
 - Scalable (to the entire Internet) secure distribution of cryptographic keys.

Cryptographic functions

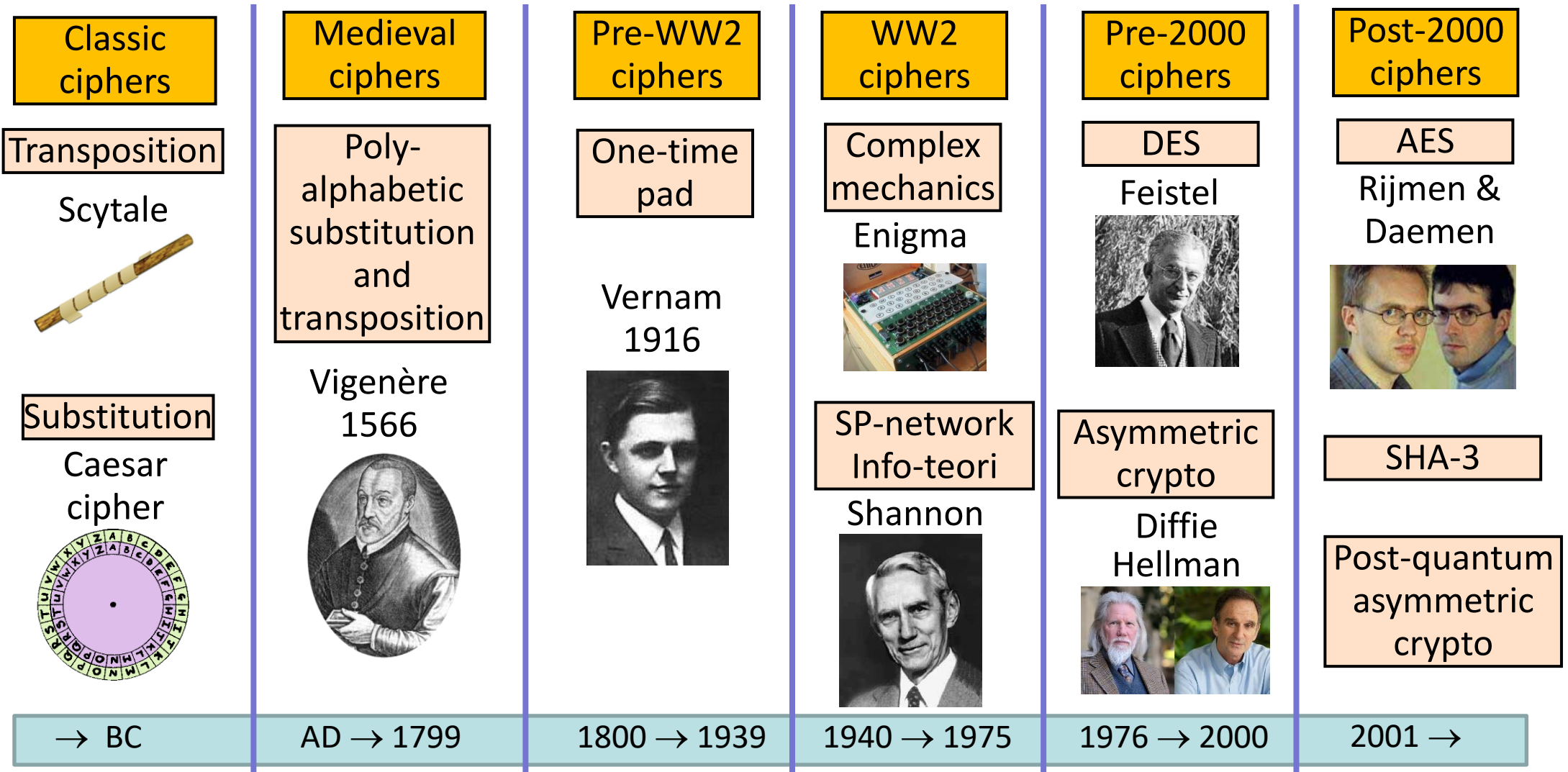


Terminology and notation

- **Encryption:** Plaintext M is transformed to ciphertext C with encryption function E controlled by encryption key k .
 - Formal notation: $C = E(M, k)$.
- **Decryption:** Ciphertext C is transformed back to plaintext M with decryption function D controlled by encryption key k .
 - Formal notation: $M = D(C, k)$.
- **Symmetrical cipher:** The same secret key is used for both encryption and decryption. The secret key is shared between authorized entities.
- **Asymmetric cipher:** Key pair with a private and a public key.
 - Encryption with public key and decryption with private key
 - Digital signature with private key and verification of signature with public key
 - The private key is known only to the key owner
 - The public key can be shared with everyone.



History of Cryptography



The strength of a cipher



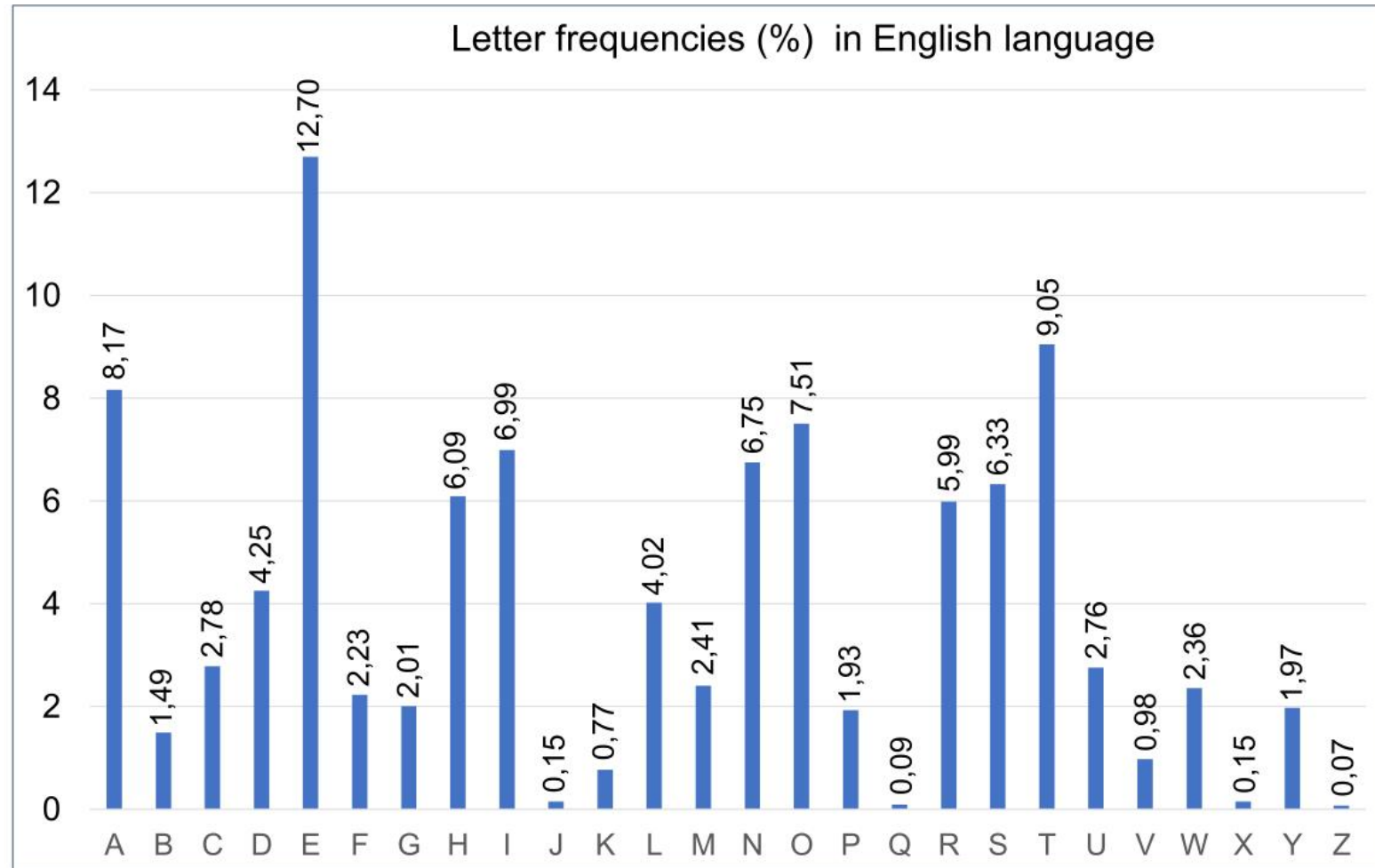
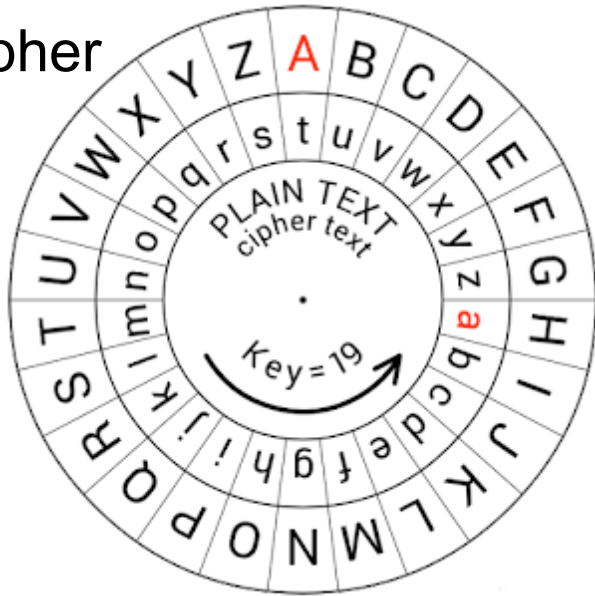
Factors that determine cipher strength:

- **Key size.**
 - The required time for an exhaustive search among all keys depends on the size.
 - Typical size for symmetric block cipher is 256 bits.
 - Attacker thus must try on average $2^{256}/2$ different keys to find the right one, something that would take millions of years and is therefore impractical.
 - If there are N different keys, the key size will be: $\log_2(N)$.
- **The strength of the algorithm.**
 - Finding the key by cryptanalysis can exploit statistical irregularities in the ciphertext.
 - To prevent cryptanalysis, the bit patterns/characters in the ciphertext should have a uniform/even distribution, that is, all bit patterns/characters should be equally likely.

Letter Frequencies → Statistical Cryptanalysis

- Classical ciphers, such as the Caesar cipher, are weak because they fail to hide statistical irregularities in the ciphertext.

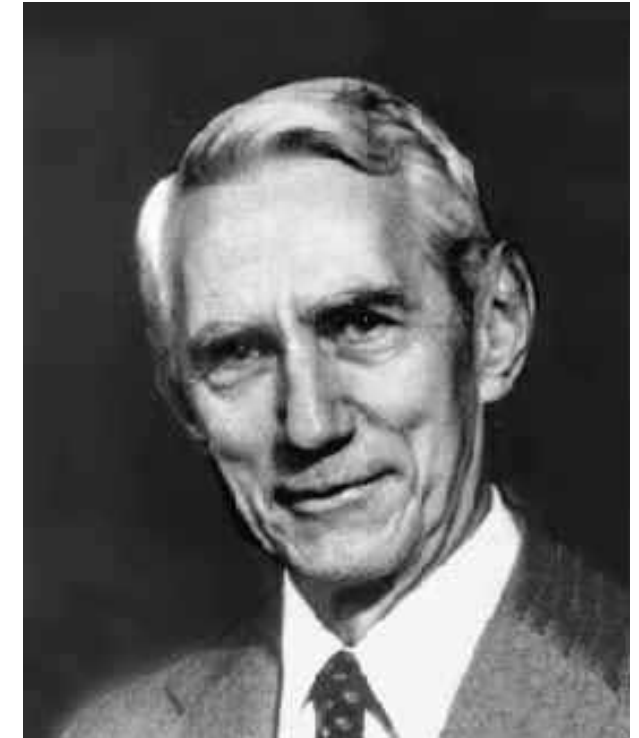
Caesar cipher



Claude Shannon (1916 – 2001)

The father of information theory – MIT / Bell Labs

- Information theory
 - Defined "binary digit" (bit) as the smallest unit of information
 - Defined information entropy as a measure of the amount of information
- Cryptography
 - Model for secrecy of cryptographic systems
 - Defined perfect secrecy (see later)
 - Principle of encryption with SP network (Substitution and Permutation) to erase statistical patterns



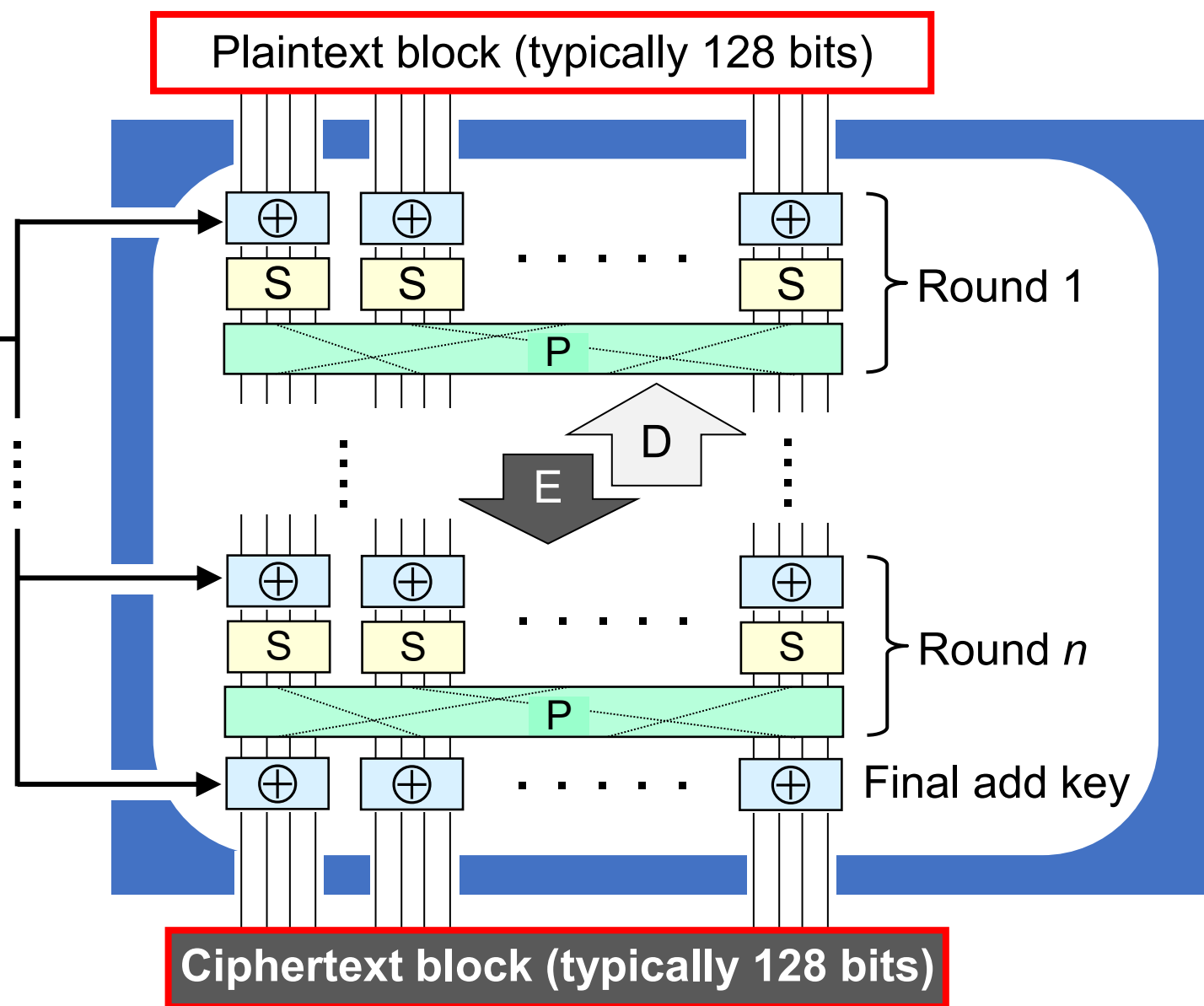
Claude Shannon

Shannon's SP-network

Erases statistical patterns

- Repeat substitution and permutation, typically 10-20 rounds.
- Substitution S
 - Sub-blocks are substituted (replaced) with other blocks
 - “Confusion” to hide connection between plaintext and ciphertext.
- Permutation P
 - Sub-blocks are moved around.
 - “Diffusion”, i.e. changing a single bit in plaintext gives a large change in ciphertext.
- The key is included in S or P, or in a separate add-key function $\oplus$

Key 



AES - Advanced Encryption Standard

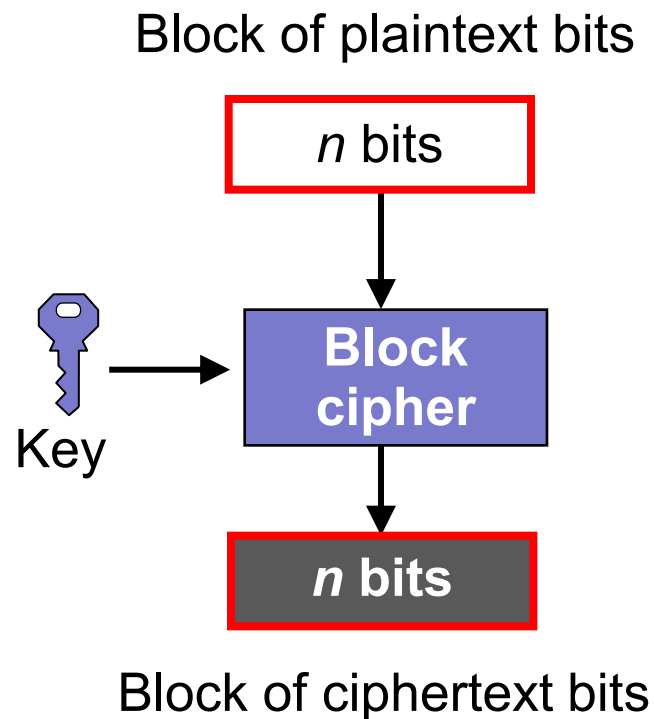
- The 1977 DES (Data Encryption Standard) had a 56-bit key and a 64-bit block. In the mid-1990s, DES could be cracked with exhaustive key search.
- In 1997, NIST announced an open competition to design a new block cipher to replace DES.
- The best proposal called "Rijndael" (designed by Vincent Rijmen and Joan Daemen from Belgium) was considered the best, and nominated to become AES (Advanced Encryption Standard) in 2001.
- AES has key sizes of 128, 192, or 256 bits and block sizes of 128 bits.
- AES follows Shannon's principle of SP network.

Joan Daemen and Vincent Rijmen

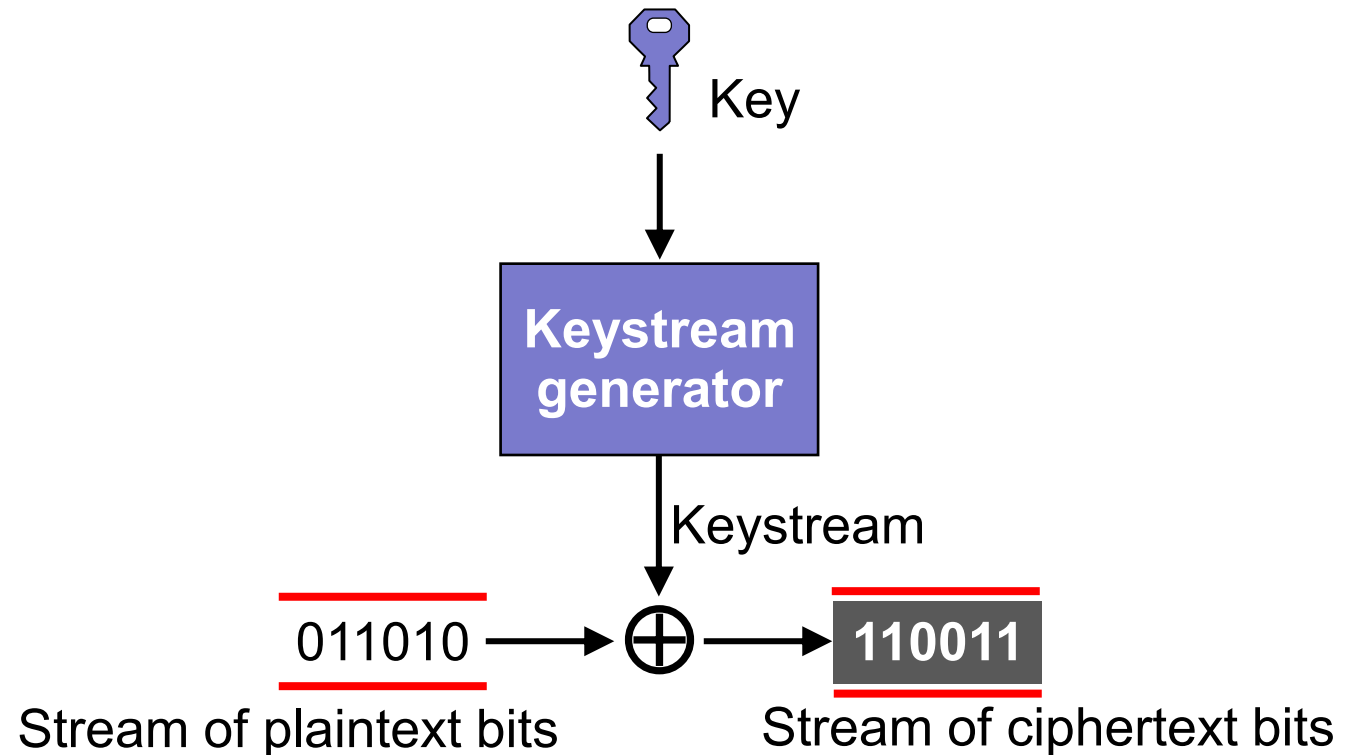


Block cipher and stream cipher

Block cipher



Stream cipher



Block cipher: Modes of operation

- A block cipher encrypts a block of (typically) 128 bits, which is only about 16 characters.
- For encryption of more than one block, a specific mode is used.
- The encryption modes have different characteristics.
- Common modes are:

- **CounTeR** Mode (CTR)
- **Cipher Block Chaining** (CBC)
- **Output FeedBack** (OFB)
- **Cipher FeedBack** (CFB)



secure

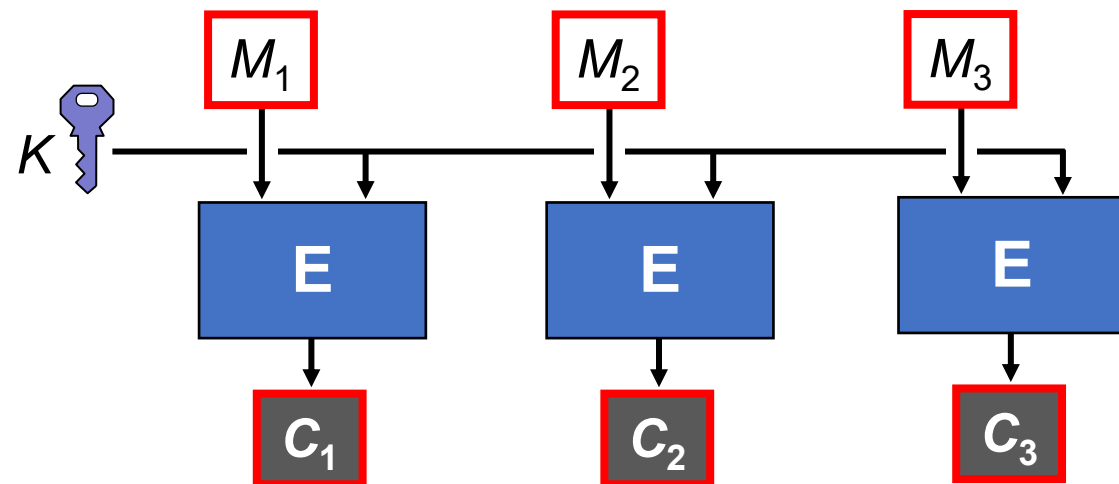
- **Electronic Code Book** (ECB)



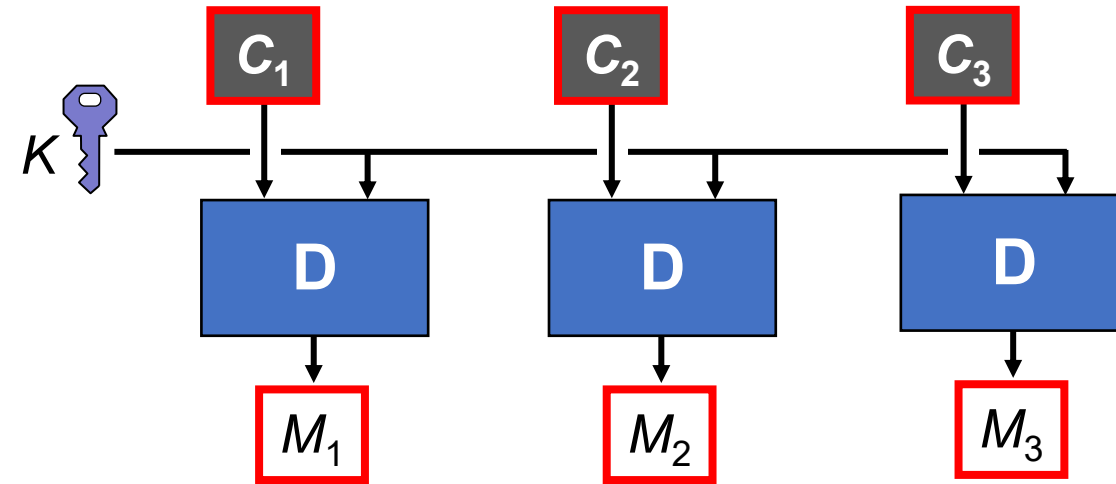
insecure

Electronic Code Book (ECB)

- The simplest mode of operation, but insecure
- The plaintext is divided into blocks $M_1, M_2, \dots, M_n$
- Each block is encrypted separately.
 - Notation for encryption: $C_1 = E(M_1, K)$
 - Notation for decryption: $M_1 = D(C_1, K)$
 - Identical plaintext blocks give identical ciphertext blocks, this is the problem!



Encryption



Decryption

Vulnerability of using ECB mode



Plaintext



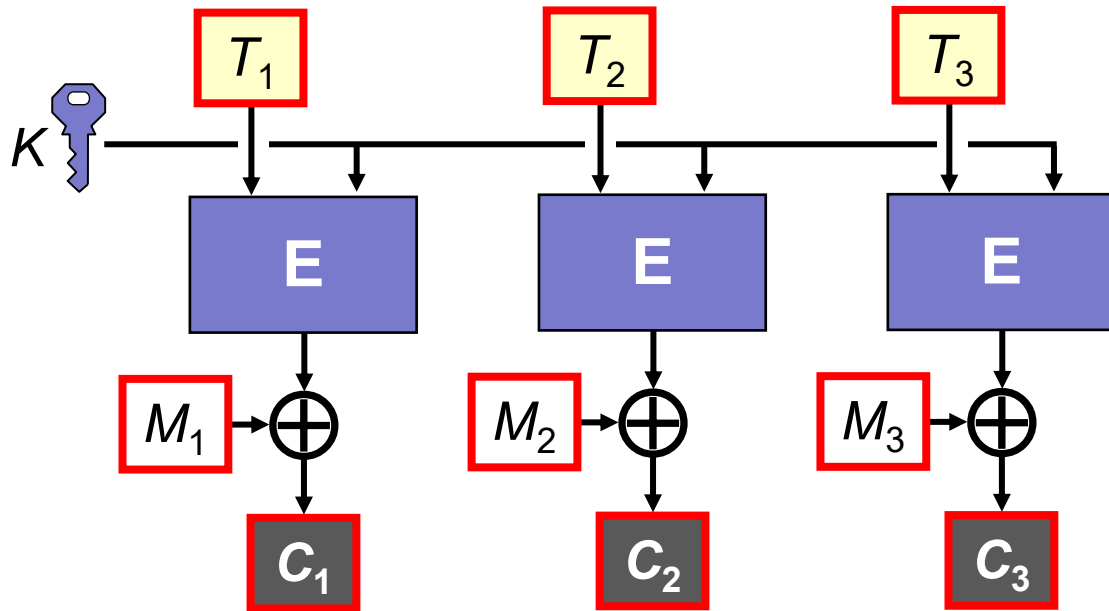
Ciphertext with ECB mode



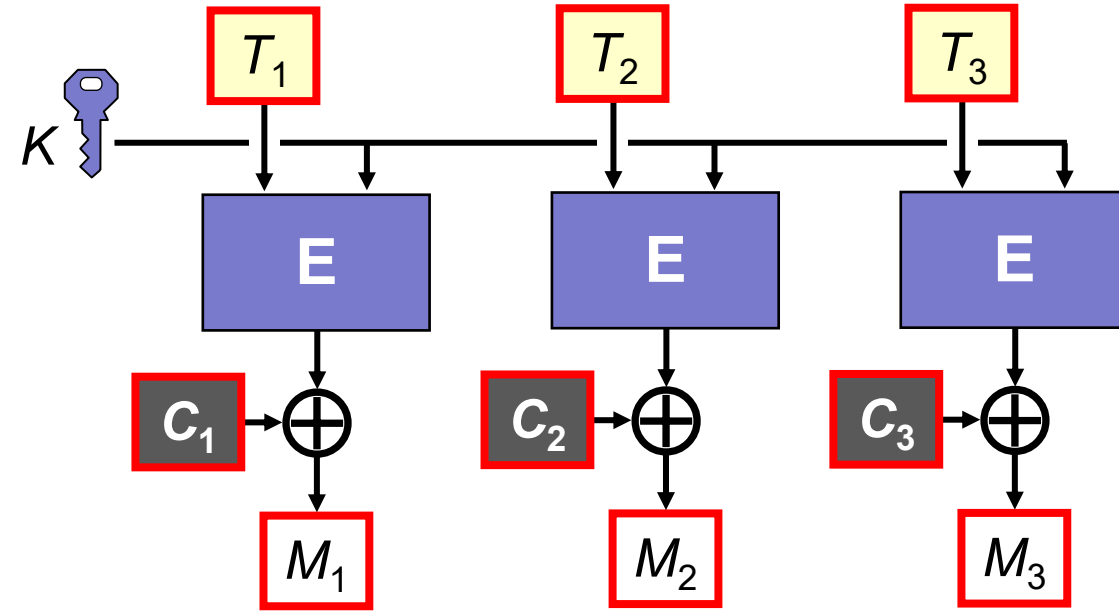
Ciphertext with secure mode

Counter Mode (CTR)

- The plaintext is divided into blocks $M_1, M_2, \dots, M_n$
- An incremental counter value T is consecutively encrypted
- Each encrypted count value is added to the plaintext block with binary XOR $\oplus$
 - Identical plaintext blocks are encrypted to different ciphertext blocks, this provides security!



Encryption



Decryption

CTR encryption and binary addition with XOR

- The plaintext is divided into blocks: $M_1, M_2, \dots, M_n$
- Incremental counter values $T_1, T_2, \dots, T_n$ have the same size as the block
- Initial counter value set with random nonce value
- The counter values are encrypted and added to the plaintext blocks:
 - Notation for encryption: $C_1 = E(T_1, K) \oplus M_1$
 - Notation for decryption: $M_1 = E(T_1, K) \oplus C_1 = E(T_1, K) \oplus E(T_1, K) \oplus M_1$
 - The encryption function E is used for both encryption and decryption
- Binary addition with XOR $\oplus$
 - Addition of bit with itself always gives zero
- Example encryption and decryption: $M_1 = 1111$ $E(T_1, K) = 1001$
 - Encryption: $C_1 = 1001 \oplus 1111 = 0110$
 - Decryption: $M_1 = 1001 \oplus 0110 = 1111$

$0 \oplus 0 = 0$	$0 \oplus 1 = 1$
$1 \oplus 1 = 0$	$1 \oplus 0 = 1$

Essential property of XOR

$$M \oplus U \oplus U = M$$

- No matter what the unknown U or message M is, message M will be unchanged after XOR-e with U twice

$$0 \oplus 0 \oplus 0 = 0$$

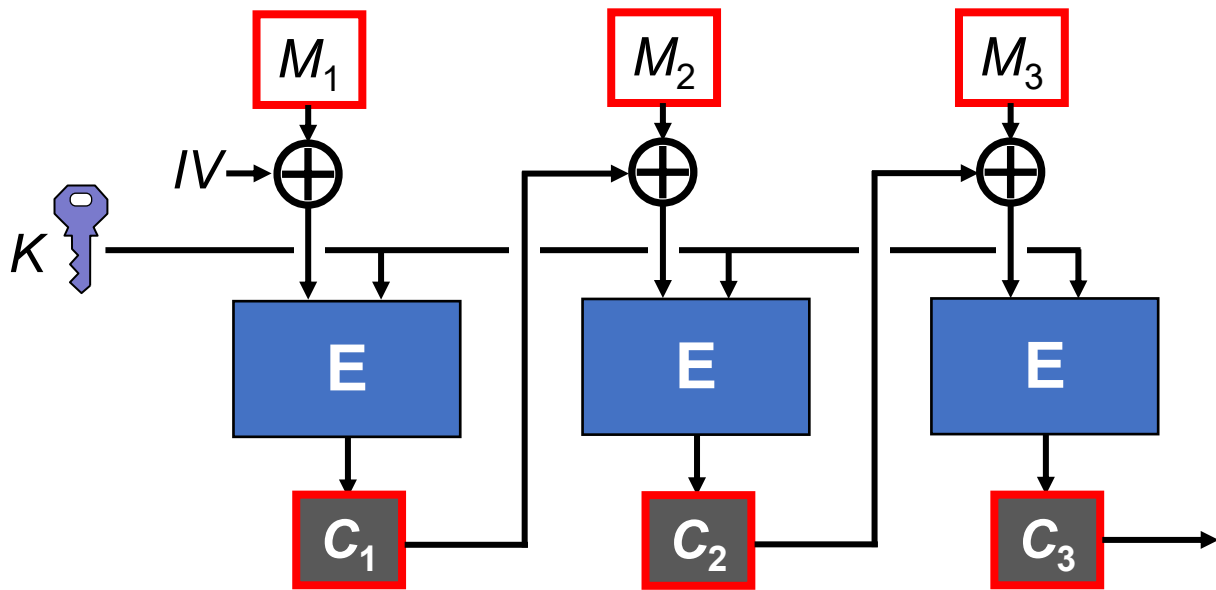
$$1 \oplus 0 \oplus 0 = 1$$

$$0 \oplus 1 \oplus 1 = 0$$

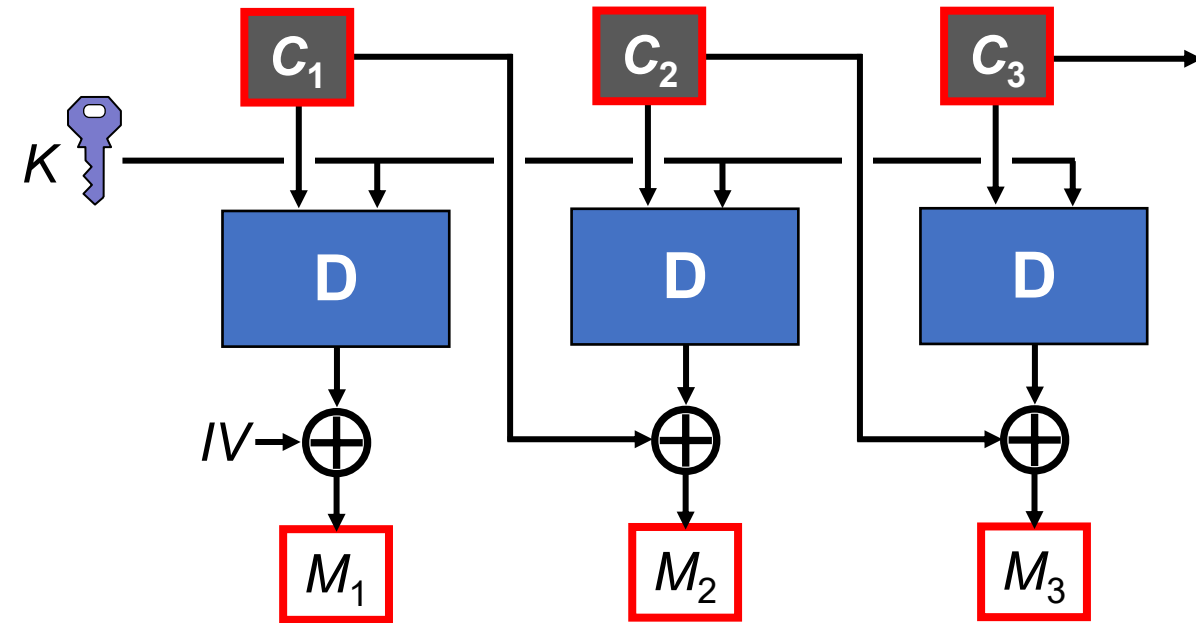
$$1 \oplus 1 \oplus 1 = 1$$

Cipher Block Chaining (CBC)

- The plaintext is divided into blocks $M_1, M_2, \dots, M_n$
- Each cipher block is XOR added to the next plaintext block before encryption
 - Starts with a random IV (initialization vector) that doesn't have to be secret
 - Identical plaintext blocks are encrypted to different ciphertext blocks, this provides good security!



Encryption



Decryption

Notation for CBC

- The plaintext is divided into blocks: $M_1, M_2, \dots, M_n$
- Each cipher block is XOR added to the next plaintext block before encryption

- Notation for encryption:
$$C_1 = E((IV \oplus M_1), K)$$
$$C_2 = E((C_1 \oplus M_2), K)$$
$$C_n = E((C_{n-1} \oplus M_n), K)$$

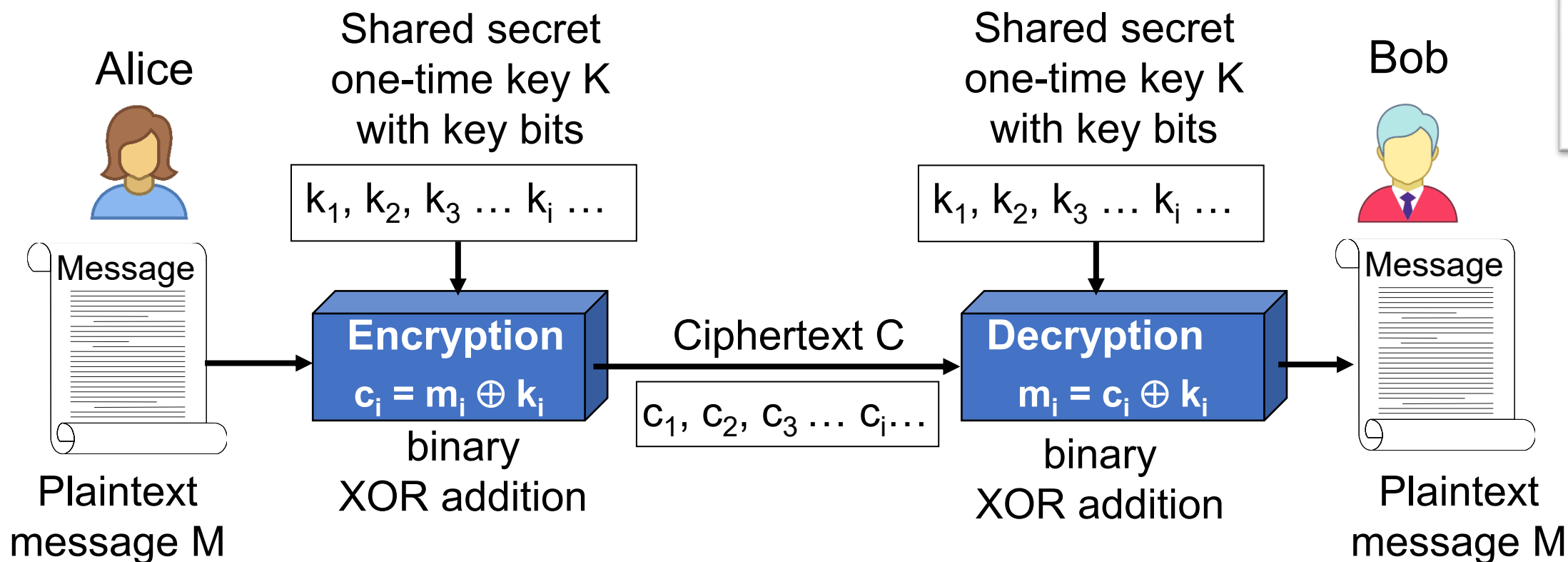
- Notation for decryption:
$$M_1 = D(C_1, K) \oplus IV$$
$$M_2 = D(C_2, K) \oplus C_1$$
$$M_n = D(C_n, K) \oplus C_{n-1}$$

CBC is vulnerable to the padding oracle attack if incorrectly implemented, see:
https://en.wikipedia.org/wiki/Padding_oracle_attack

One-Time Pad (Gilbert Vernam, 1917)



Gilbert Vernam



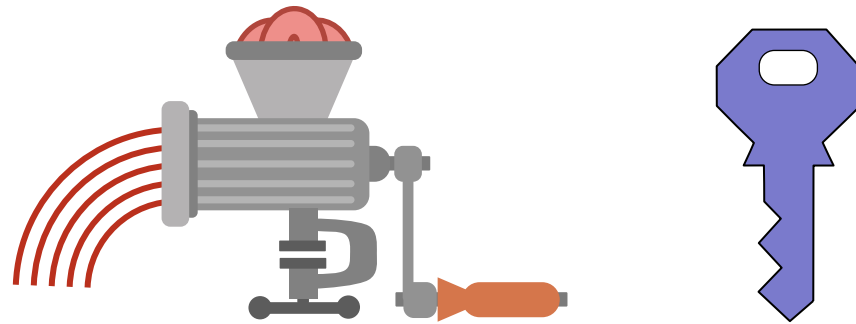
- Bit-by-bit binary XOR addition: $c_i = m_i \oplus k_i$
 $m_i = c_i \oplus k_i = m_i \oplus k_i \oplus k_i = m_i \oplus 0 = m_i$
- OTP provides perfect secrecy by assuming that the OTP key is completely random, of the same length as the message, and is only used once.

The perfect cipher machine: One-Time-Pad



- Telex machine with OTP on perforated tape, manufactured by STK in Oslo, Norway
- Modern versions can use DVDs with Gbytes of random data

HASH FUNCTIONS AND MESSAGE AUTHENTICATION

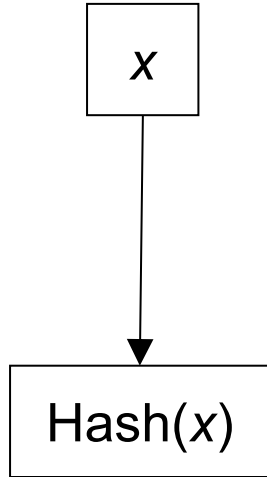


Hash functions

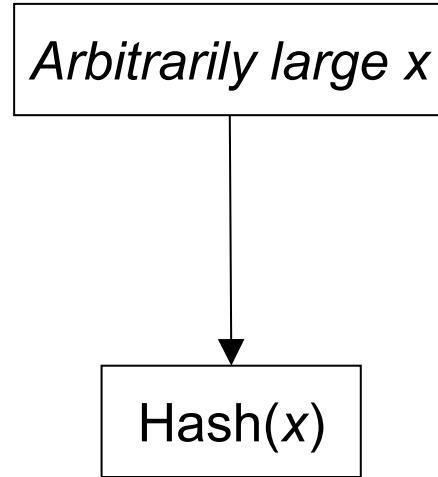
Requirements for a hash function **Hash**:

1. Easy to compute: Given the input **x**, it should be easy to compute **Hash(x)**.
2. Compression: Compresses arbitrarily large **x** into a fixed-size hash value of size **n** (typically 256 bits or 512 bits).
3. One-way: Given the hash value **y**, it should be practically impossible to find input **x** such that **Hash(x) = y**.
4. Collision resistance (weak): Given input **x** and the associated hash value **Hash(x)**, it should be practically impossible to find another input **x'** such that **Hash(x) = Hash(x')** (weak collision resistance).
5. Collision resistance (strong): It should be practically impossible to find two different inputs **x** and **x'** so that **Hash(x) = Hash(x')** (strong collision resistance).

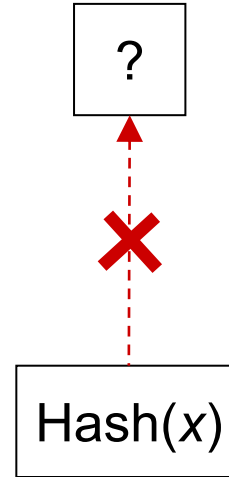
Characteristics of hash functions



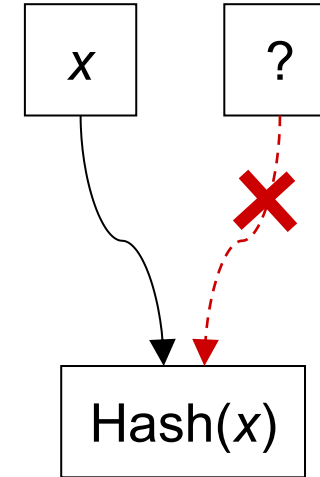
Easy to
compute



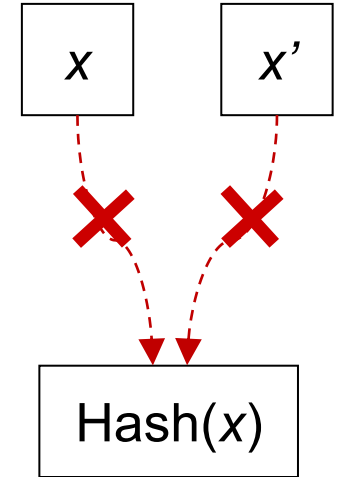
Computed hash
of fixed size



One-way-
function



Weak
collision
resistance



Strong
collision
resistance

Well-known hash functions

- **MD5** (1991): 128 bits hash value. Easy to find collisions, due to small hash size and poor design. No longer to be used.
- **SHA-1** (Secure Hash Algorithm): 160 bit hash value. Designed by the NSA in 1995. Relatively easy to find collisions. No longer to be used, but unfortunately is sometimes still used.
- **SHA-2** designed by the NSA in 2001. Can generate hash values of 256, 384, and 512 bits. Considered safe. Replacement for SHA-1.
- **SHA-3**: designed by Joan Daemen and others in 2010. Standardized in 2015. Can generate hash values of 256, 384, and 512 bits. SHA-3 has little use, because SHA-2 is still considered safe.

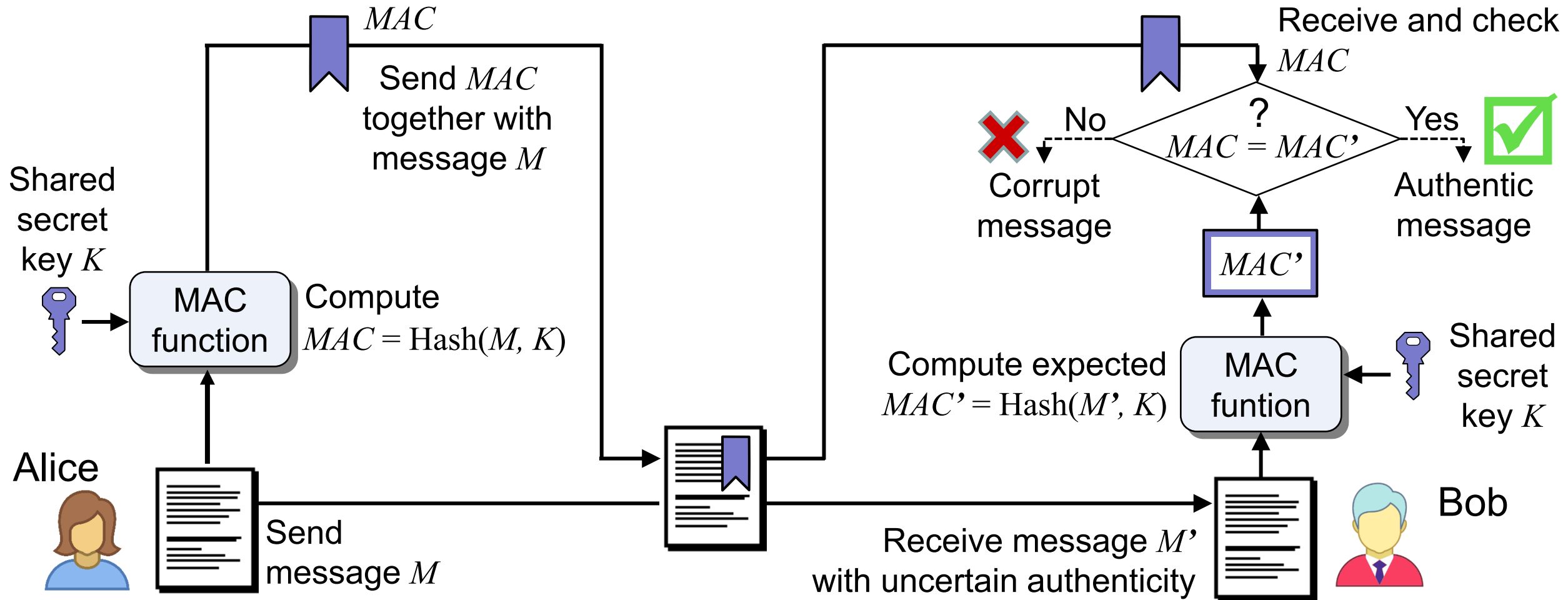
Applications of hash functions

- Comparing files
- Password protection
- Integrity checks, e.g. of SW deployments
- When generating Message Authentication Codes (MAC)
- When generating digital signatures
- For bitcoin and other cryptocurrencies
- Pseudo-random number generation
- Crypto key generation

MAC (Message Authentication Code)

- A message M with hash $\text{Hash}(M)$ can easily be compromised by an attacker.
- To prevent attacks, it is necessary to use an authenticated hash value.
- MAC (message authentication code) includes a secret key k for hash function computation, which provides an authenticated hash value $\text{MAC} = \text{Hash}(M, k)$.
- To authenticate a message, the recipient must have the same secret key that was used by the sender to compute MAC .
- A third party who doesn't know the key can't verify the MAC .

Message authentication with MAC

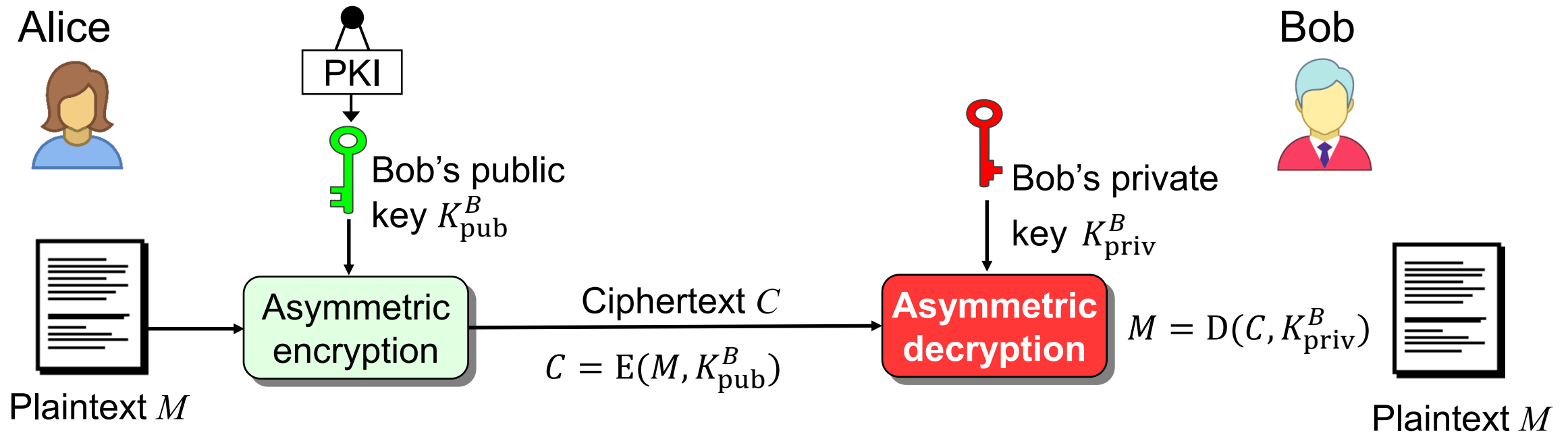


ASYMMETRIC CRYPTOGRAPHY

- Asymmetric encryption
- Diffie-Hellman key exchange
- Digital signature



Asymmetric encryption – basic principle



- Asymmetric encryption and decryption require heavy computation, and are not used for direct encryption as shown above.
- In practice, hybrid encryption is used, which combines both an asymmetric and a symmetric algorithm.

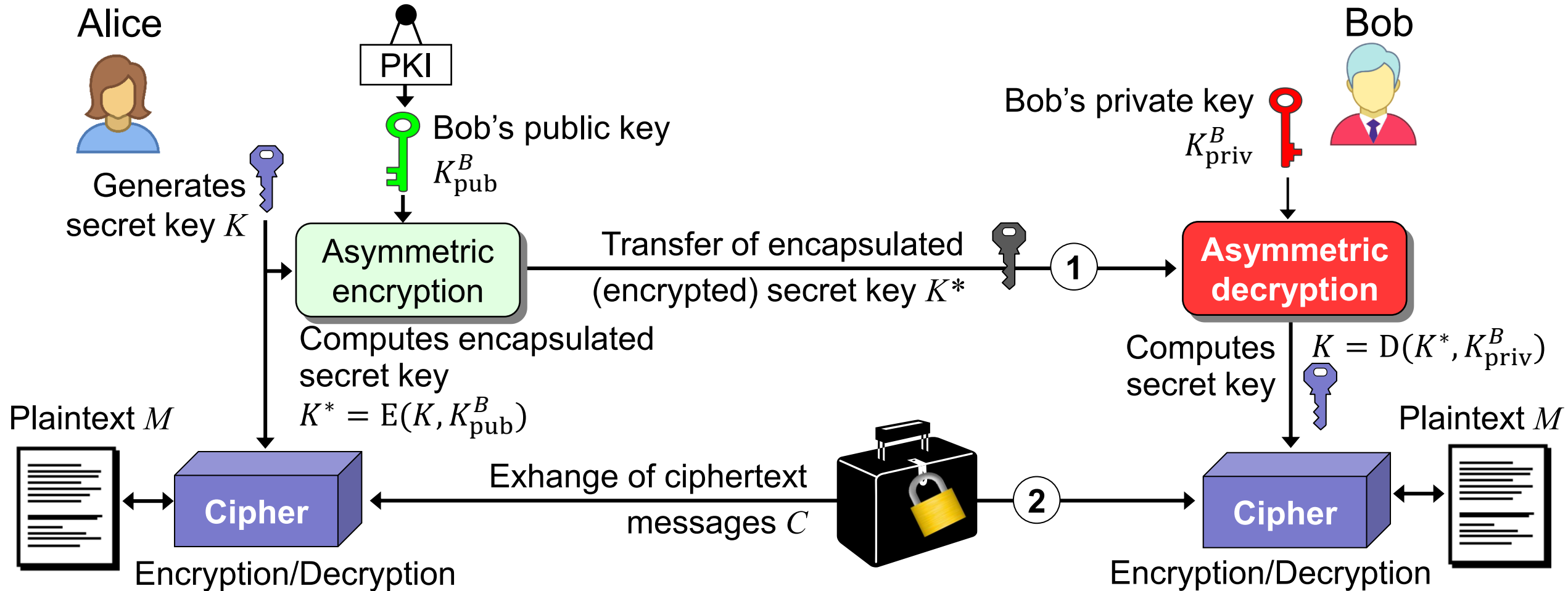
Traditional asymmetric encryption algorithms

- RSA: The most well-known asymmetric algorithm.
 - RSA = Rivest, Shamir and Adleman (published 1977)
 - History: British cryptographer Clifford Cocks invented the same algorithm in 1973, but didn't publish it because it was classified.
 - Eventually requires large keys (typically 2048 bits) to maintain security
- Elliptical Curve Cryptography (ECC)
 - Based on the difficulty of solving EC discrete logarithms.
 - Keys have a smaller size (typically 256 bits) than RSA.

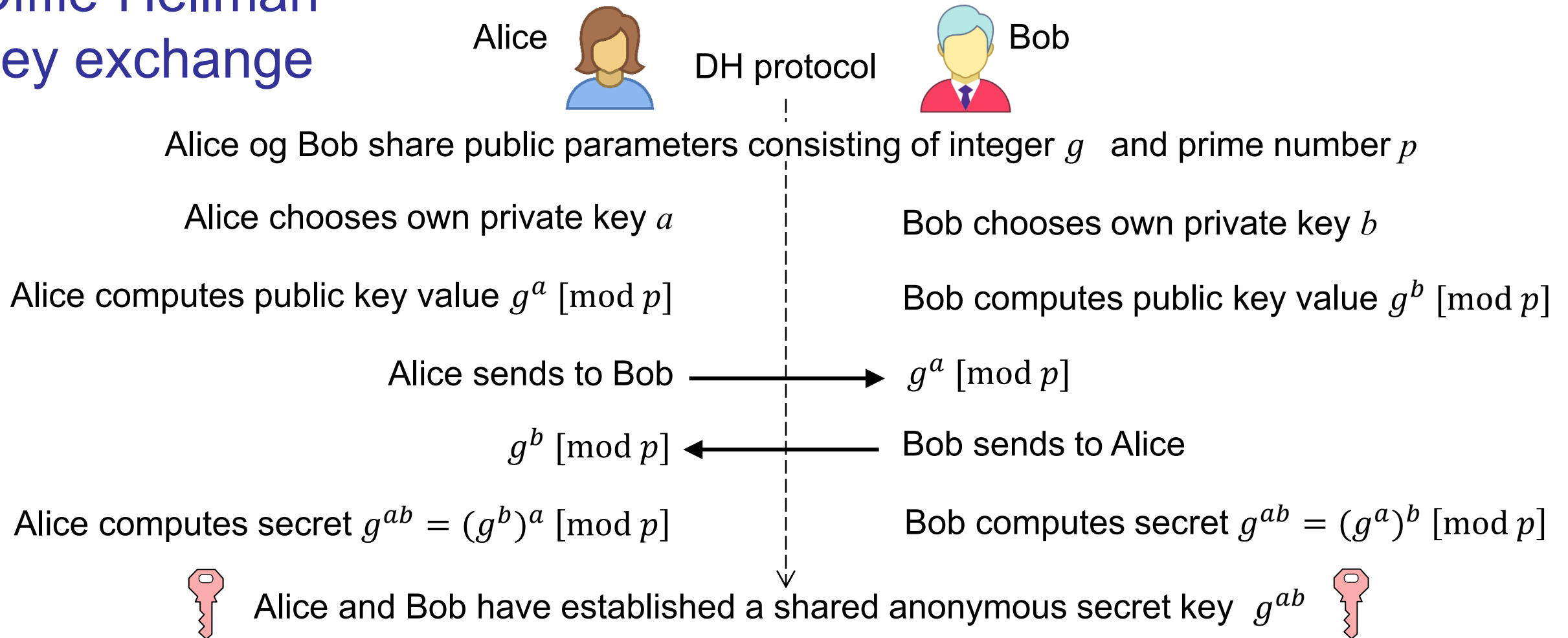
Hybrid encryption

- Symmetric ciphers are much faster than asymmetric ciphers (because symmetric ciphers use relatively simple mathematical calculations), but ...
- Asymmetric ciphers support efficient key distribution, therefore...
- Convenient to use a combination of both symmetric and asymmetric ciphers - a hybrid system:
 - The asymmetric cipher is used to distribute a secret symmetric key.
 - The Symmetric Secret Key with the Symmetric Cipher is used for fast encryption of data sets and messages.
- The weakness of hybrid encryption is that it does not provide forward secrecy.
 - In contrast, forward secrecy can be achieved by establishing session key based on the Diffie-Hellman protocol (as in TLS 1.3).

Hybrid encryption (does not provide forward secrecy)



Diffie-Hellman key exchange



Attackers cannot find the secret subkeys a and b because calculating discrete logarithm of large integers is difficult. I.e. computing of $a = \log(g^a \pmod{p})$ og $b = \log(g^b \pmod{p})$ is difficult. Thus, attackers cannot compute the secret key $= g^{ab} \pmod{p}$.

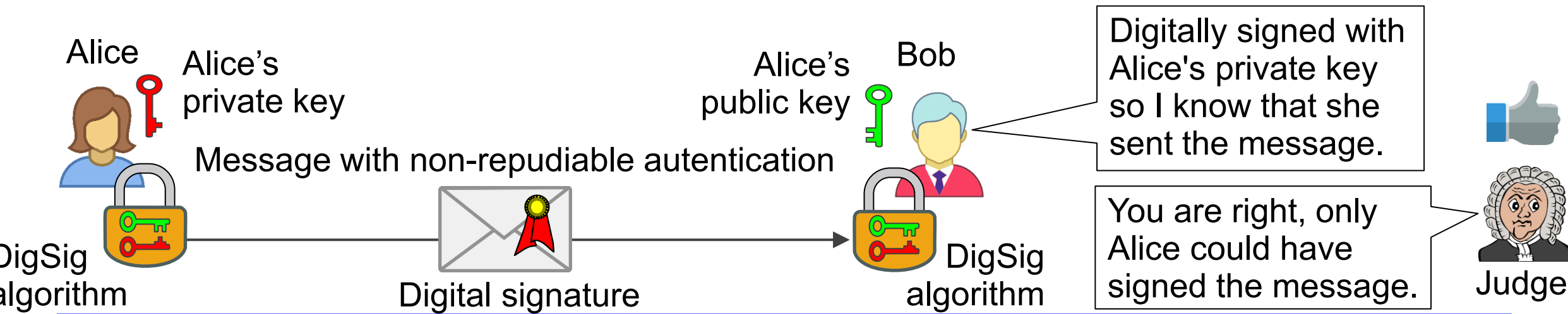
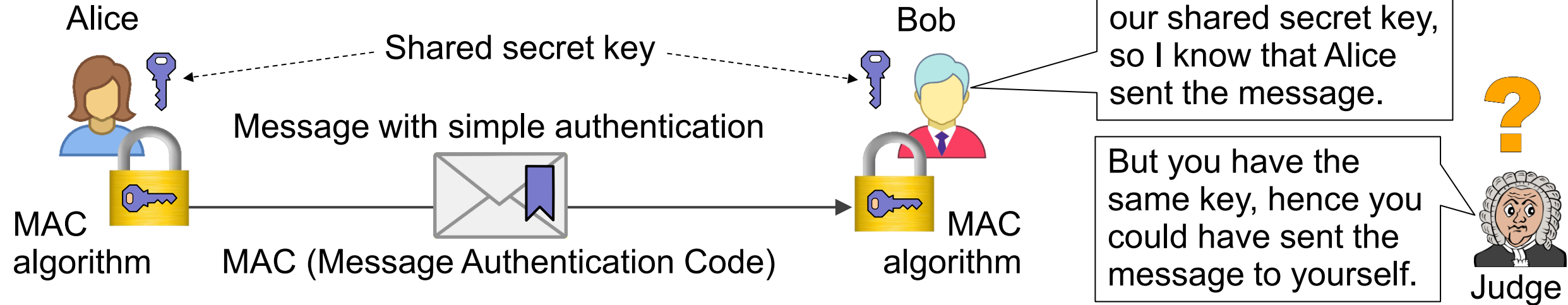
Diffie-Hellman key exchange

- Problem:
 - Provides no authentication
 - Alice and Bob can't know who they're communicating with
 - Man-in-the-middle attack possible
- Solution
 - Combine with digital signature provides authenticated key exchange
- Applications:
 - Transport Layer Security (TLS) and https
 - IKE (Internet Key Exchange) and IPSec (IP Security)

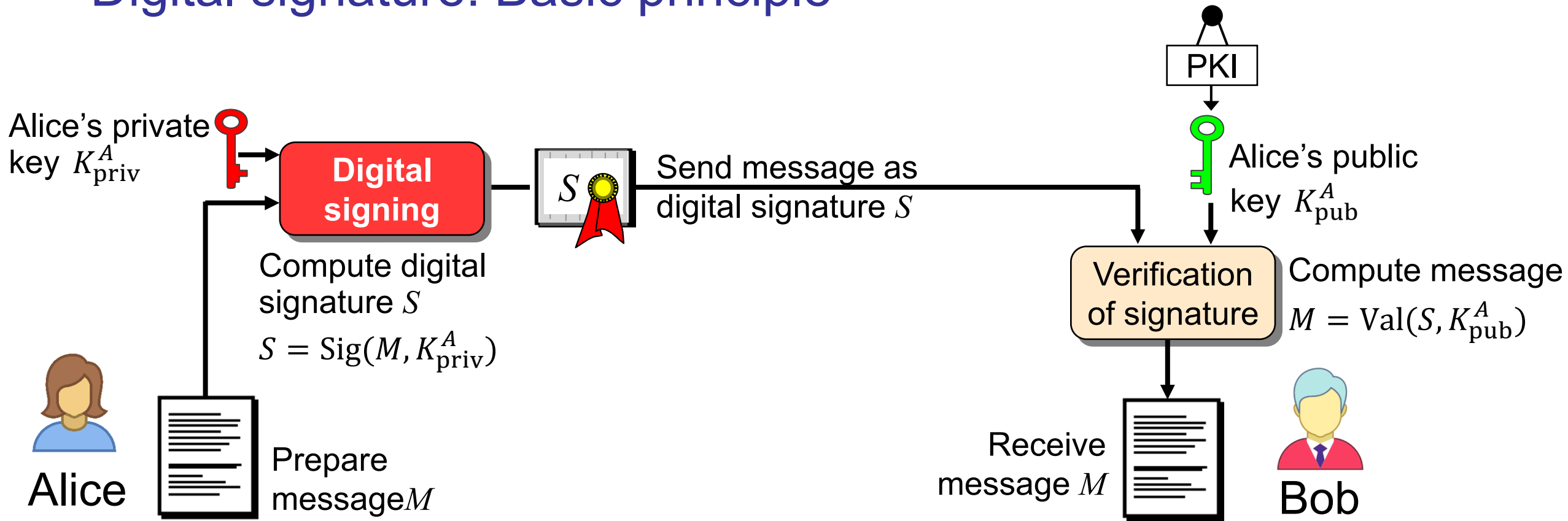
Need for digital signature

- A MAC cannot be used as proof of data authenticity to be verified by a 3rd party
- Digital signatures can be validated by third parties
 - Provides non-repudiation, which means strong data authentication,
- Functions used for digitale signature:
 - Signing (using private key)
 - Verification (using public key)

Simple or strong message authentication

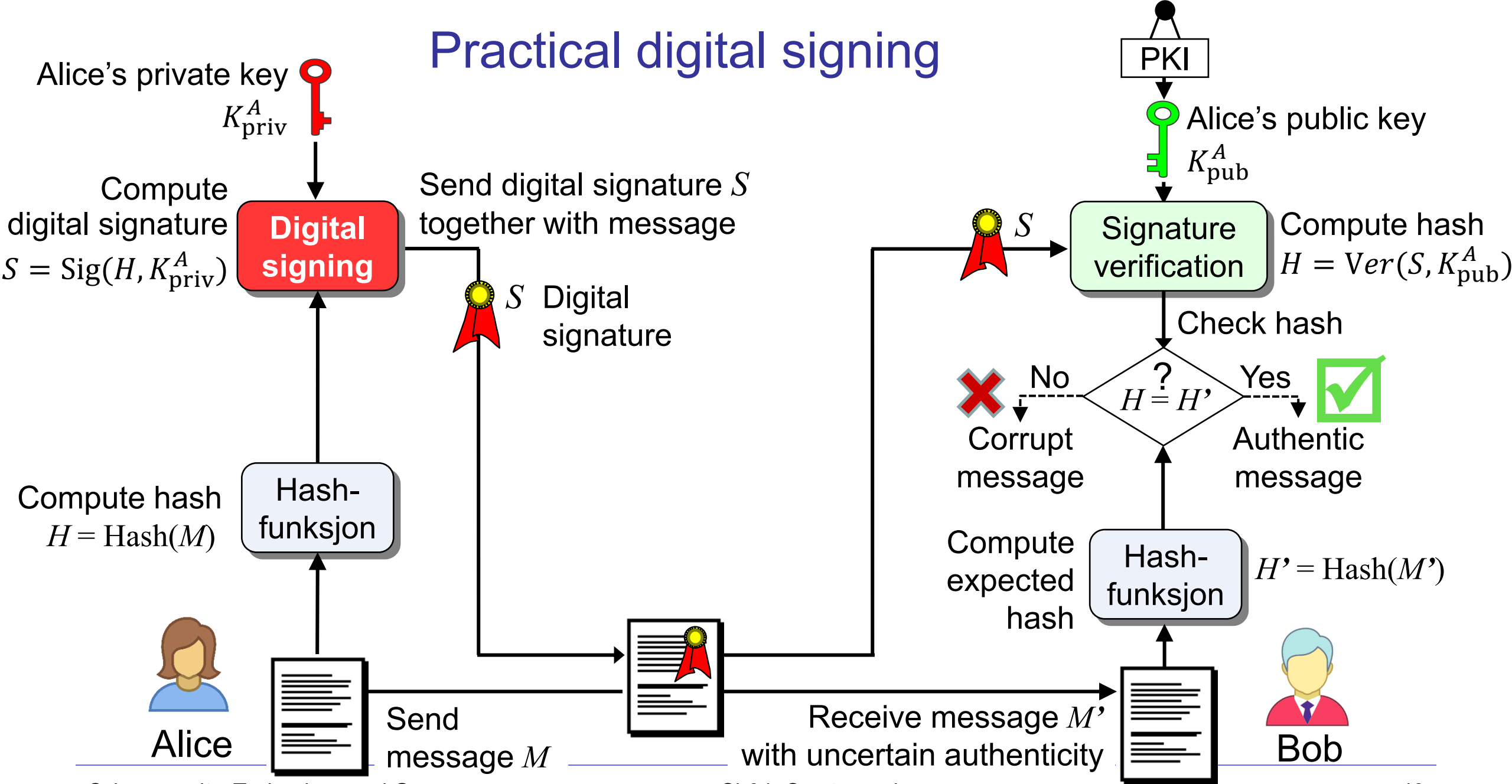


Digital signature: Basic principle

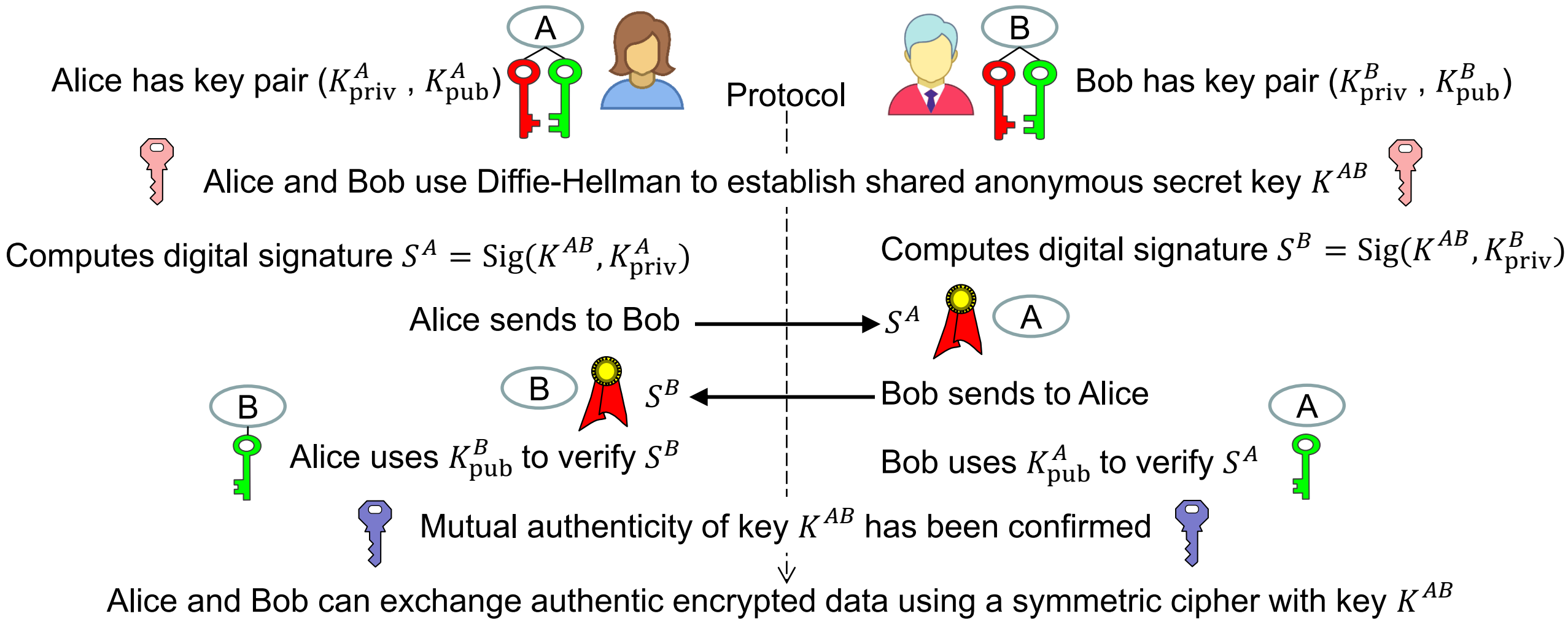


- Digital signing and verification require heavy computation, and are not used for direct signing as shown above.
- In practice, hybrid signing is used, which combines hashing and digital signing.

Practical digital signing

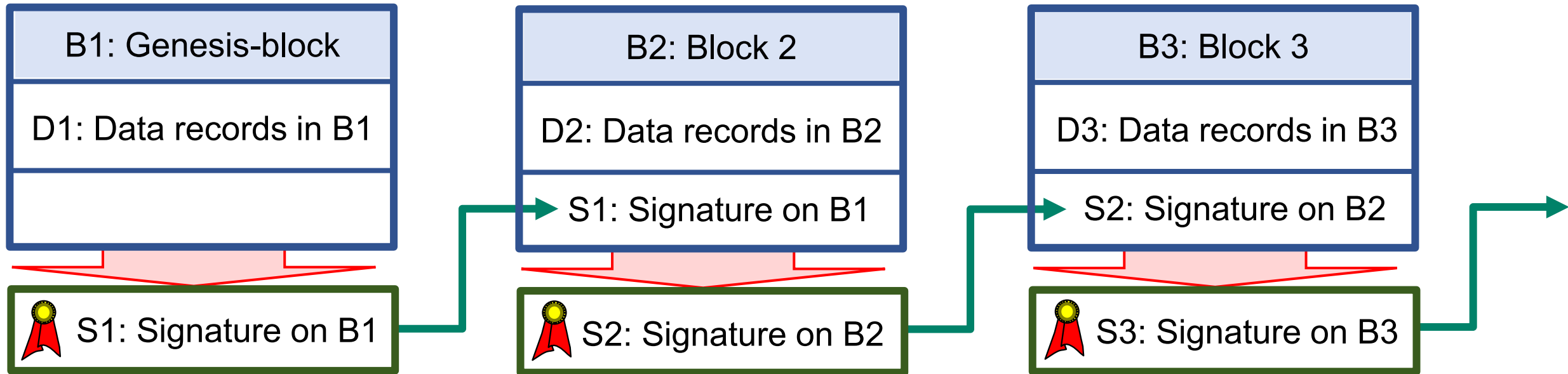


Authenticated key exchange with forward secrecy



Digital signature in blockchains

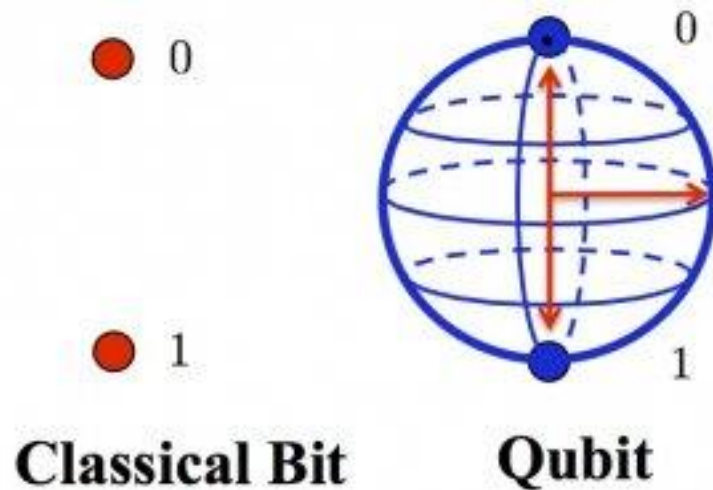
- Each block contains a set of data records (e.g., transactions)
- Signing of each block is done with a digital signature on the hash value of the block
- Crypto miners must find a hash of the next block, where the hash value has a number of trailing zeros
- It takes billions of hash computations to find a hash with many trailing zeros
- The first miner to find such a hash value can sign the next block, and gets a new bitcoin as a reward
- There's a huge global electricity consumption among miners to get each new block signed.



POSTQUANTUM CRYPTOGRAPHY

Quantum computers

- Quantum Computing (QC) is based on quantum "qubits" instead of binary bits

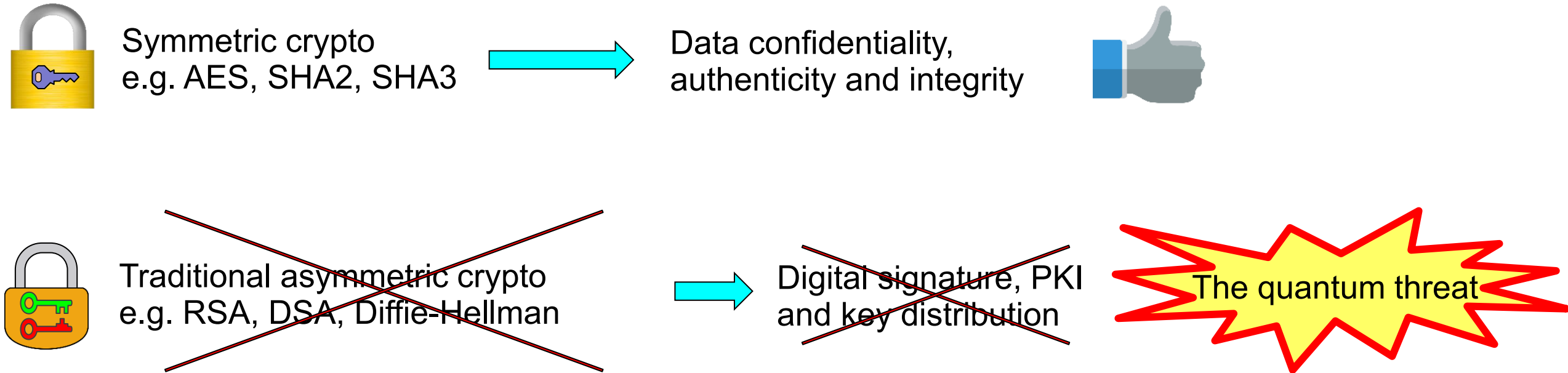


Experimentell
quantum
computer



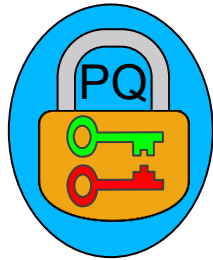
-
- Quantum algorithms, i.e. algorithms for quantum computers, can potentially crack traditional asymmetric crypto algorithms, e.g. RSA, DSA and Diffie-Hellmann.

How quantum computers can affect crypto algorithms



- Symmetric crypto is little affected by quantum computers.
- Traditional asymmetric crypto can in theory be cracked by (super-powerful) quantum computers.

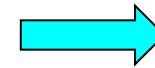
Asymmetric crypto algorithms that cannot be cracked by quantum computers



Post-Quantum asymmetric encryption & digital signature
e.g. with Lattice-based, Multivariate, Hash-based algorithms

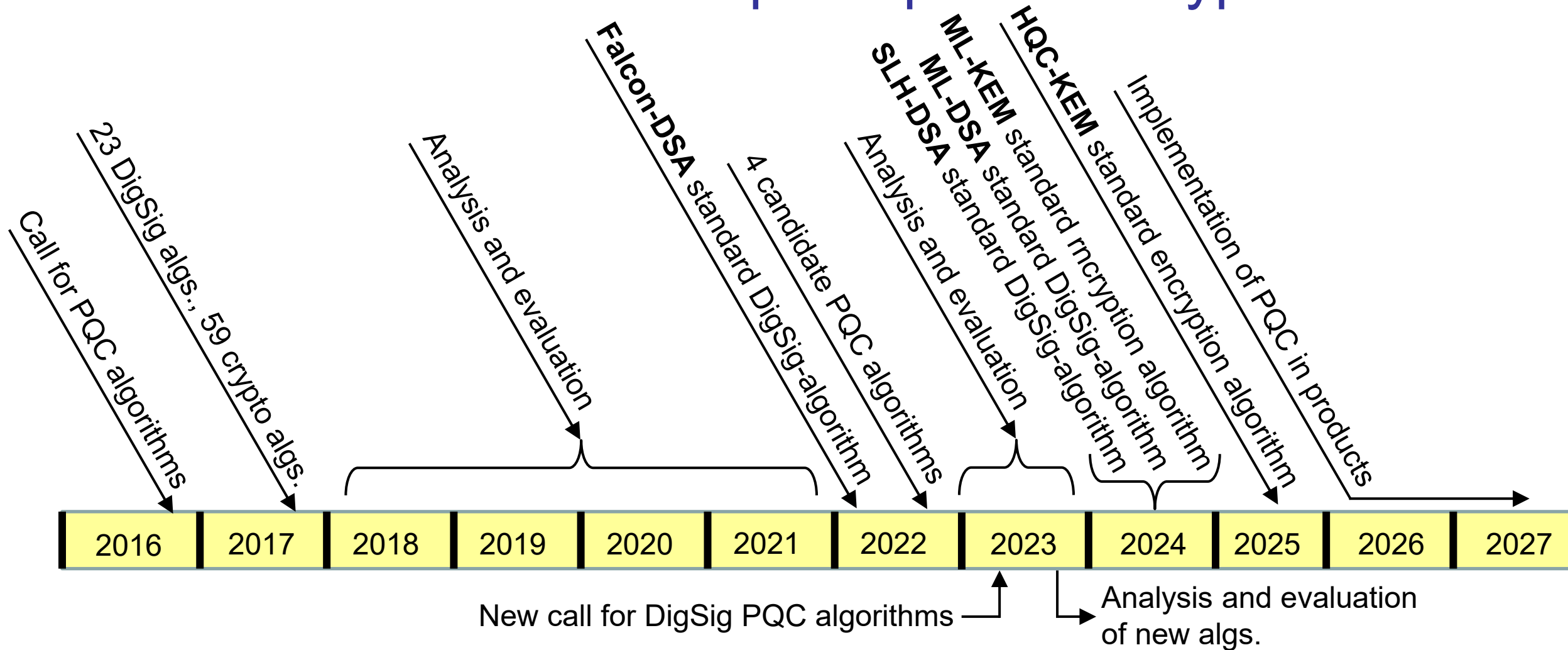


With post-quantum crypto, we can keep DigSig and PKI even with quantum computers that have capacity more than 1 million qubits.



Digital signature
PKI/key distribution

Standardization of post-quantum crypto



- The term "Post-Quantum Crypto" means cryptography that cannot be cracked by quantum computers, also called quantum-resistant algorithms.

Standard PQC-algoritmer, august 2025

FIPS: Federal Information Processing Standard

ML-KEM: FIPS 203

- **Module-Lattice-Based Key-Encapsulation Mechanism**
- Nickname: «CRYSTALS-Kyber»

HQC-KEM

- **Hamming Quasi-Cyclic Key-Encapsulation Mechanism**

Falcon-DSA

- **Digital Signature Algorithm**

ML-DSA: FIPS 204

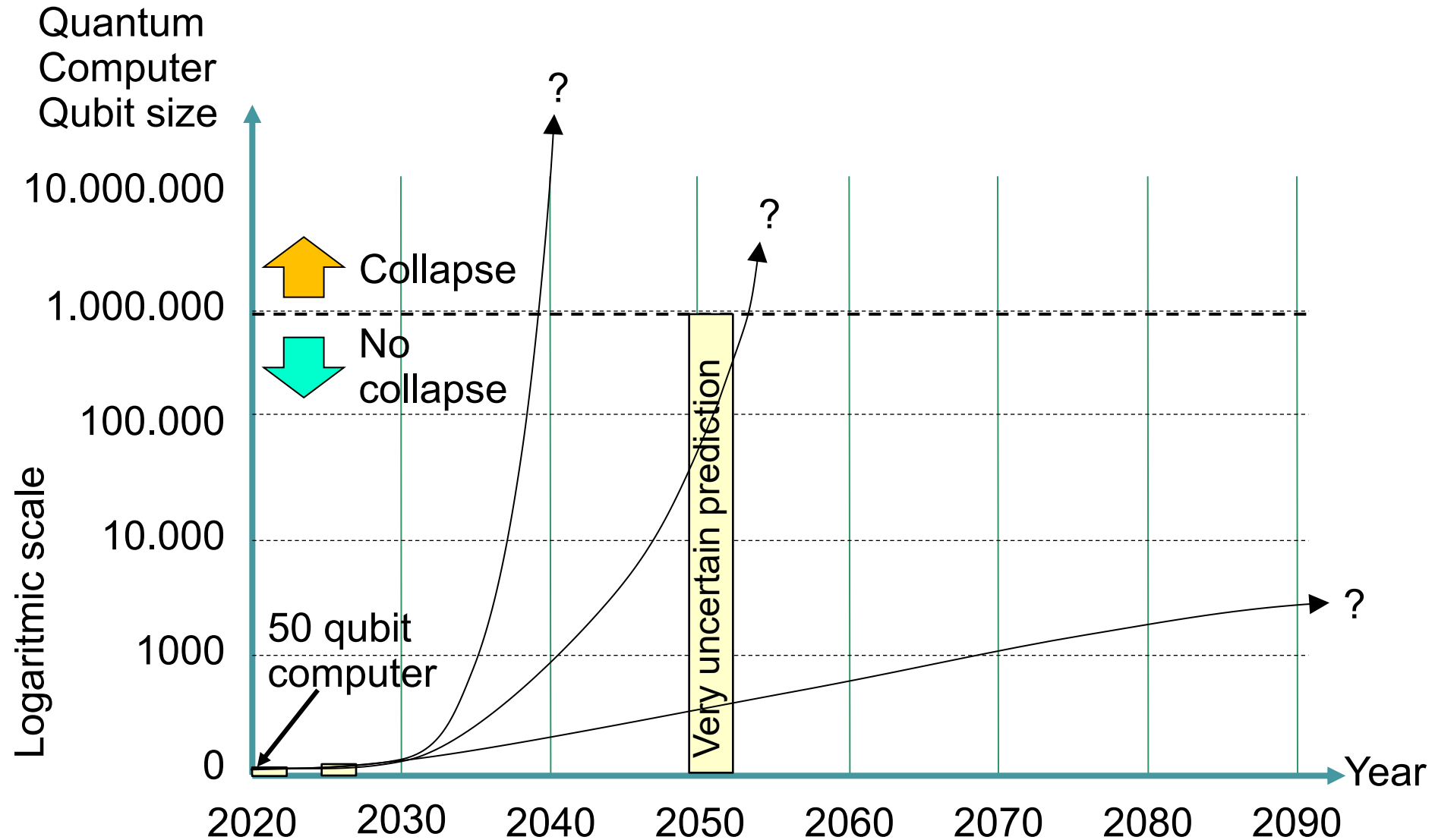
- **Module-Lattice-Based Digital Signature Algorithm**
- Nickname «CRYSTALS-Dilithium»

SLH-DSA: FIPS 205

- **StateLess Hash-Based Digital Signature Algorithm**
- Nickname: «SPHINCS+»

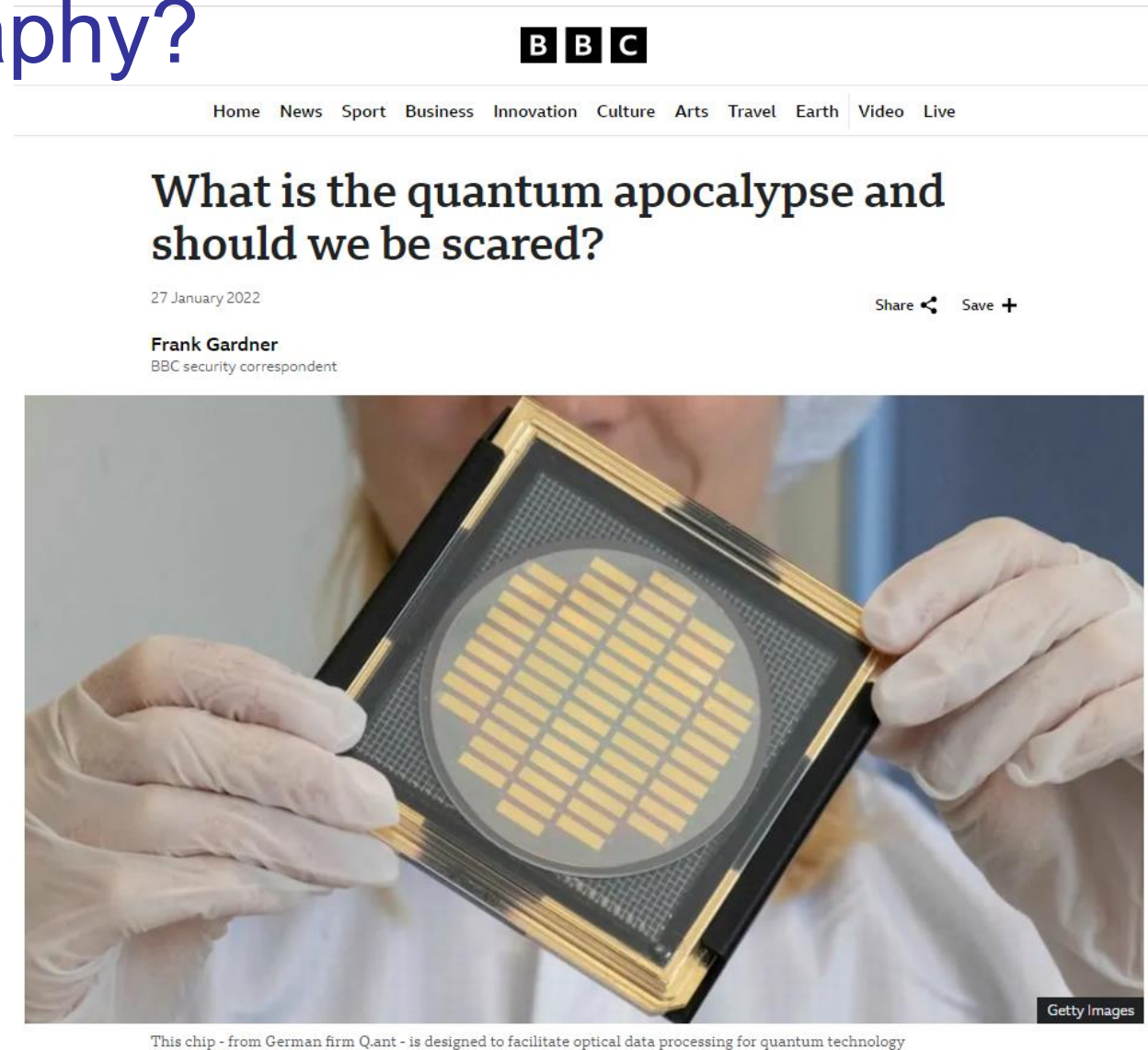
NIST wants to standardize new DigSig algorithms that are not based on lattice or hash. This is to have variation in the mathematical basis in the case of a mathematical breakthrough which could break all PQC algorithms with the same mathematics.

Collapse of traditionally asymmetric crypto?



Why postquantum cryptography?

- I. Use post-quantum crypto because quantum computers are likely to break RSA, Diffi-Hellmann, and DSA sometime in the future.
- II. Use post-quantum crypto because you don't want your organization to end up on the front page of news media, accused of being irresponsible.



BBC.com, 27 January 2022

End of presentation

