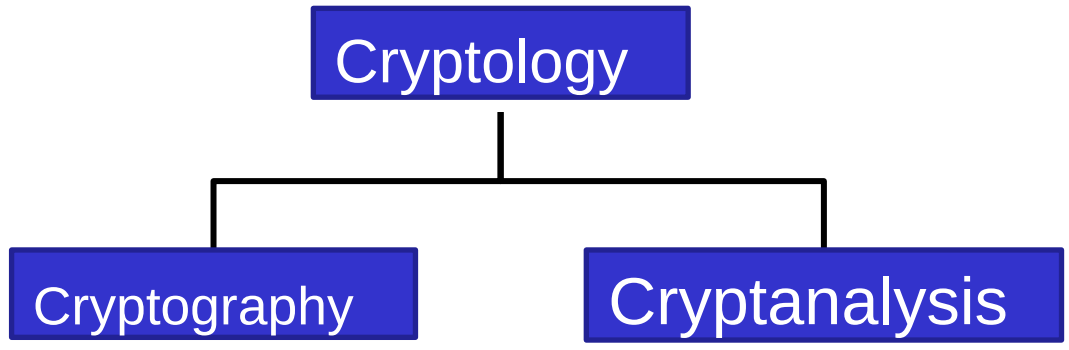
Cryptography

Overview

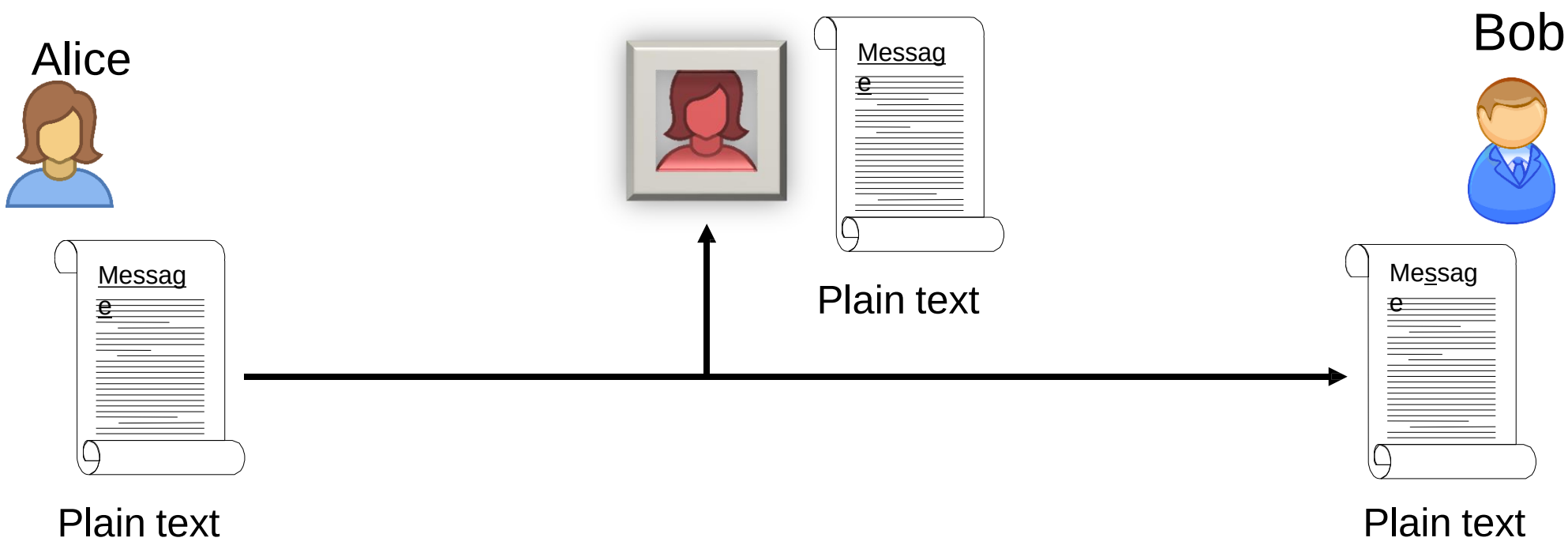
- What is cryptography?
- Symmetric cryptography
  - Brief crypto history
  - Stream cipher
  - Block cipher
  - Hash functions
- Asymmetric cryptography
  - Asymmetric encryption in general
  - Diffie-Hellman key exchange
  - Digital signatures
- Post quantum cryptography

Terminology

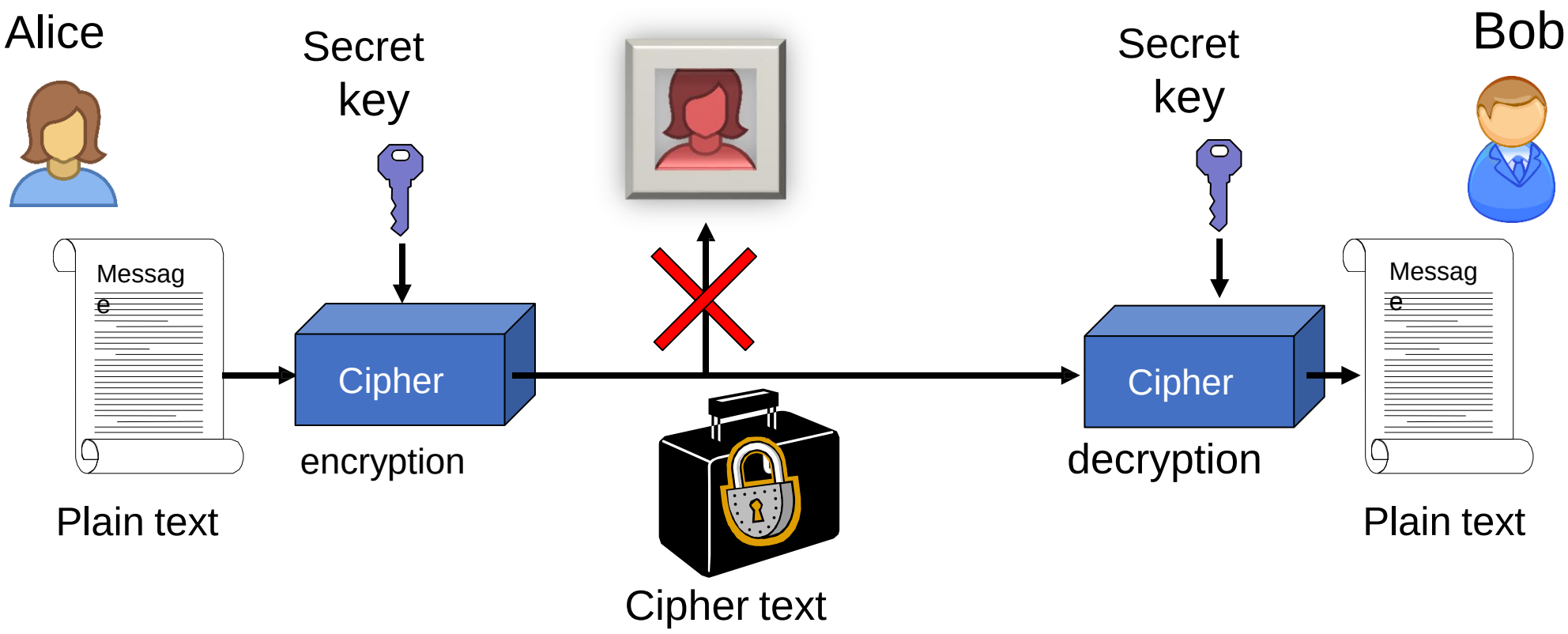


- **Cryptography**is the science of secret writing for the purpose of concealing the meaning of a message.
- **Cryptanalysis**is the science of cracking cryptography.
- **Cryptology**covers both cryptography and cryptanalysis.

Cryptography at a glance



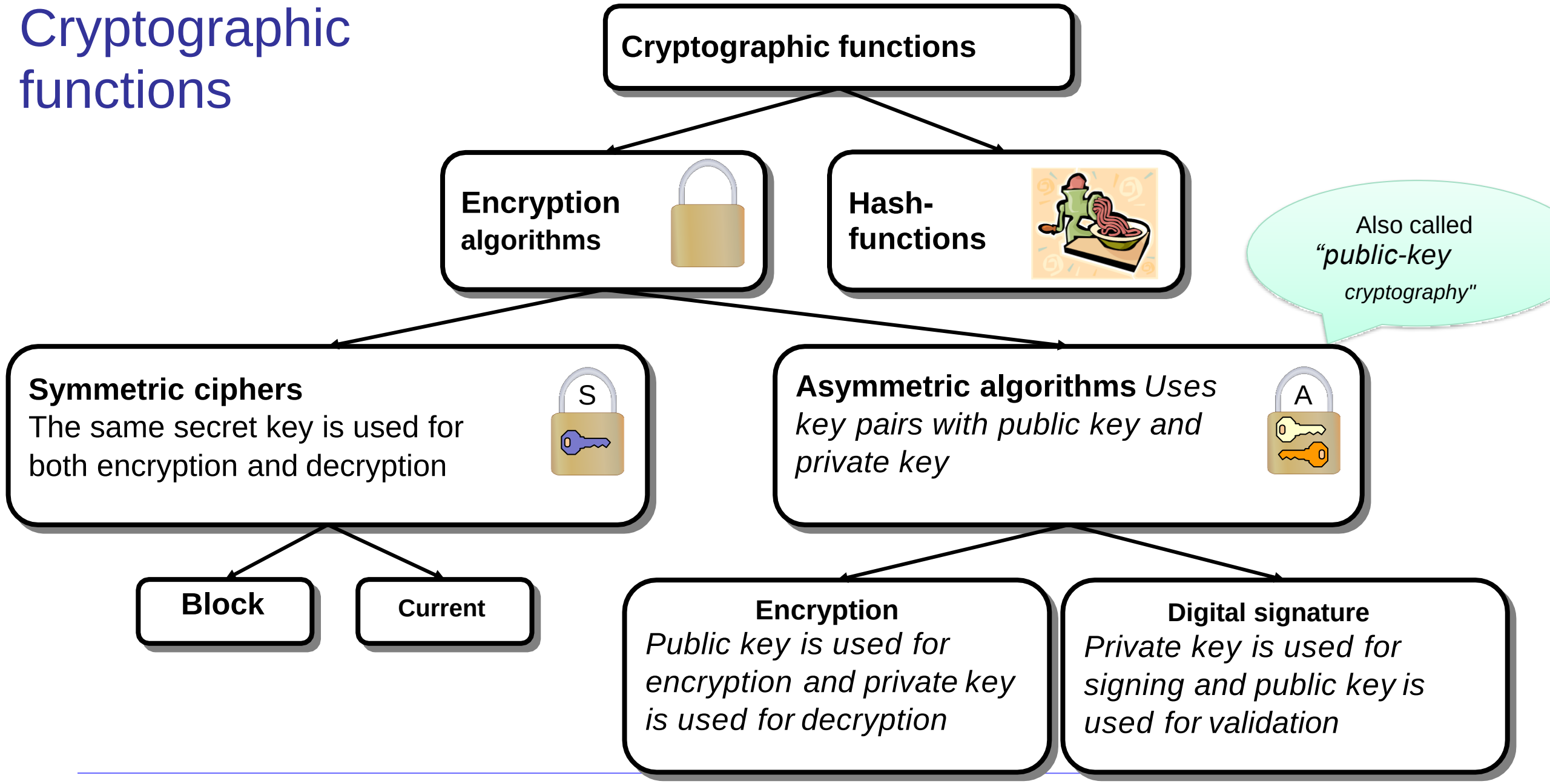
Cryptography at a glance



What can cryptography be used for?

- Cryptography supports the following security goals:
  - **Confidentiality:**
    - Makes data unreadable by devices that do not have the correct cryptographic keys, even if they have the data.
  - **Data integrity:**
    - Devices with correct cryptographic keys can verify that data is correct and has not been altered by unauthorized persons.
  - **Authentication:**
    - Communicating entities can be assured that the identity of the other user/ entity or the sender of a message is what it claims to be.
  - **Digital Signature and PKI (Public-Key Infrastructure):**
    - Strong evidence of data authenticity that can be verified by third parties.
    - Scalable (to the entire Internet) secure distribution of cryptographic keys.

Cryptographic functions



Terminology

- Encryption: plain text *M* is transformed with an encryption function E to ciphertext *C* controlled by encryption key *k*.
  - Formal spelling: *C* =E(*M, k*).
- Decryption: cipher text *C* transformed with decryption function D for plain text *M* controlled by encryption key *k*.
  - Formal disc style: *M* =D(*C*,*k*).
- Symmetric cipher: the same secret key is used for both encryption and decryption.
- 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 validation of signature with public key

Private key 

Public key 

Cryptography| the algebraic prespective

$\mathcal{A}$, the alphabet, is a finite set.

$\mathcal{M} \subseteq \mathcal{A}$ is the message space. $M \in \mathcal{M}$ is a plaintext (message).

$\mathcal{C}$ is the ciphertext space, whose alphabet may differ from $\mathcal{M}$.

$\mathcal{K}$ denotes the key space of keys.

Each $K_e \in \mathcal{K}$ determines a bijective function from $\mathcal{M}$ to $\mathcal{C}$, denoted by E_{Ke}.

E_{Ke} is the encryption function.

For each $K_d \in \mathcal{K}$, D_{Kd} denotes a bijection from $\mathcal{C}$ to $\mathcal{M}$.

D_{Kd} is the decryption function.

- An encryption scheme consists of two sets $\{E_e : k_e \in \mathcal{K}\}$, and $\{D_d : k_d \in \mathcal{K}\}$ verifying

$$\forall k_e \in \mathcal{K}, \exists! k_d \in \mathcal{K} \left| D_{k_d} = E_{k_e}^{-1} \right.$$

- Ciphering equation

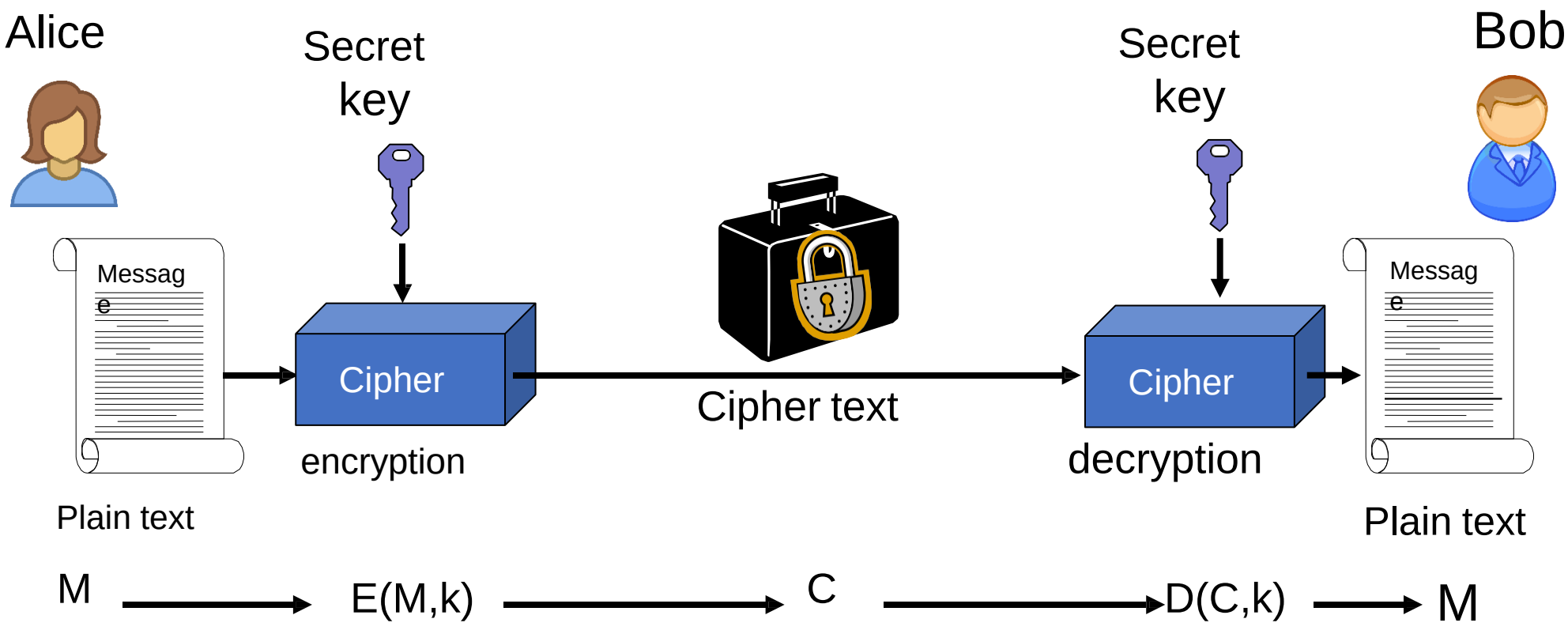
$$c = E_{k_e}(p)$$

- Decryption equation

$$p = D_{k_d}(c)$$

$$p = D_{k_d}(E_{k_e}(p))$$

Symmetric encryption (with secret key)



- "Secret key" means that the key is shared *In secret* between all entities authorized to encrypt/decrypt.

The history of cryptography

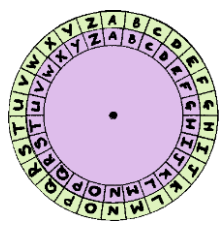
Classic cipher

Transposition

Scythal

Substitution

caesar-cipher



Middle Ages cipher

Poly-alphabetical substitution and transposition

Vigenère

1566

Pre-WW2 cipher

disposable key

Vernam

1916

WW2 cipher

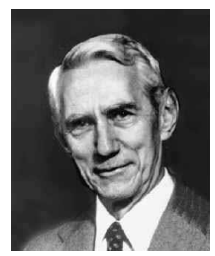
Complex mechanics

Enigma

SP network

Info theory

Shannon



Pre-2000 cipher

DEC

Feistel

Asymmetrical crypto

Diffie

Hellmann

Post-2000 cipher

AES

Rhyme & Daemen

SHA-3

Post quantum asymmetrical crypto

→ BC | A.D→1799 | 1800→1939 | 1940→1975 | 1976→2000 | 2001→

Caesar Cryptosystem

Let P= C= Z_{26}. For 0 <*K*<25, *we define*

$$E(x, K) = x + K \pmod{26}$$

and

$$D(y, K) = y - K \pmod{26}$$

($x, y \in Z_{26}$)

Substitution ciphers

- Substituting a character (or symbol) for each character of the plaintext message.
- Example (Caesar Cipher):

  Shift alphabet three (3) places: A →D, B → E,...

Key	3
Plaintext	D A T A I S C I P H E R E D
Ciphertext	G D W D L V F L S K H U H G

- A very simple cipher

Key	123
Plaintext	D A T A I S C I P H R E D
Ciphertext	E C W B K W D K S I T H E+

Transposition ciphers

- Rearranging the characters of a message.

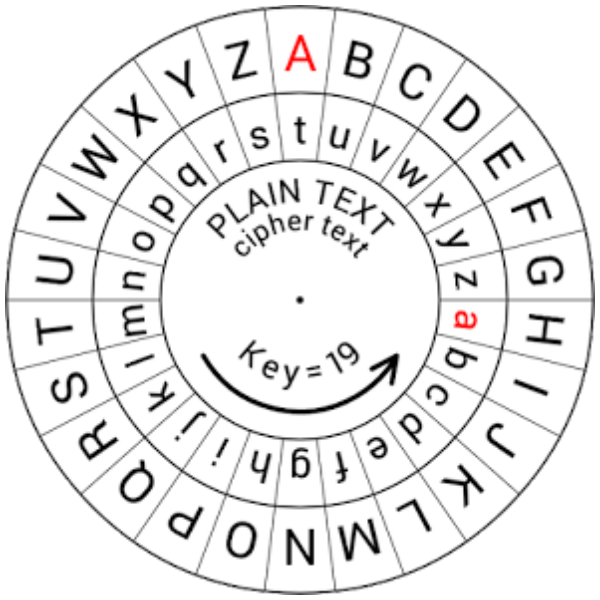
Key	3
Plaintext	D A T A I S C I P H E R E D
Transposed	D A C H E A I I E D T S P R
Ciphertext	D A C H E A I I E D T S P R

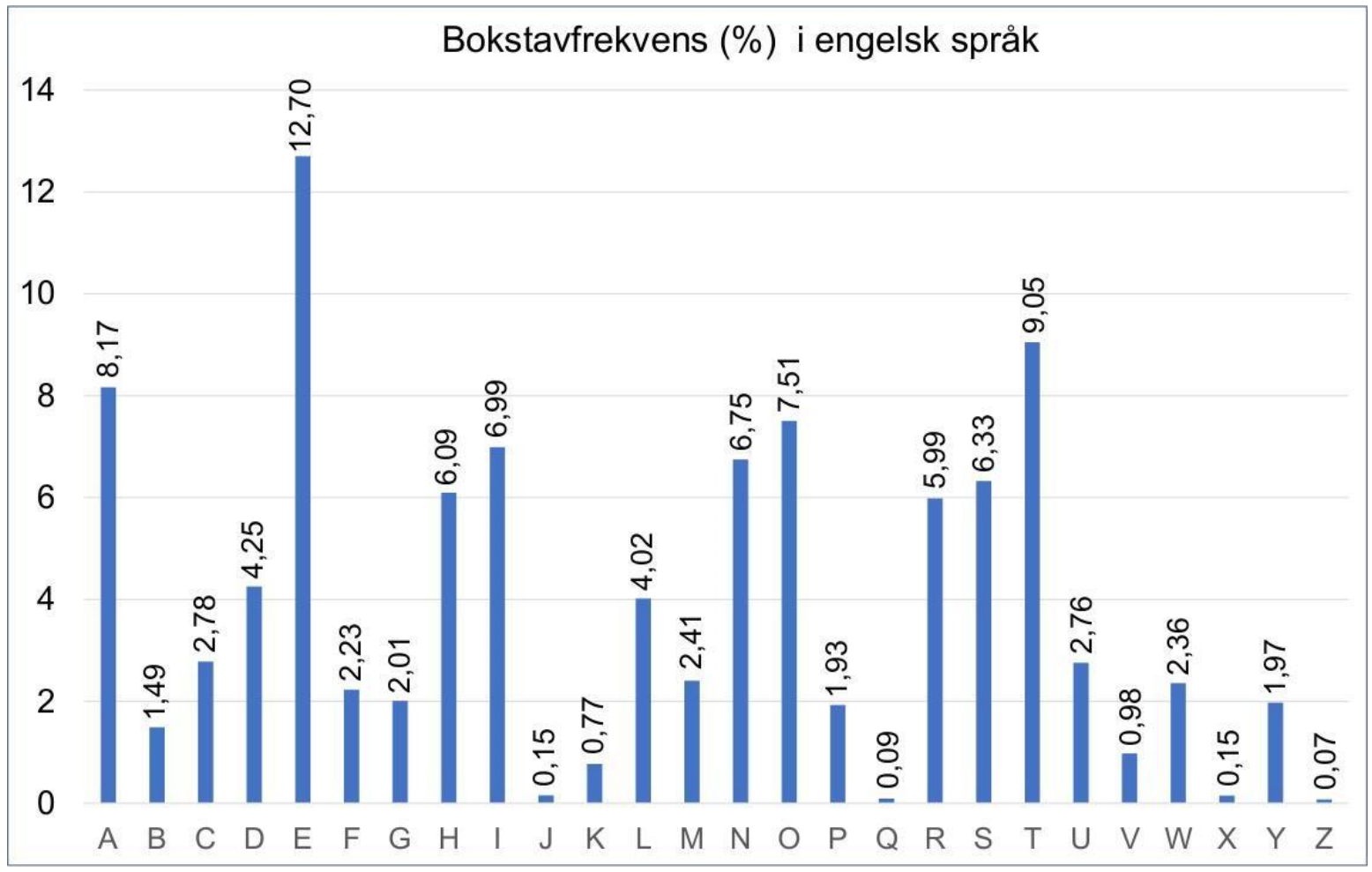
Letter frequencies→ Statistical cryptanalysis

- Classical ciphers, such as the Caesar cipher, are weak because they fail to hide statistical ones

  irregularities in the cipher text.

Caesar chi





Designing a Crypto-System

Kerckhoffs' Design Principles (ca. 1883)

1. The system must be practically, if not mathematically, indecipherable;
2. It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience;
3. Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents;
4. It must be applicable to telegraphic correspondence;
5. It must be portable, and its usage and function must not require the concourse of several people;
6. Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

Crypto system design

- Kerchoff's Law:

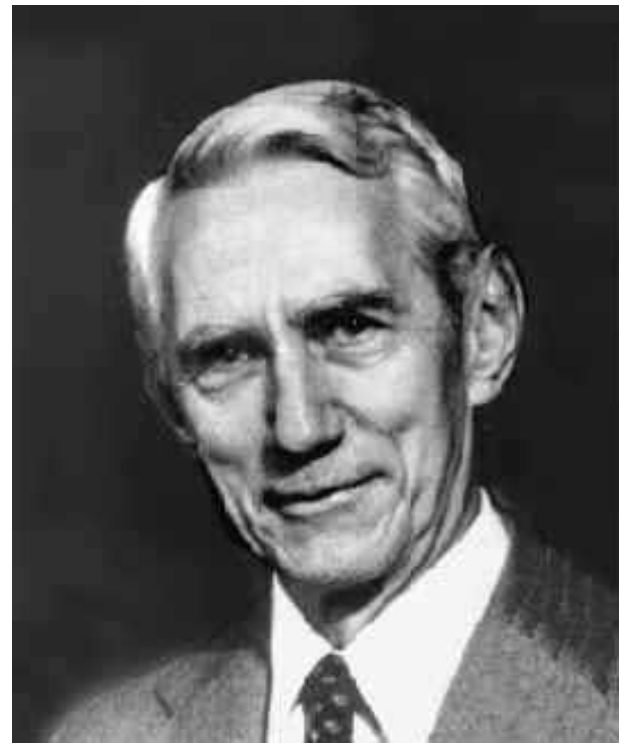
  Attacker always knows encryption algorithm: **E** ( )

➔Algorithm is public, the key is private.

- The attack mounted will depend on what information is available to the adversary :
  - Ciphertext
  - plaintext

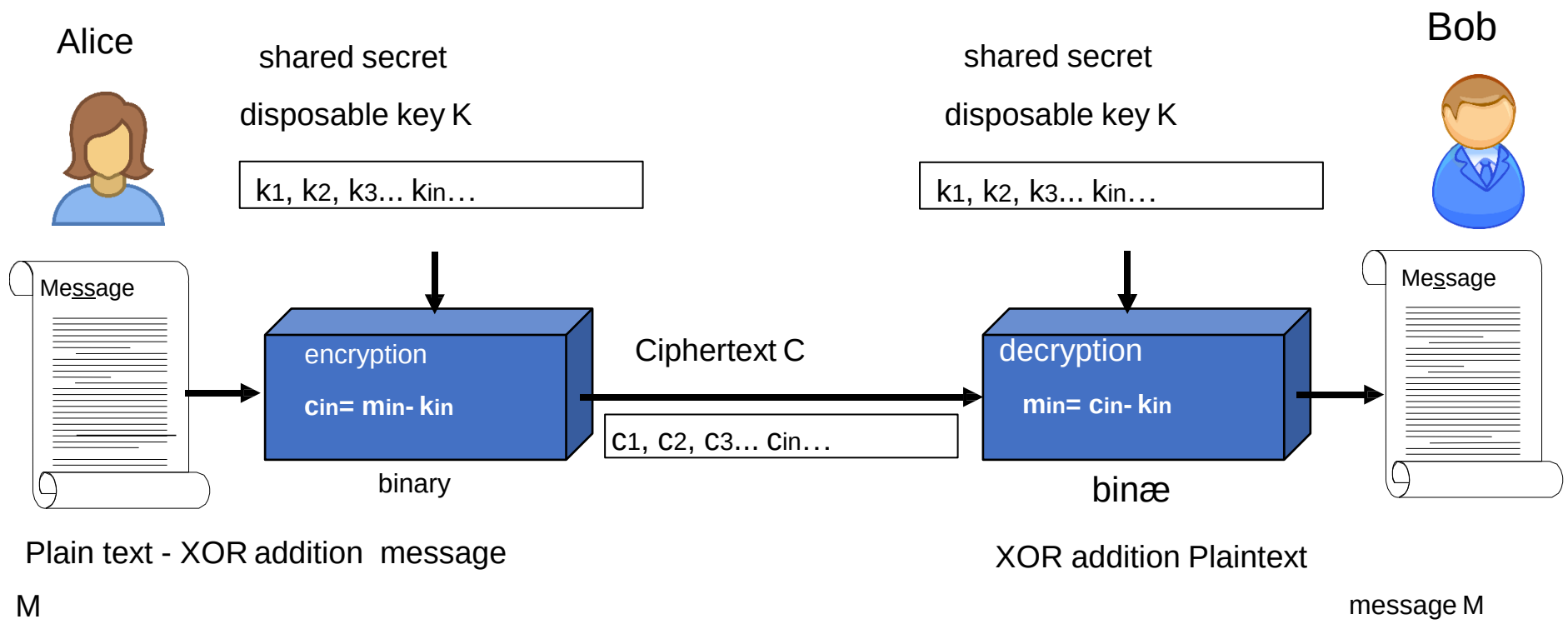
Claude Shannon (1916 – 2001)

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 secret secure systems
  - – Defined perfect secrecy security
  - – Principle of encryption with SP network (substitution and permutation) to erase statistical irregularities

One-Time Pad: One-Time Pad (Gilbert Vernam, 1917)



- Bit by bit binary XOR addition:

$c_{in} = m_{in} - k_{in}$

$m_{in} = c_{in} - k_{in} = m_{in} - k_{in} - k_{in} = m_{in} - 0 = m_{in}$

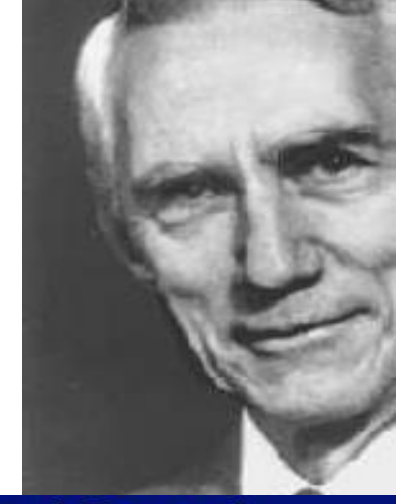
- OTP provides perfect security by assuming that the OTP key is completely random, of the same length as the message, and is used only once.

The perfect cipher machine: One-Time-Pad

- Telex with OTP on punched tape, produced by STK on Økern
- Modern versions can use DVDs with Gbytes of random data

Does a perfect secure system exist?

Yes, Perfect encryption scheme can exist only if the secret information *k* is as long as the plaintex t [Claude Shannon, 1943]



Time Pad

Key as long as message
Key must be absolutely random keys Key must never be re-used

guarantees perfetc security y Key Key management

very hard
Computer generated random number sequences are lly) not good Enough

The one-time pad (OTP)

- Assume you have a secret bit string s of ler known only to two parties, Alice and Bob
  - Alice sends a message m of length of n to Bob
  - Alice uses the following encryption function to ciphertext bits:

$$\sum_{i=0}^{n} c_i = m_i \oplus k_i$$

    - E.g., XOR the data with the secret bit string
  - An adversary Mallory cannot retrieve any part
- Simple version of the proof of security:
  - Assume for simplicity that value of each bit in k likely, then you have no information to work wi

Modern cryptography

The assumption that the Adversary has unlimited computing resources is abondoned

Encryption, decryption, and the Adversary are modeled by probabilistic algorithm

The running time of the encryption, decrytion, and the Adversary algorithms are asured as functions of a security parameters

The strength of a cipher

Factors that determine cipher strength:

- **Key size**.
  - – Time required for a complete search among all keys depends on size.
  - – Typical size for a symmetric block cipher is 256 bits.
  - - Attacker must try 2 on average$^{256}/2$ different keys to find the right one, which 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 cipher text.
  - 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.

What is a good cipher?

Good Cipher = Confusion + Diffusion

- Confusion: Message not readily recognizable, interceptor unable to predict change in ciphertext from given change in plaintext
- Diffusion: Information from plaintext should be spread all over the ciphertext, change in plaintext implies many changes in ciphertext

→ Product ciphers

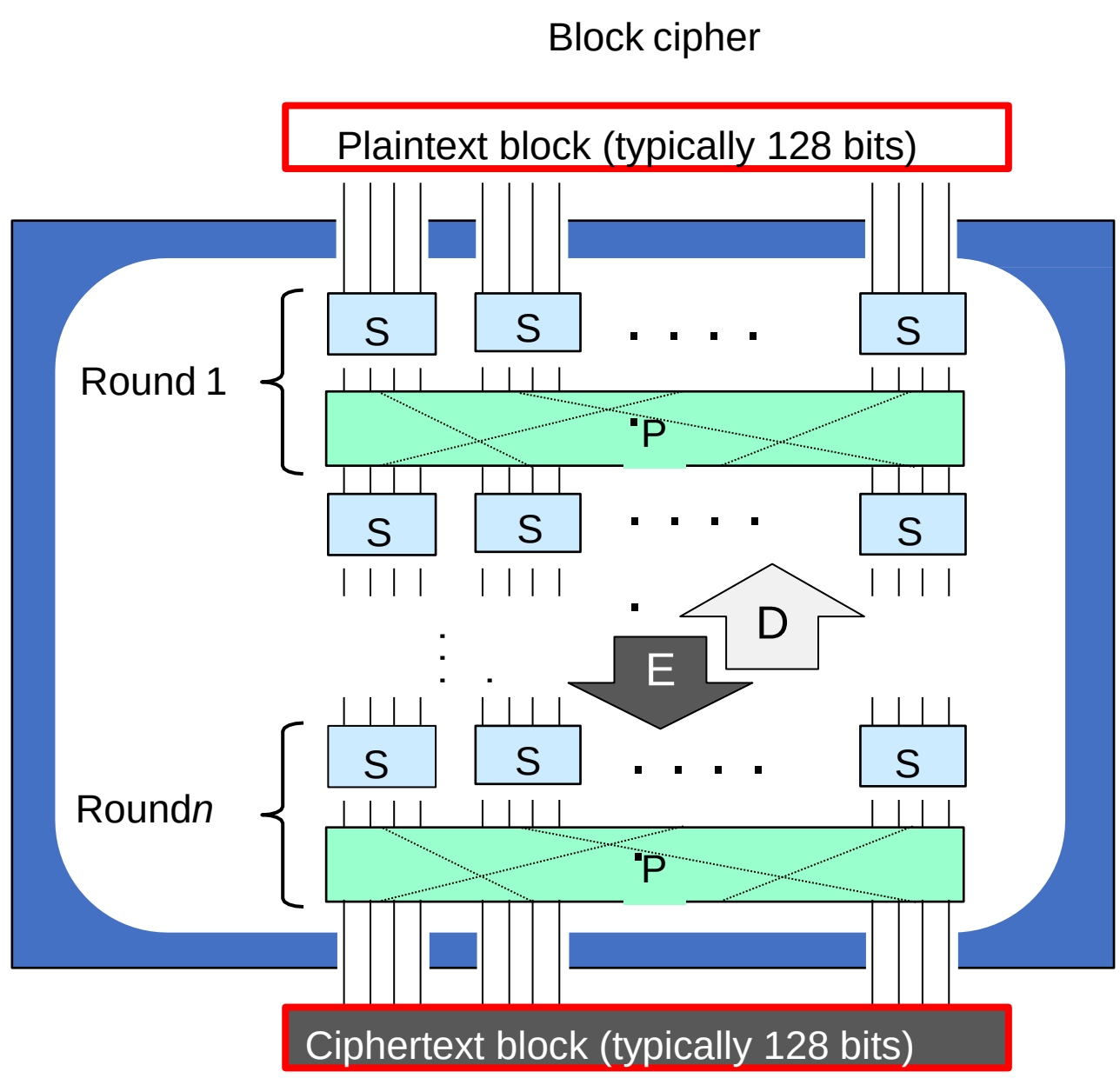
- Substitution ~ Confusion
- Transposition ~ Diffusion

Shannon's SP network (1949)

Erases statistical irregularities

- Repeat substitution and permutation et
  sufficient number of times, typically 10-20 rounds.
- Substitution
  - Clear text block is divided into sub-blocks
  – Substitution of bits in each sub-block e.g. 0001 is substituted with 0110
  - Gives "confusion", i.e. hides the connection between the plaintext block and the ciphertext block.
- Permutation
  – Sub-blocks are moved around the block.
  – Provides “diffusion”, i.e. that changing a single plaintext bit (or key bit) causes many ciphertext bits to change.

•The key is included in S or P or in a separate function



The Data Encryption Standard (DES)

- Created by IBM (and NSA) in the 1970's
- A product cipher using substitution and permutation
- 16 cycles (repetitions)
- Keys are 64 bit (only 56 bits are used)
- Works on words of size 64 bit
- Uses only standard arithmetic and logical operations
  - Easy to implement efficiently
- Superseded by the AES (Rijndael)

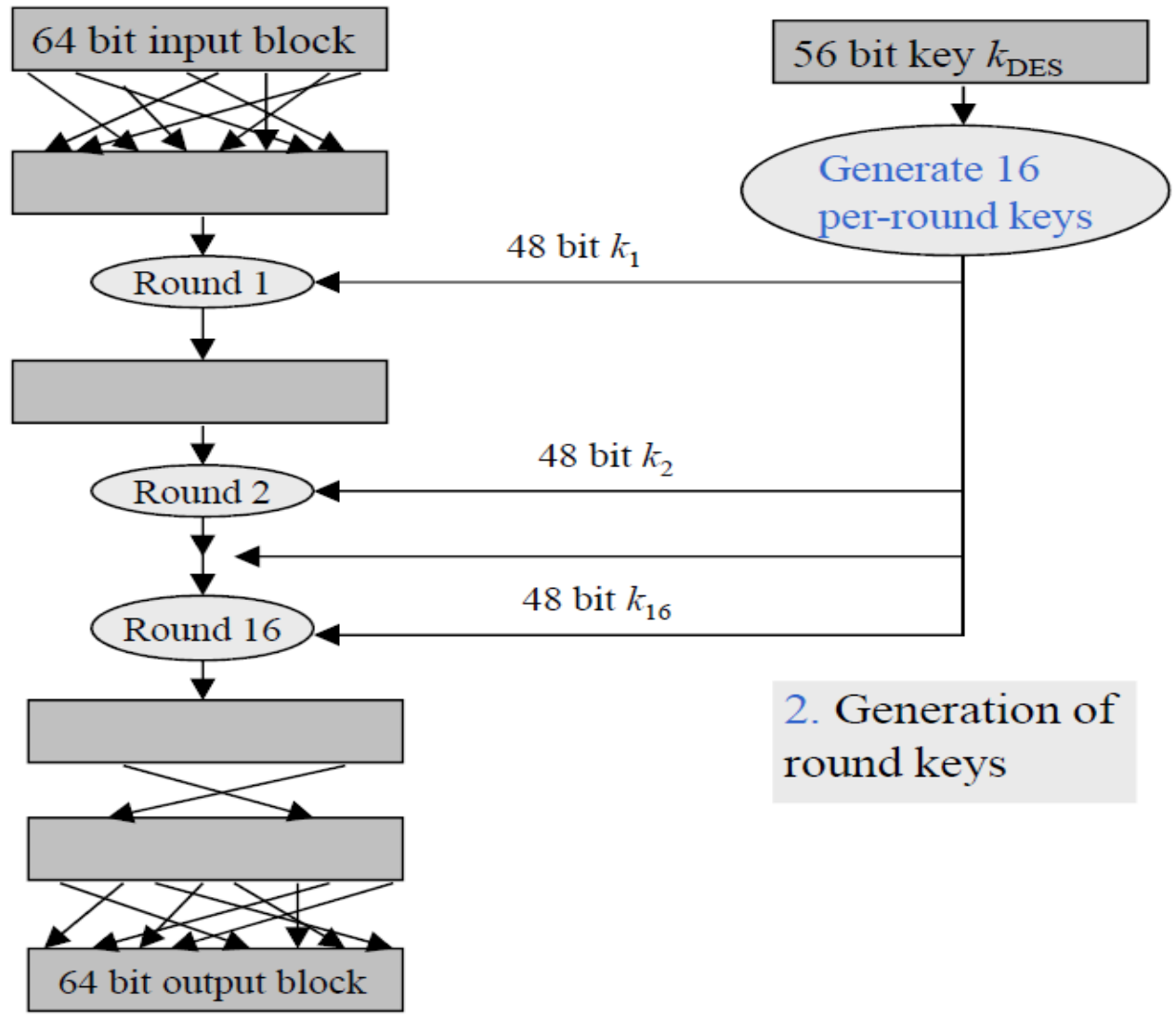
Data Encryption Standard (DES)

1. Initial permutation

3. Encryption in 16 identical rounds:

Additional step: swap Left and right halves

4. Final permutation

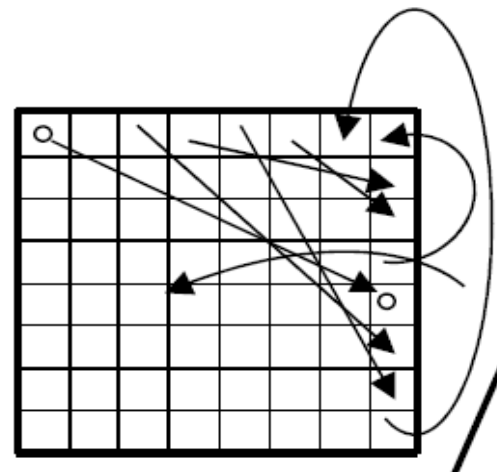


2. Generation of round keys

Step 1 and 4: input and output permutations

Input permutation (IP)

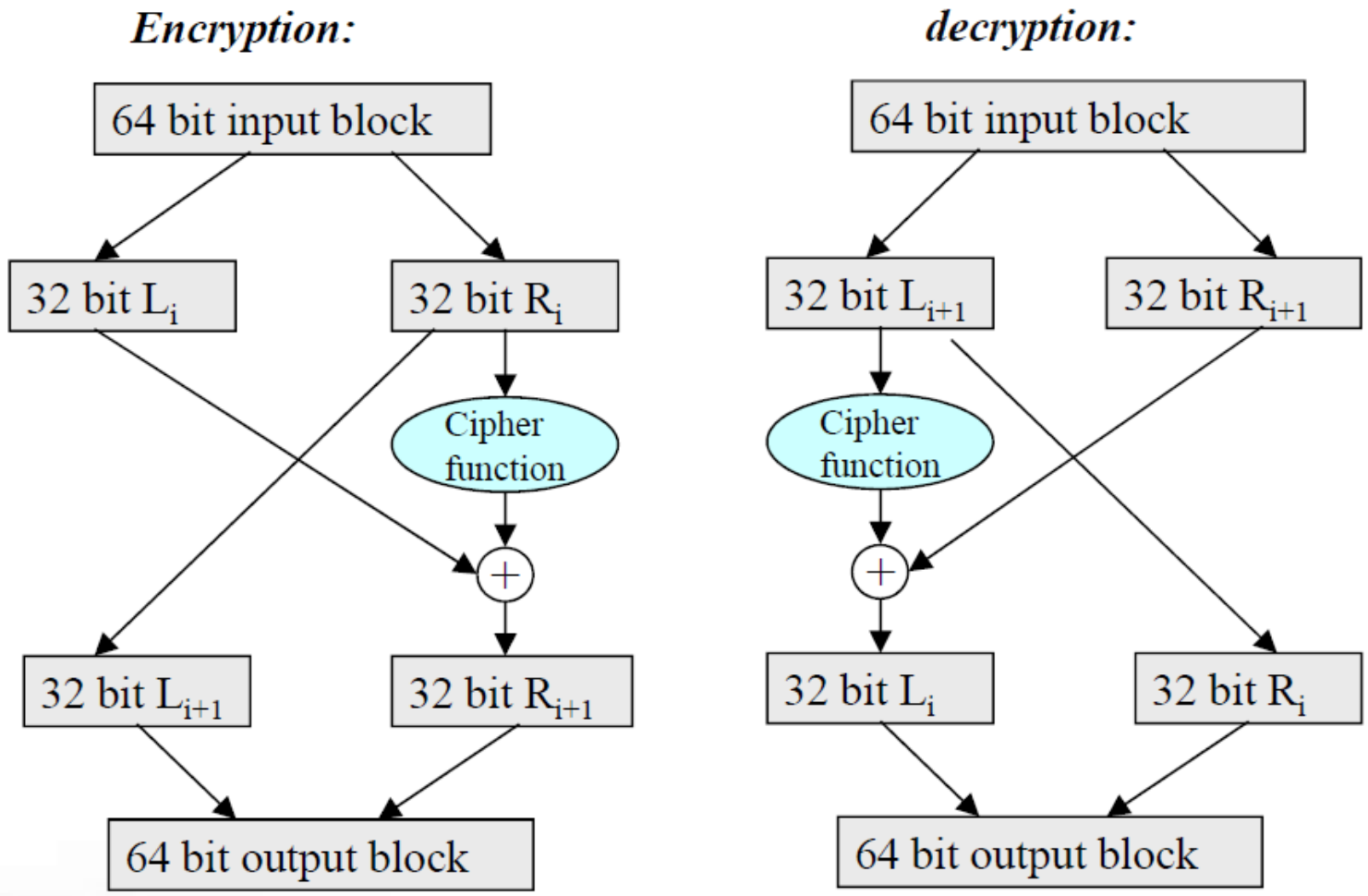
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7



Output permutation (IP^{-1})

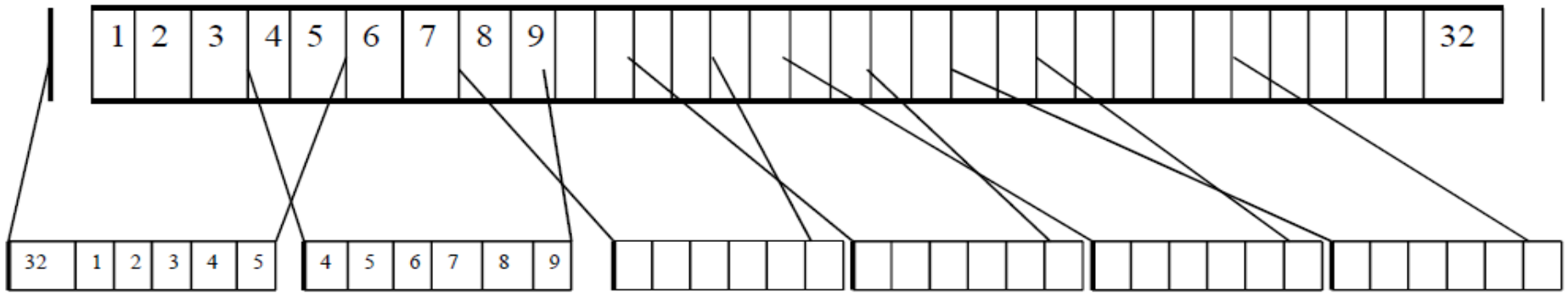
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Step 3 : one DES round



Step 3: cipher function - expansion

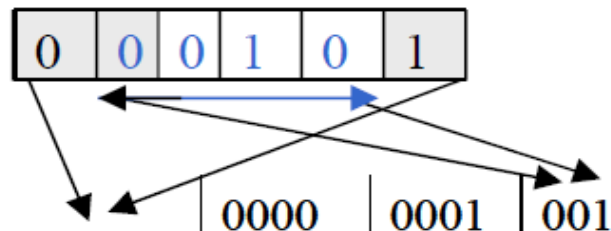
- Expand 32-bit input block Ri to a 48 input block Ri'
  - Divide 32-bit input block into 8 chunks of 4 bit
  - Expansion: enhance segments of 4-bit with their neighbour bits to 6 bit



Step 3 : cipher function: S-Boxes for substitution

ubstitution box (or S-box) is
d to obscure the relationship
ween the plaintext and the
hertext:

- Shannon's property of confusion: the relationship between
- key and ciphertext is as complex as possible.
- In DES S-boxes are carefully chosen to resist cryptanalysis.
- Thus, that is where the security comes from.



	0000	0001	0010	0011	0100	.	1111
00	1110	0100	1101	0001	0010	.	0111
01	0000	1111	0111	0100	1110	.	1000
10	0100	0001	1110	1000	1101	.	0000
11	1111	1100	1000	0010	0100	.	1101

Outputs of S-Boxes are combined into a 32 bit b

Weakness of DES

- Complements (one's complement):
- Weak keys: $C = E_K(M)$ and $M = E_K(C)$
- Semiweak keys: same as weak keys, but for specific key pairs
- Key clustering: $E_{K1}(M) = C = E_{K2}(M)$

- Attacking DES
  - possible to brute force (1998: 112 hours, $130.000)

DES variations

Double DES:

- Use 2 keys: K1 and K2.
- Encryption is $E_{K1}(E_{K2}(P))$
- Is double DES reducible to DES?
- MIM: 2^{57} alternatives

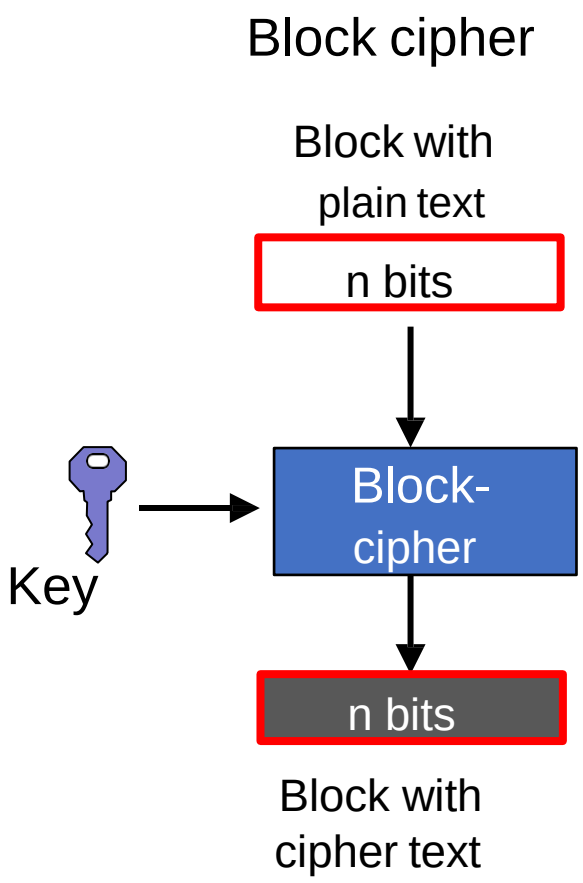
Triple DES

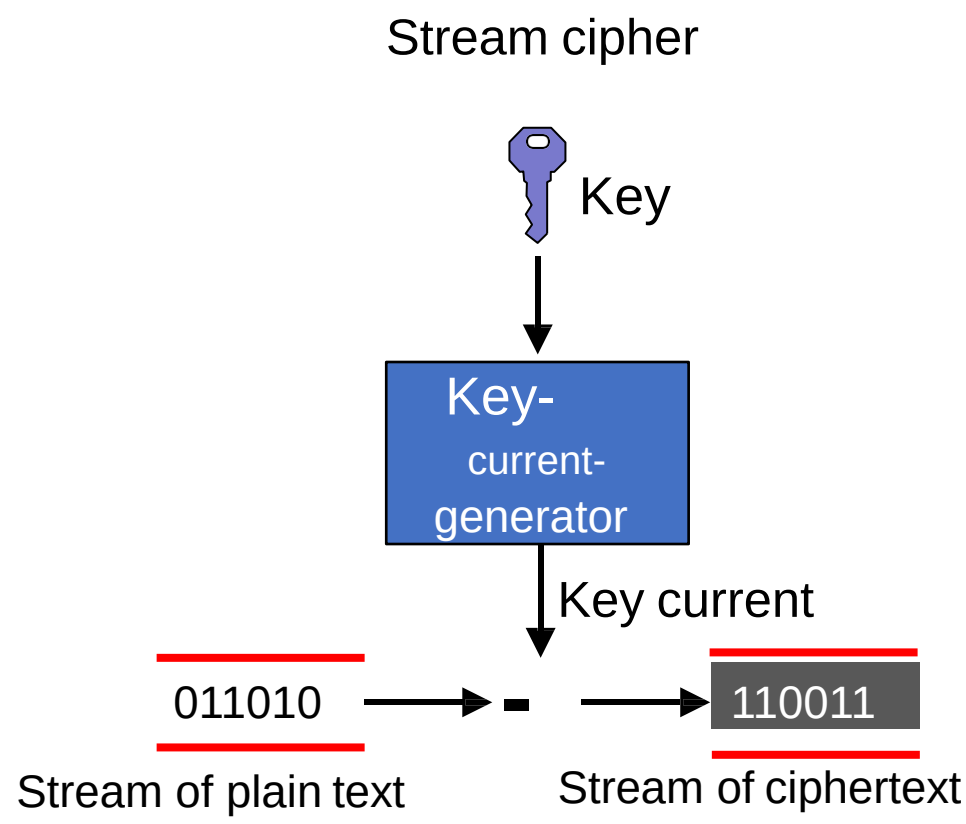
- Use 2 or 3 keys
- Encryption:
  - $E_{K1}(E_{K2}(E_{K3}(P)))$
  - $E_{K1}(D_{K2}(E_{K1}(P)))$
- ~2^{112} alternatives

AES - Advanced Encryption Standard

- DES (Data Encryption Standard) from 1977 had a 56-bit key and a 64-bit block. In the mid-1990s, DES could be cracked with full 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.

Block cipher and stream cipher





Block Cipher: Modes of Operation

- A block cipher encrypts a block of (typically) 128 bits, which is only about 16 letters.
- For encryption of more than one block, a specific mode is used.
- The encryption modes have different properties.
- Common modes are:
  - **C**oun**T**e**R**Fashion (CTR)
  - **C**ipher**B**lock**C**haining (CBC)
  - **O**utput**F**oath**B**ack (OFB)
  - **C**ipher**F**oath**B**ack (CFB)

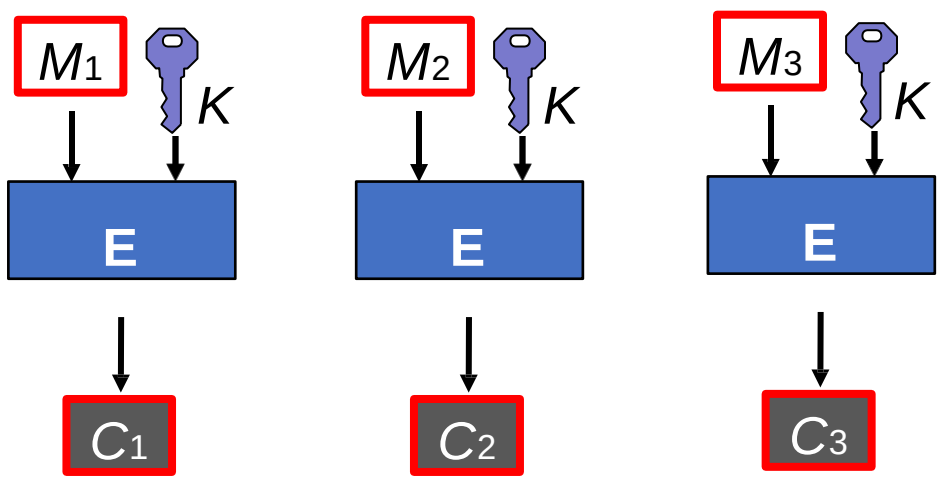
secur e

  - **E**electronic**C**desolate**B**also (ECB)

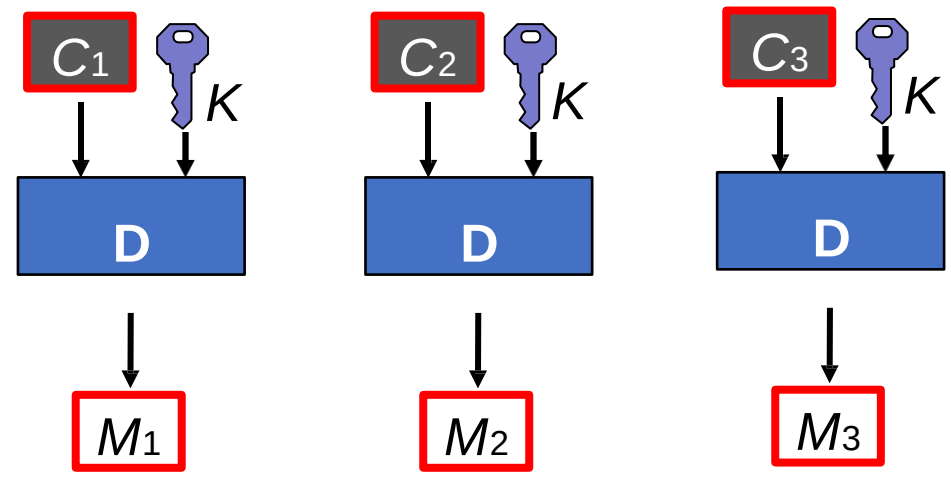
unsure

Electronic Code Book (ECB)

- Simplest encryption mode
- The plain text is divided into blocksM_1, M_2, …, M_n
- Each block is encrypted separately.
  - Notation encryption: $C_1 = E(M_1,K)$
  - Notation decryption: $M_1 = D(C_1,K)$
  - Equal plaintext blocks give equal ciphertext blocks, this is the problem!



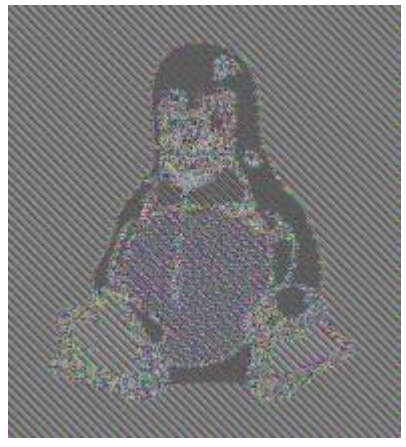
Encryption



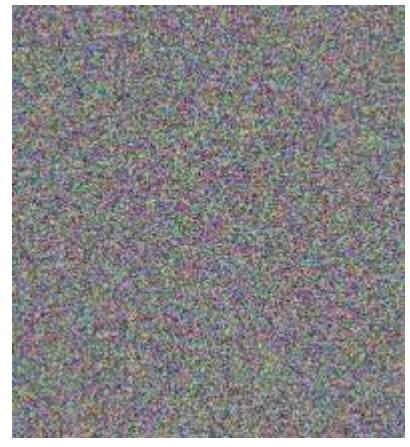
Decryption

Vulnerability using ECB mode

Plain text



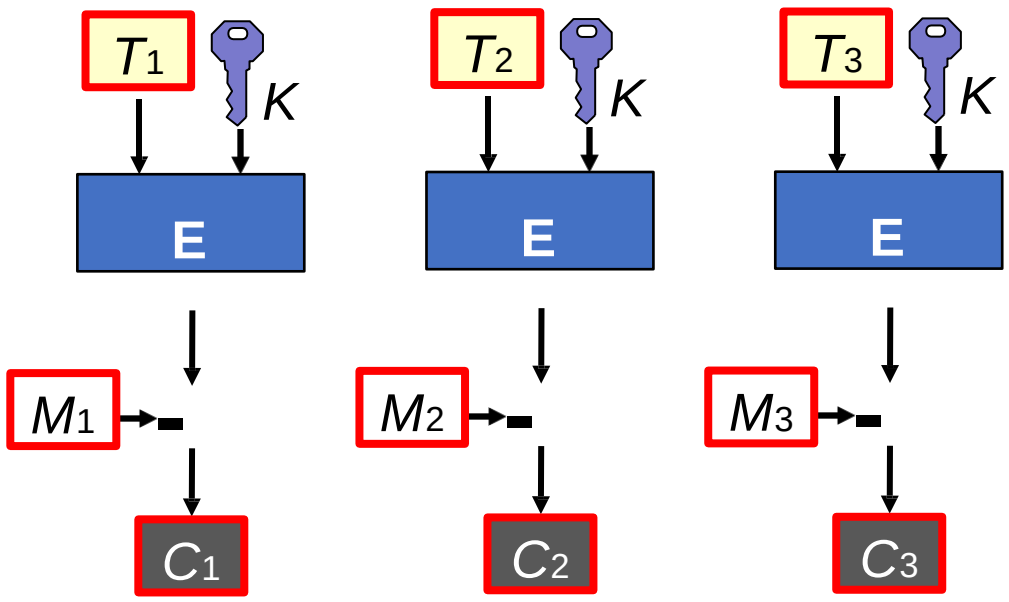
Ciphertext with ECB mode



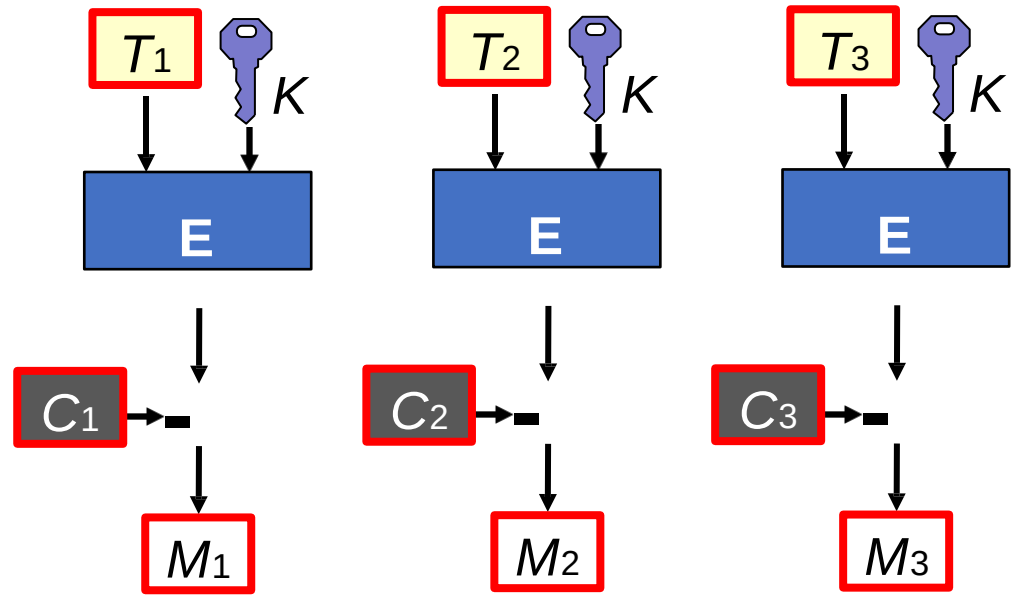
Cipher text with secure mode

Counter Mode (CTR)

- The plain text is divided into blocks$M_1, M_2, \ldots, M_n$
- An incrementing counter value$T$is encrypted
- Each encrypted count value is added to the plaintext block with binary XOR-
  - Identical plaintext blocks give different ciphertext blocks, this provides security!



Encryption



Decryption

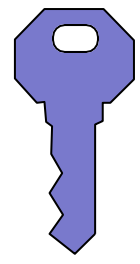
CTR encryption and binary addition with XOR

- The plain text is divided into blocks: $M_1, M_2, \ldots, M_n$
- Incrementing counter values with the same block size: $T_1, T_2, \ldots, T_n$
- The counter values are encrypted and added to the plaintext blocks:
  - Notation encryption: $C_1 = E(T_1,K)\text{-}M_1$
  - Notation decryption: $M_1 = E(T_1,K)\text{-}C_1 = E(T_1,K)\text{-}E(T_1,K)\text{-}M_1$
  - The encryption function **E** used for both encryption and decryption
- Binary addition with XOR-
  - Addition of bit with itself always gives zero

0 - 0 = 0	0 - 1 = 1
1 - 1 = 0	1 - 0 = 1

- Example encryption and decryption: $M_1 = 1111$ $E(T_1,K) = 1001$
  - Encryption: $C_1 = 1001\text{-}1111 = 0110$
  - Decryption: $M_1 = 1001\text{-}0110 = 1111$

HASH FUNCTIONS AND MESSAGE AUTHENTICATION

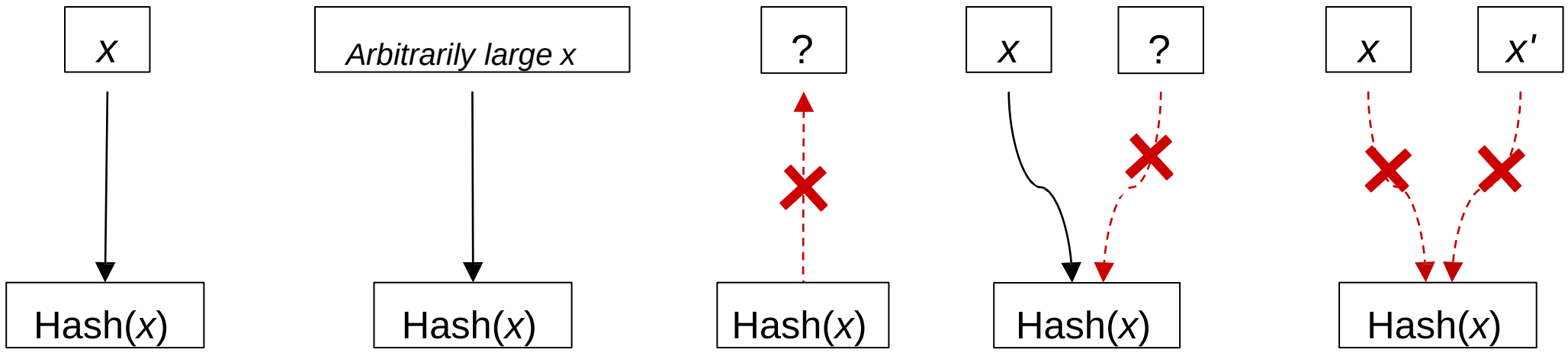


Hash functions

Requirements for a hash functionHash:

1. Easy to calculate: Given input data$x$, it should be easy to calculateHash(x).
2. Compression: Compresses arbitrarily large$x$to a hash valueHash(x) with fixed size$n$(typically 256 bits or 512 bits).
3. One way: Given hash value$y$, it should be practically impossible to find input data$x$so thatHash(x)=y.
4. Collision resistance (weak): Given input data$x$and associated hash value Hash(x),it should be practically impossible to find another data setx'so thatHash(x)=Hash(x')(weak collision resistance).
5. Collision resistance (strong): It should be practically impossible to find two different data sets$x$andx'so thatHash(x)=Hash(x')(strong collision resistance).

Properties of hash functions



Easy to calculate

Compression to fixed size

one-way function

Weak collision resistance

Strong collision resistance

Well known hash functions

- **MD5**(1991): 128-bit hash value. Easy to find collisions, due to small hash size and poor design. Should no longer be used.

- **SHA-1**(Secure Hash Algorithm):160 bit hash value. Designed by NSA in 1995. Relatively easy to find collisions. Should no longer be used, but occurs in still older applications.

- **SHA-2**designed by NSA in 2001. Can generate 256, 384 and 512 bit hash value. Considered safe. Replacement for SHA-1.

- **SHA-3:**designed by Joan Daemen + others in 2010. Standardized in 2015. Can generate: 256, 384, and 512 bit hash value. SHA-3 has little use, because SHA-2 is still considered secure.

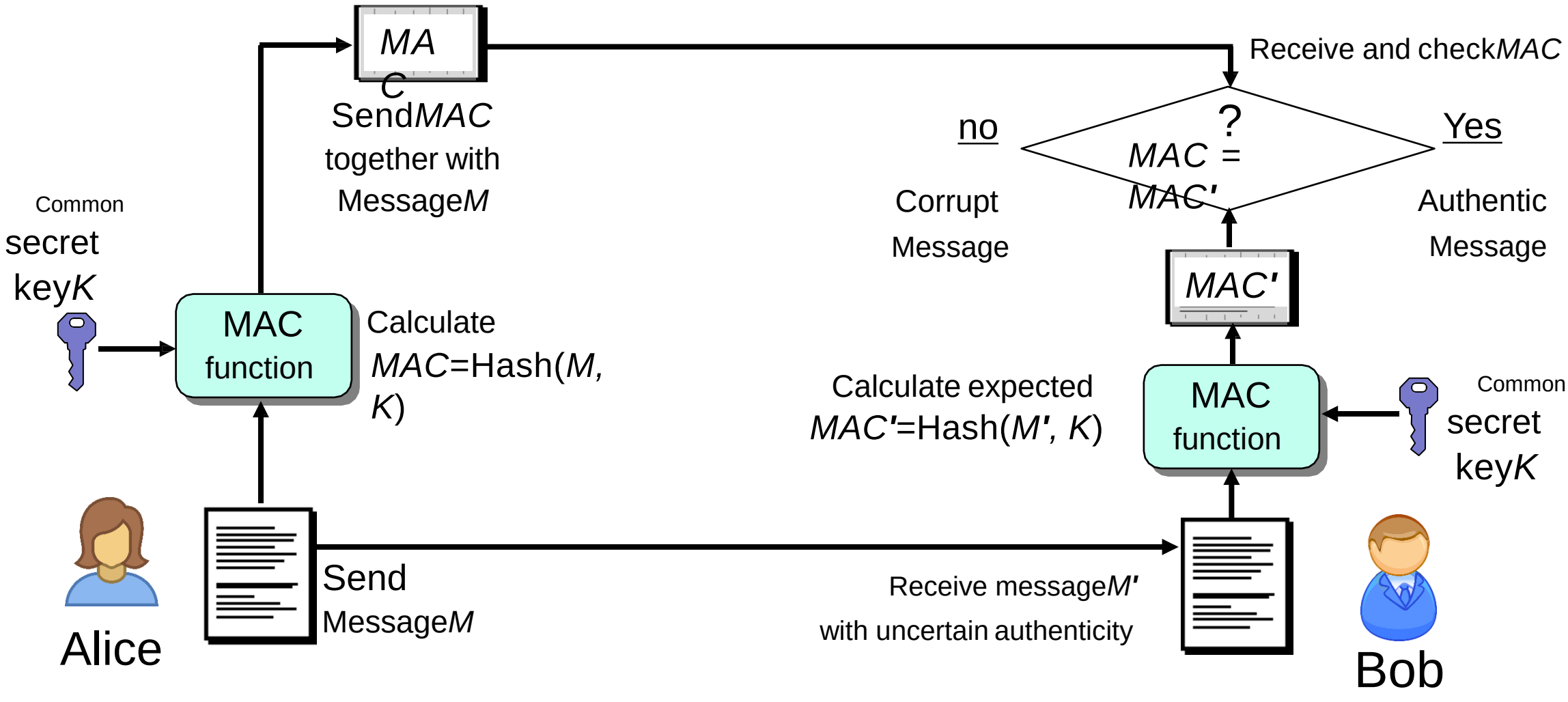
Applications of hash functions

- Comparison of files
- Password protection
- Integrity check
- Generation of Message Authentication Codes (MAC)
- Digital signatures
- Bitcoin and cryptocurrency
- Generation of pseudorandom numbers
- Generation of crypto keys

Message Authentication Code - MAC (Message Authentication Code)

- A message *M* with a simple hash value Hash(*M*) can be easily changed by 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 calculation, which provides an authenticated hash value *MAC*=Hash(*M, k*).
- To validate and authenticate a message, the recipient must have the same secret key *k* which was used by the sender to calculate *MAC*.
- A third party who does not know the key cannot validate *MAC*-value.

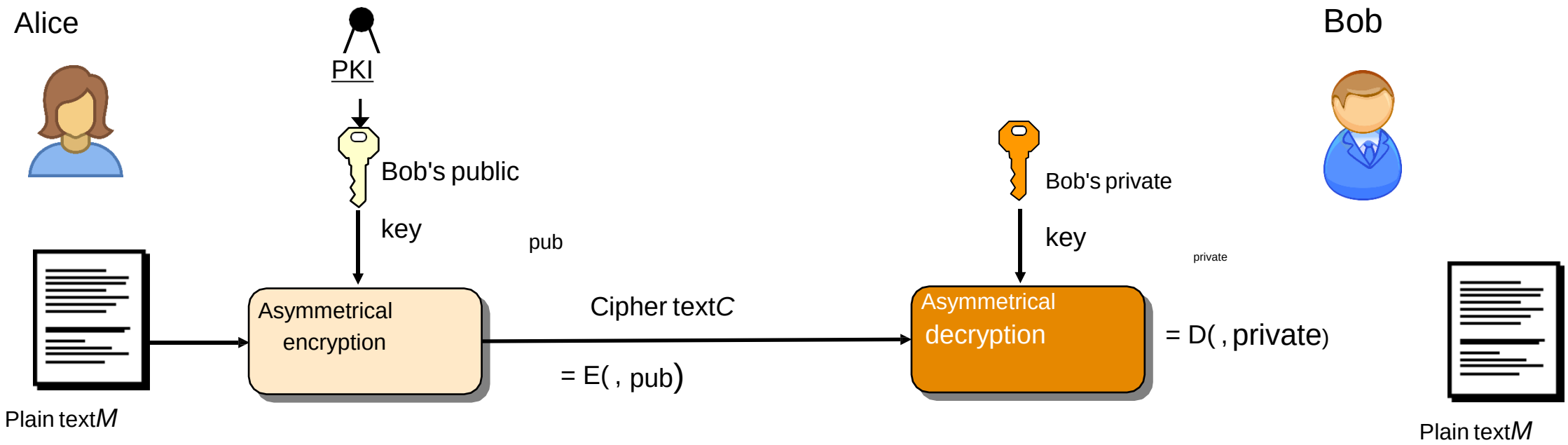
Message authentication with MAC



ASYMMETRICAL 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: best known asymmetric algorithm.
  - – RSA = Rivest, Shamir and Adleman (published 1977)
  - - History: British cryptographer Clifford Cocks invented the same algorithm in 1973, but did not publish it because it was classified.
  - – Eventually requires large keys (typically 2048 bits) to maintain security

- Elliptic curve cryptography
  - – Based on the difficulty of solving EC discrete logarithms.
  - - Keys are smaller (typically 256 bits) than RSA.

RSA algorithm

- R. Rivest, A. Shamir, L. Adleman (1977, pub. 1978)
- Public-key cryptosystem and digital signature scheme
- Based on factoring problem
  - given composite integer n, find primes p and q such that $n=pq$
  - hard for sufficiently large n
- Also based on “root extraction” problem
- Moderate speed, high security

RSA Key Generation

- Generate two random prime integers p and q and compute $n = p.q$

- Generate a random integer e such that

$$1<e<(p-1).(q-1)$$

and e is coprime with $(p-1).(q-1)$

- Compute integer d such that

$$ed \equiv 1 \pmod{(p-1).(q-1)}$$

RSA Key Generation

Public key: n, e

Private key: d

- where
  - n is a composite integer ("modulus")
  - e is an integer ("public exponent")
  - d is an integer ("private exponent")
  - $\phi(n)=(p-1).(q-1)$ is the Euler function

RSA: Encryption / Decryption

- Encryption:
  - Raises message m to the e^{th} power, i.e., encrypts m under receiver's public key:

$$c = m^e \bmod n$$

  - sends c ("ciphertext") to the receiver

- Decryption:
  - Raises the ciphertext c to the d^{th} power, i.e., decrypts c under the receiver's private key:

$$m = c^d \bmod n$$

RSA Complexity

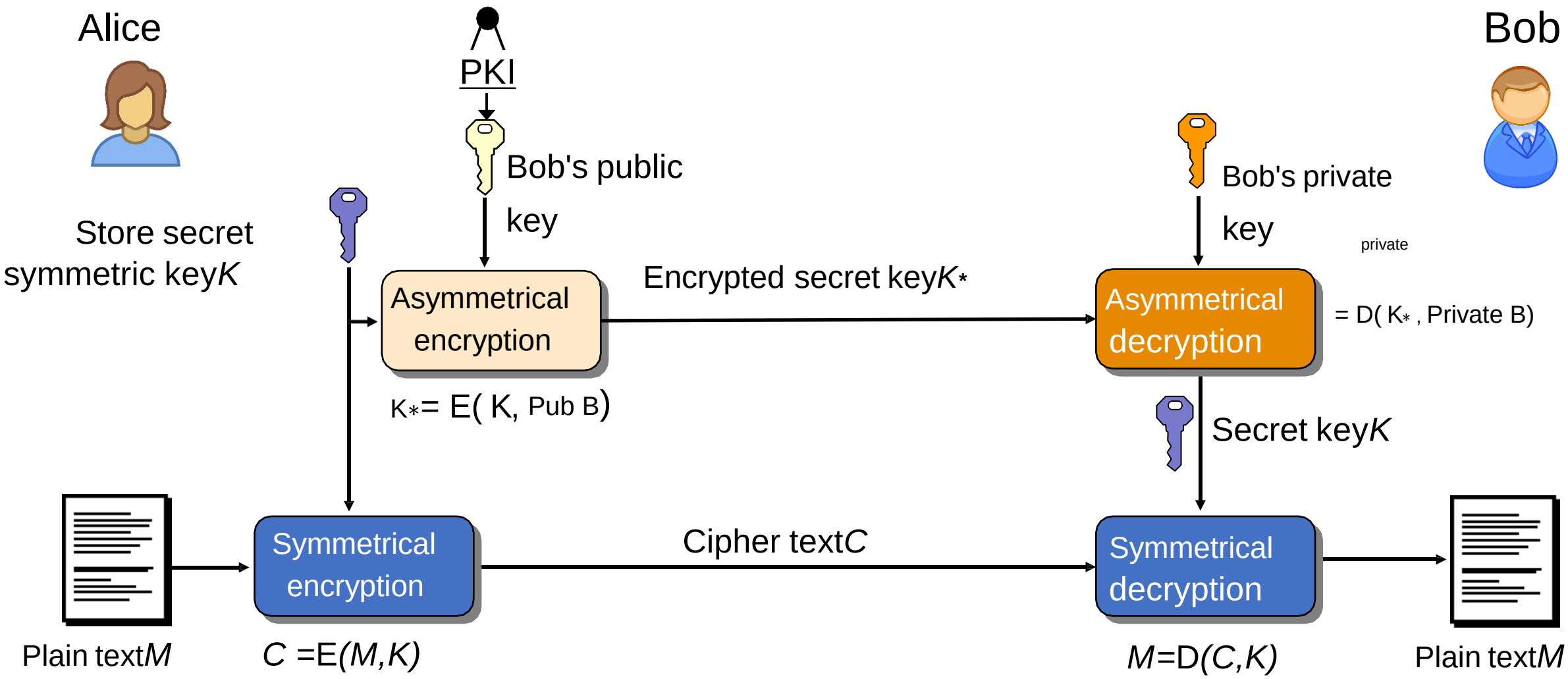
- Sender's work is easy: computes as few as two modular multiplications
- Receiver's work is easy: computes about 1.5 log*n* modular multiplications. Four times faster given primes *p, q*

- *Attacker's work is hard*
  - solves factoring problem, or root extraction
  - no efficient solutions known
- Security considered "super polynomial" in size of modulus

Hybrid encryption

- Symmetric ciphers are much faster than asymmetric ciphers (because symmetric ciphers have simple mathematical computation), but ...
- Asymmetric ciphers simplify key distribution, therefore…
- Practical 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 secret symmetric key is used together with the symmetric cipher to encrypt data sets and messages.

Hybrid encryption



Diffie-Hellman key exchange

Alice chooses private subkey a

Bob chooses private subkey b

Alice calculates publicly [mod]

Bob calculates publicly [mod]

Alice sends to Bob ⟶

[mod]

[mod] ⟵ Bob sends to Alice

Alice secretly calculates = ( ) towards [ ]

Bob secretly calculates = ( ) towards [ ]

Alice and Bob have exchanged anonymous secret key

Attackers cannot find the secret subkeys a and b because calculating the discrete logarithm of large integers is difficult. Thus, attackers cannot calculate the secret key $= g_{ab} \bmod p$.

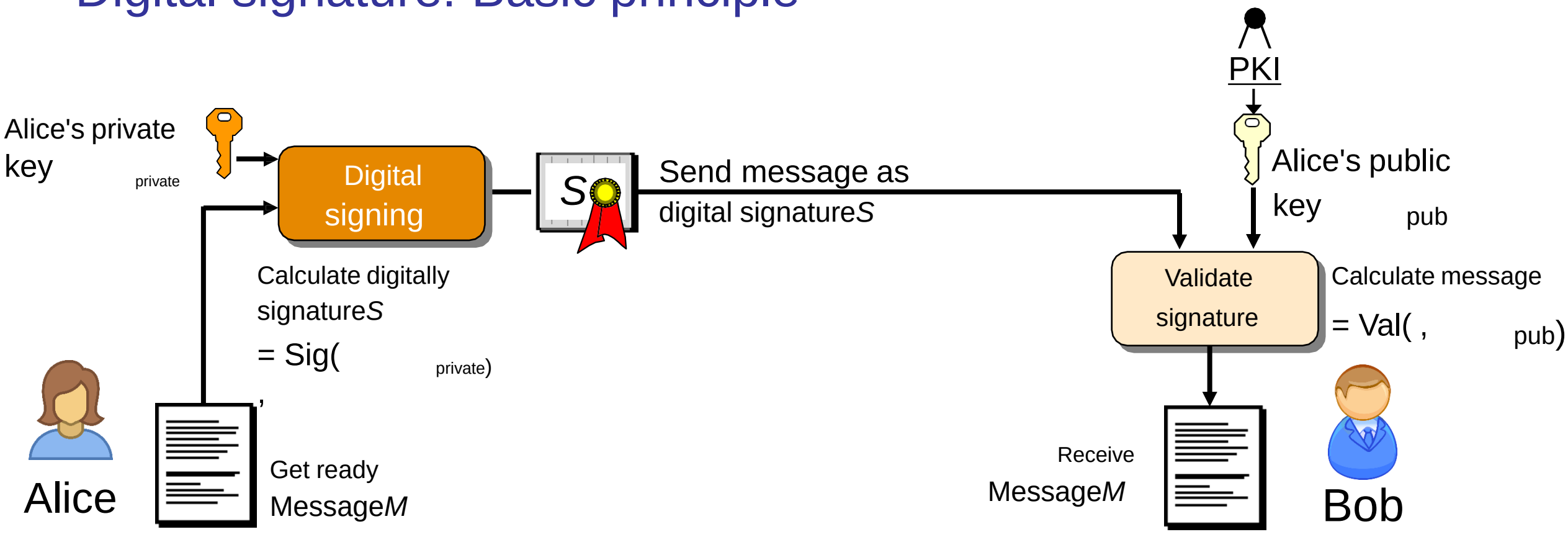
Diffie-Hellman key exchange

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

Need for digital signature

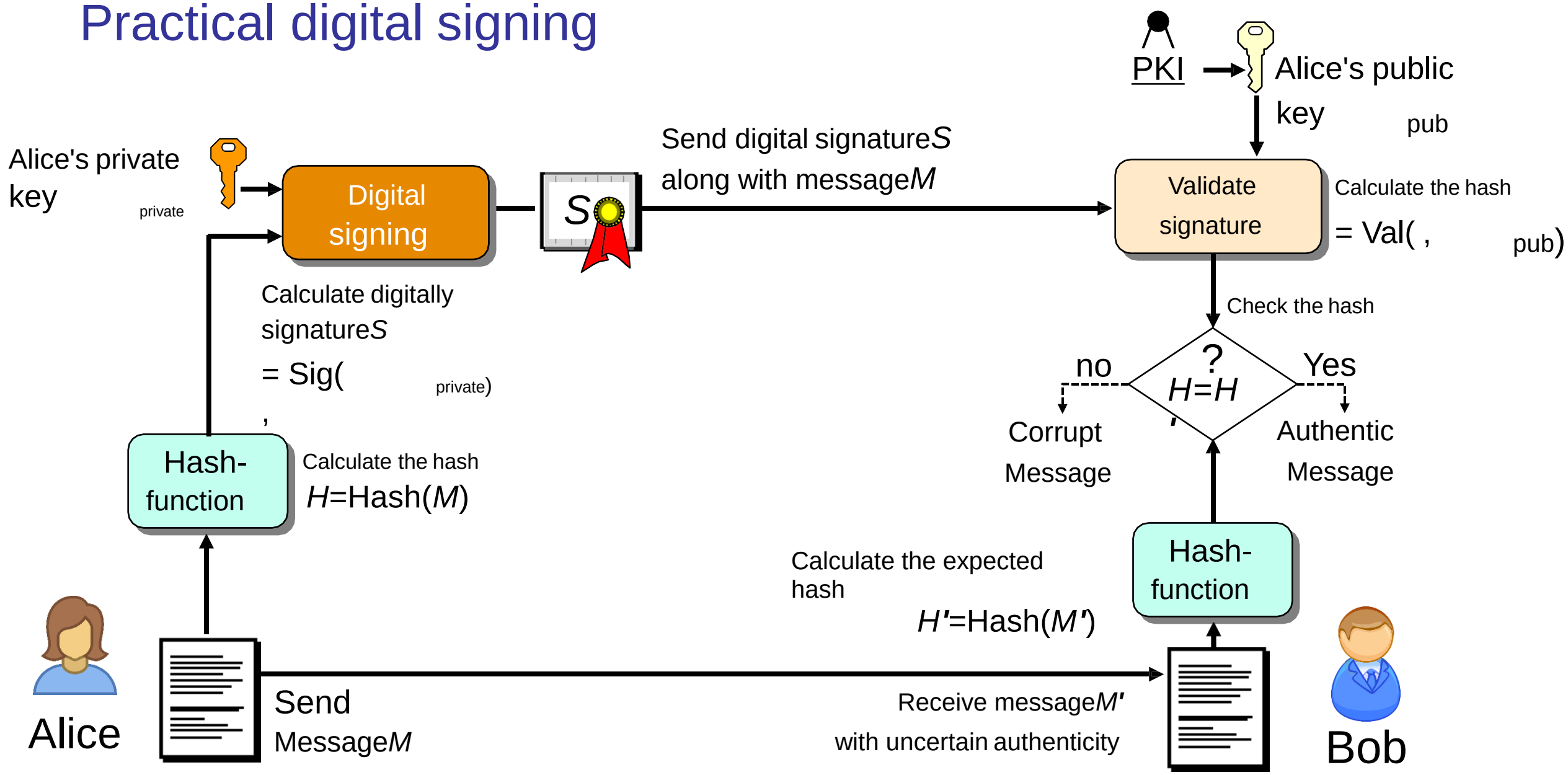
- A MAC cannot be used as proof of data authenticity to be verified by a third party
- Digital signatures can be validated by third parties
  - Provides strong (non-repudiable) data authentication,
- Features for digital signature:
  - Signing (using private key)
  - Validation (uses public key)

Digital signature: Basic principle

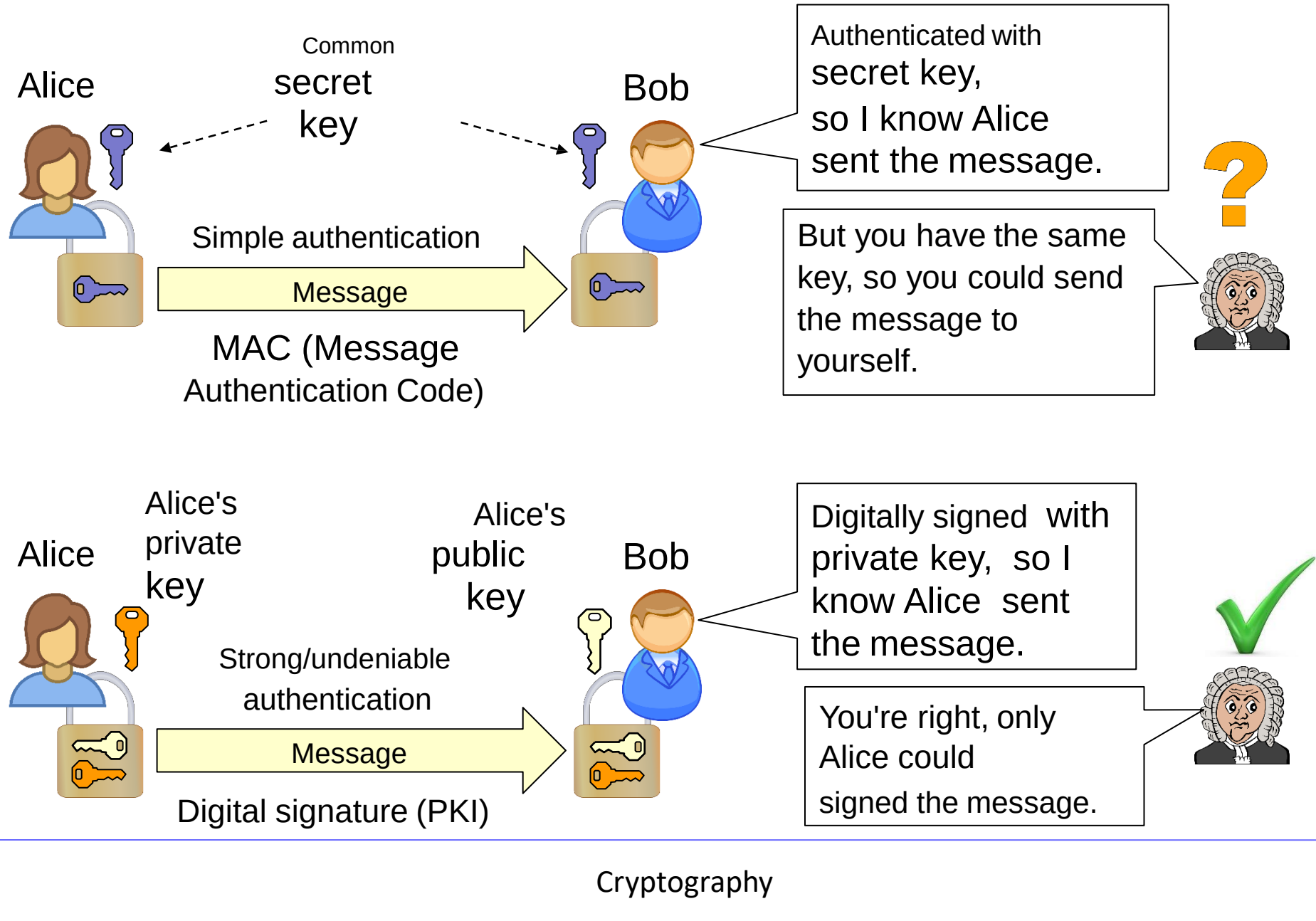


- Digital signing and validation require heavy calculation, and are not used for direct signing as shown above.
- In practice, hybrid signing is used which combines a hash function and digital signing.

Practical digital signing



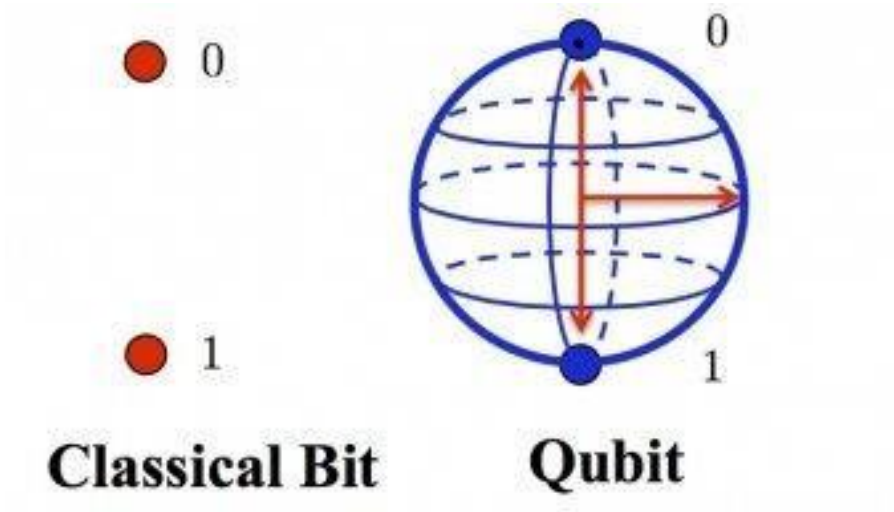
Simple and strong data authentication



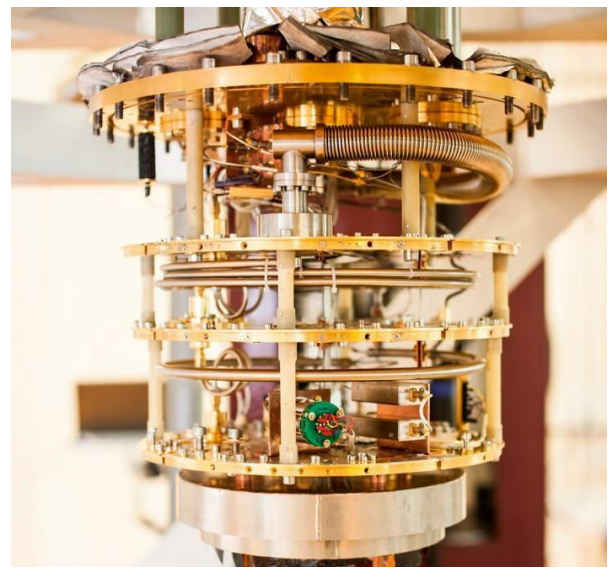
POSTQUANTUM CRYPTOGRAPHY

Quantum computers

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



Experimental quantum computer

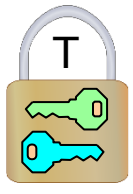


- Quantum algorithms, i.e. algorithms for quantum computers, can potentially break common asymmetric crypto-algorithms,
  e.g. RSA, DSA and Diffie-Hellmann.

Cryptographic functions

Symmetrical encryption

Hash-functions



Asymmetrical encryption & Digital Signature (Traditional), e.g. RSA, DSA, Diffie-Hellmann

Confidentiality

Authenticity / Integrity

Digital Signature
PKI / key distribution

Confidentiality

The quantum threat

Cryptographic functions

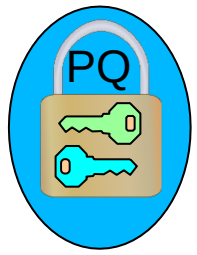
Symmetrical encryption

→ Confidentiality

→ Authenticity / Integrity

Hash-functions

→ Digital Signature
→ PKI / key distribution

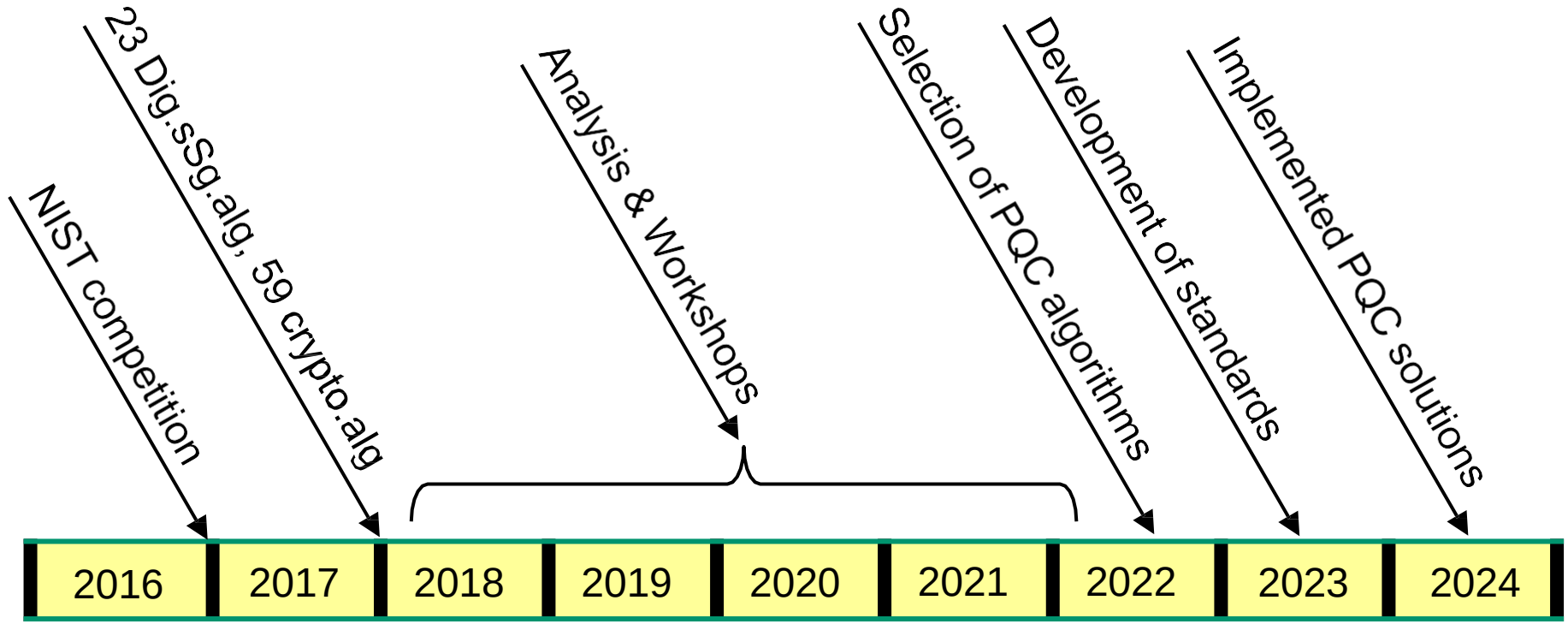


Asymmetrical encryption & Digital Signature (Post-Quantum), e.g. Lattice-based, Multivariate, Hash-based, Code-based, Elliptic curve isogeny

→ Confidentiality

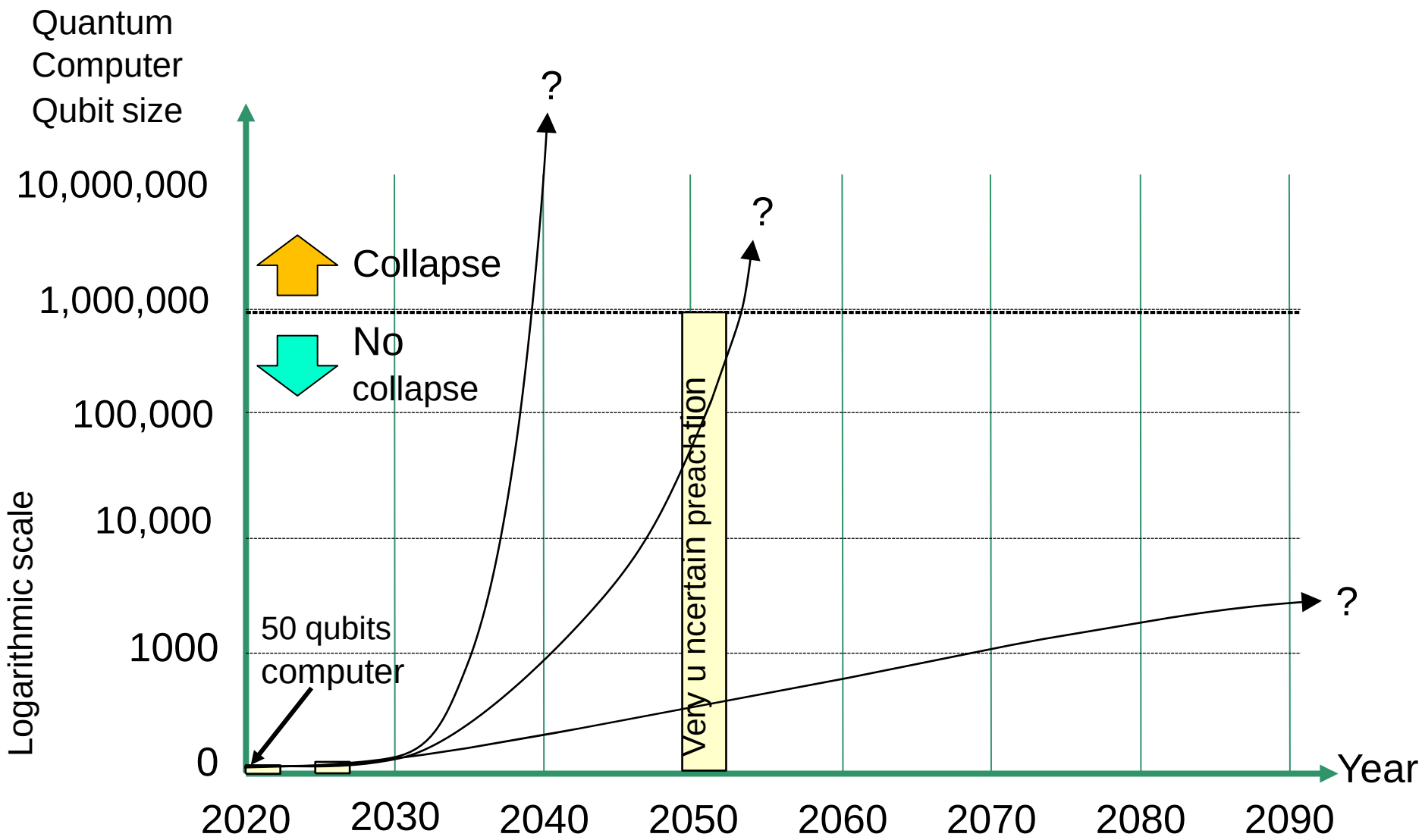
With post quantum crypto we can keep DigSig and PKI even with quantum computers that have 1 million qubits

Standardization of post quantum crypto



- The term "post-quantum crypto" (Post-Quantum Crypto) means cryptography that cannot be broken by quantum computers.

Breakdown of trad. asymmetric crypto?



Why move to post-quantum cryptography?

A. Use post-quantum crypto because quantum computers will probably crack RSA, Diffi-Hellmann and DSA sometime in the future.

B. Use post quantum crypto because you don't want your organization to end up on the front page of the newspaper, accused of being irresponsible.

END OF THE PRESENTATION

Refrence

Information Security , Åvald Åslaugson Sommervoll, University of Oslo