University of Minho
School of Engineering

# Distributed Data Processing Environments

## Bachelor in Data Science

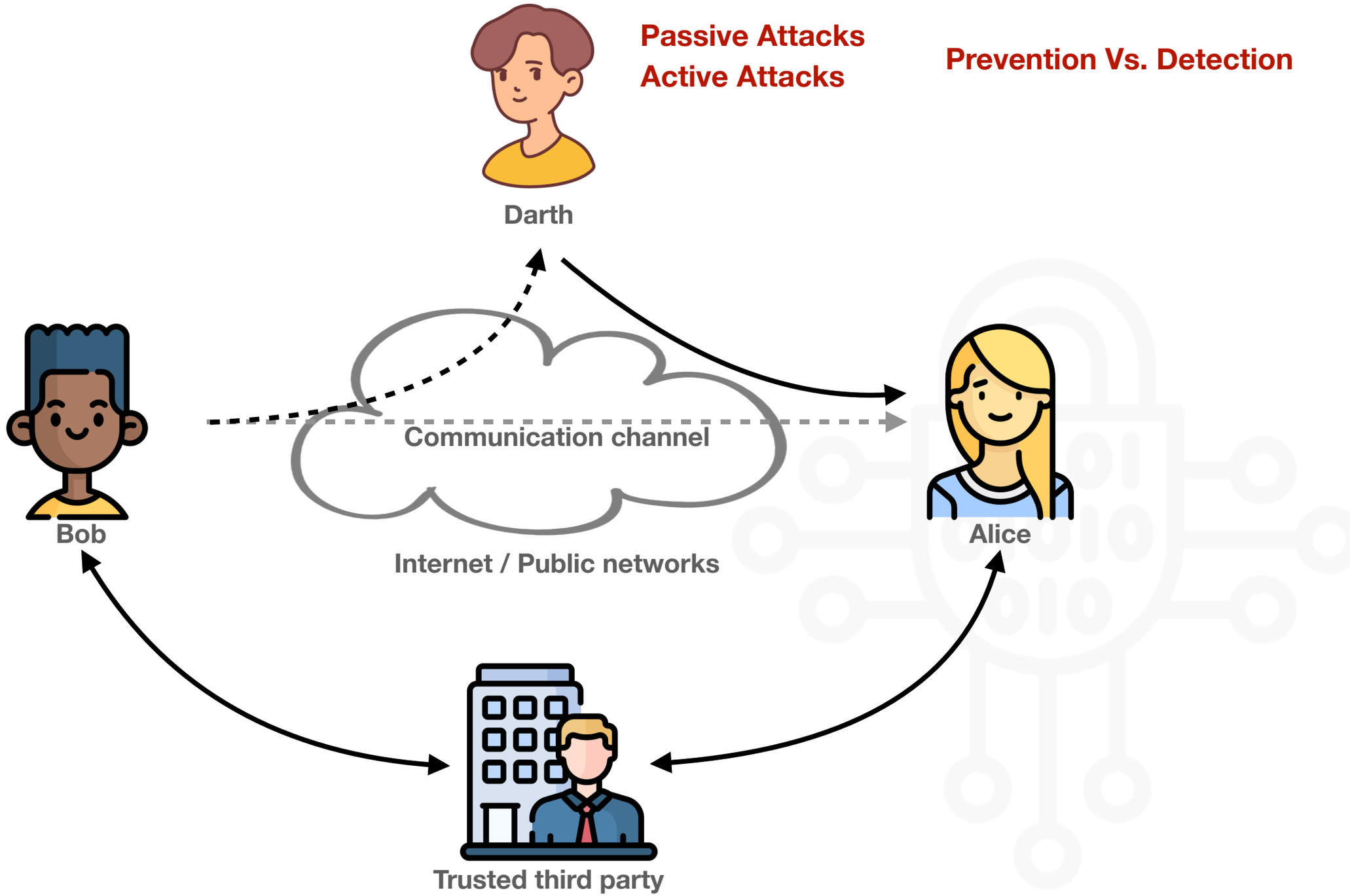
**João Marco Silva**

Department of Informatics
joaomarco@di.uminho.pt

2024/2025

# Principles

## Communication Model




**Passive Attacks**
**Active Attacks**

**Prevention Vs. Detection**

# Principles

## What is Cryptography?

**The discipline that embodies the principles, means, and methods for the transformation of data to hide their semantic content, prevent their unauthorised use, or prevent undetected modification.** NIST SP 1800-21B

## It aims at protecting security properties

- Confidentiality
- Integrity
- Availability
- Authenticity
- Non-repudiation

# Principles

- Summary of the History of Modern Cryptography
  - 1948-1949: Cryptography was initiated as a scientific area after Claude Shannon's development of the Information Theory that allows formalising notions of security.
  - 1970-1977: Data Encryption Standard (DES).
  - 1976: Asymmetric Cryptography (*Diffie* & *Hellmann*)
  - 1978: *Rivest*, *Shamir*, & *Adelman* (RSA)
  - 1985: *El Gamal*.
  - 1995: Digital Signature Algorithm (DSA)
  - 2001: Advanced Encryption Standard (AES)
  - 2016: Report on Post-Quantum Cryptography, NIST
  - 2022: Selection of Post-Quantum Algorithms, NIST

# Principles

**Plaintext:** the original message or data fed into a cryptographic algorithm as input.

**Encryption algorithm:** a set of mathematically expressed rules for rendering data unintelligible by executing a series of conversions controlled by a key.

CNSSI 4009-2015

**Secret key:** an input the encryption algorithm uses to perform the substitutions and transformation in the plaintext.

**Ciphertext:** the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts.

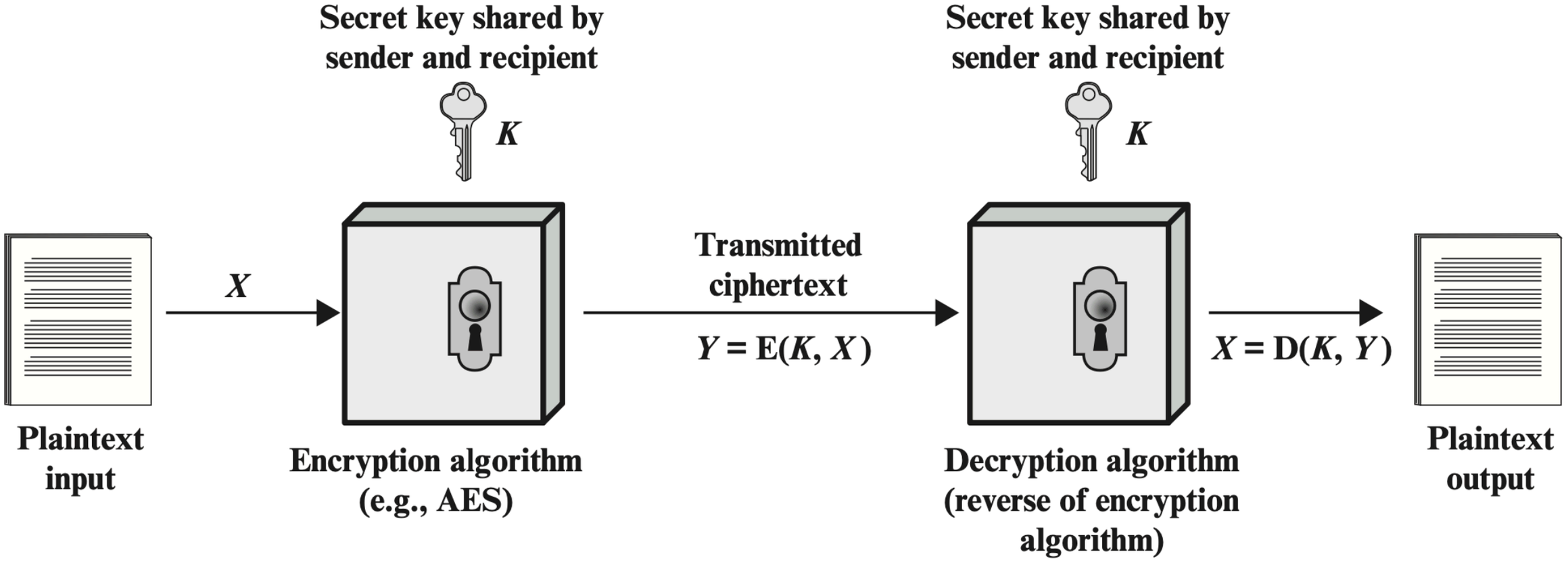
**Decryption algorithm:** a set of rules that take the ciphertext and the secret key to produce the original plaintext. This is essentially the encryption algorithm run in reverse.

**In symmetric cryptography, the encryption and decryption algorithms use the same key.**

# Symmetric Encryption

# Principles

## Simplified Model of Symmetric Encryption




Source: Stallings, W. & Brown, L. Cryptography and Network Security: Principles and Practice

# Principles

## Operation

**Block Ciphers:** the cryptographic algorithm operates on fixed-length blocks of the input (*e.g.,* plaintext) with the resulting output block (*e.g.,* ciphertext) of the same size.

- Common block sizes are 64 and 128 bits

**Stream Ciphers:** the cryptographic algorithm processes an individual bit, byte, or character of the input (*e.g.,* plaintext) at a time.

**Substitution:** each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements

**Permutation:** a sequence of plaintext elements is replaced by a permutation of the sequence without adding or removing any element from the original sequence.

# Principles

**Cryptanalysis:** the study of mathematical techniques for defeating cryptographic techniques without an initial knowledge of the used key. This includes looking for errors or weaknesses in the implementation of an algorithm or the algorithm itself.

CNSSI 4009-2015

**Cryptology:** the science that deal with cryptography and cryptanalysis.

CNSSI 4009-2015

# Principles

## Attacks on Cryptographic Systems

The objective is either to recover the plaintext or the secret key in use.

### Cryptanalysis

Rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-ciphertext pairs. The known characteristics are used to deduce a specific plaintext or the secret key.

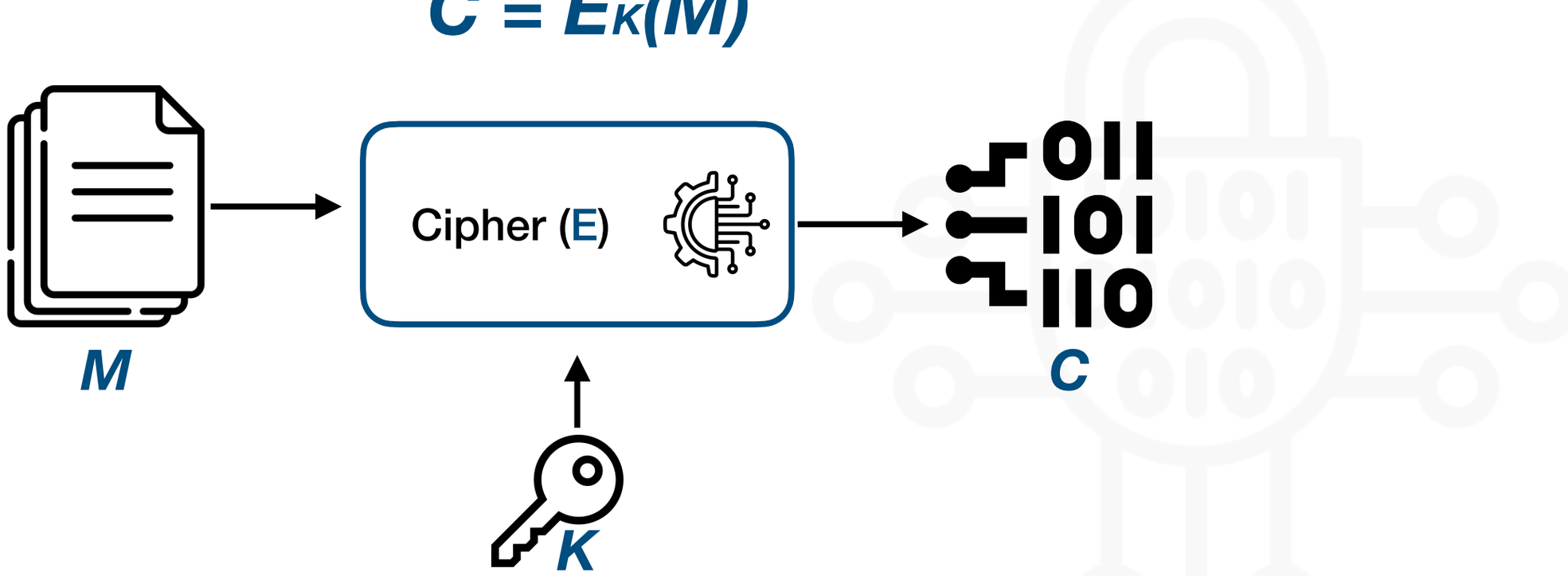
### Brute-force attack

The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

# Kerckhoff's Principle

**A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.**

$$C = E_K(M)$$




**Its security is ensured by the Secret Key**

# Principles

An encryption scheme is **unconditionally secure** if the ciphertext generated by the scheme does not contain enough information to recover the corresponding plaintext, no matter how much ciphertext is available and how much time an attacker has.

An encryption scheme is **computationally secure** if the ciphertext generated by the scheme meets one or both of the following criteria:

- The cost of breaking the cipher exceeds the value of the encrypted information;
- The time required to break the cipher exceeds the useful lifetime of the information.

# Classical Techniques

## Caesar Cipher

A substitution cipher allegedly created by Julius Caesar during the Gallic campaign.

**It works by replacing each letter of the alphabet with the letter standing three places further down the alphabet.**

**A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**

### Example

**Plaintext: attackatnineam**

**Ciphertext: DWWDFNDWQLQHDP**

**26 possible keys.**
**One of them is insecure.**

It is possible to use an arbitrary offset (key).

**Plaintext: attackatnineam**

**Ciphertext (k=6): GZZGIQGZTOTKGS**

# Classical Techniques

## Caesar Cipher

### Algorithm

By assigning a numerical equivalent to each letter, the algorithm can be expressed as follows:

| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

For each plaintext *p*, substitute the ciphertext letter *C*:

$$C = E(k,p) = (p + k) \bmod 26$$

The decryption algorithm is:

$$p = D(k,C) = (C - k) \bmod 26$$

# Classical Techniques

## Caesar Cipher

### Cryptanalysis

- The encryption and decryption algorithms are known
- There are only 26 keys to try
- The language of the plaintext is known and easily recognisable

**Ciphertext:** XQQXZHXQKFKBXJ

**Key = 1:** YRRYAIYRLGLCYK

**Key = 2:** ZSSZBJZSMHMDZL

**Key = 3:** ATTACKATNINEAM

**Key = 4:** BUUBDLBUOJOFBN

...

**Key = 24:** VOOVXFVOIDIZVH

**Key = 25:** WPPWYGWPJEJAWI

**Easy brute-force cryptanalysis**

# Classical Techniques

## Caesar Cipher

### Cryptanalysis

- The encryption and decryption algorithms are known
- There are only 26 keys to try
- The language of the plaintext is known and easily recognisable

**The plaintext output may not be recognizable if the plaintext language is unknown.**

### Example

**Original plaintext:** attackatnoonafterthesecondringofthechurchbells

**Compressed plaintext:** ILIOHNMKKOr3J3RsrPfz007Wfz00

# Classical Techniques

## Key Space

In most network situations, we can assume that the ciphers are known. What generally makes brute-force cryptanalysis impractical is the use of an algorithm that employs a large key space.

### Estimated time for brute-force cryptanalysis

| **Key size** | **Time** (1*μsec/test*) | **Time** (1*μsec/*$10^6$*test*) |
|---|---|---|
| 32 bit | 35.8 min | 2.15 msec |
| 40 bit | 6.4 days | 550 msec |
| 56 bit | 1140 years | 10 hours |
| 64 bit | 500000 years | 107 days |
| 128 bit | $5x10^{24}$ years | $5x10^{18}$ years |

- Note that increasing the key size by one bit doubles the available key space.
- Currently, $2^{80}$ is considered to provide an acceptable level of security.

# Classical Techniques

## Monoalphabetic Ciphers

- Enhances Caesar cipher by allowing arbitrary substitutions (i.e., *permutations*)
- A single cipher alphabet (mapping from plain alphabet to cipher alphabet) is used per message
- Increases the key space

### Example

**Consider the finite set of elements $S = \{a,b,c\}$**

**There are six permutations of $S$: $abc, acb, bac, bca, cab, cba$**

**What are the number of possible permutations for the alphabet?**

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

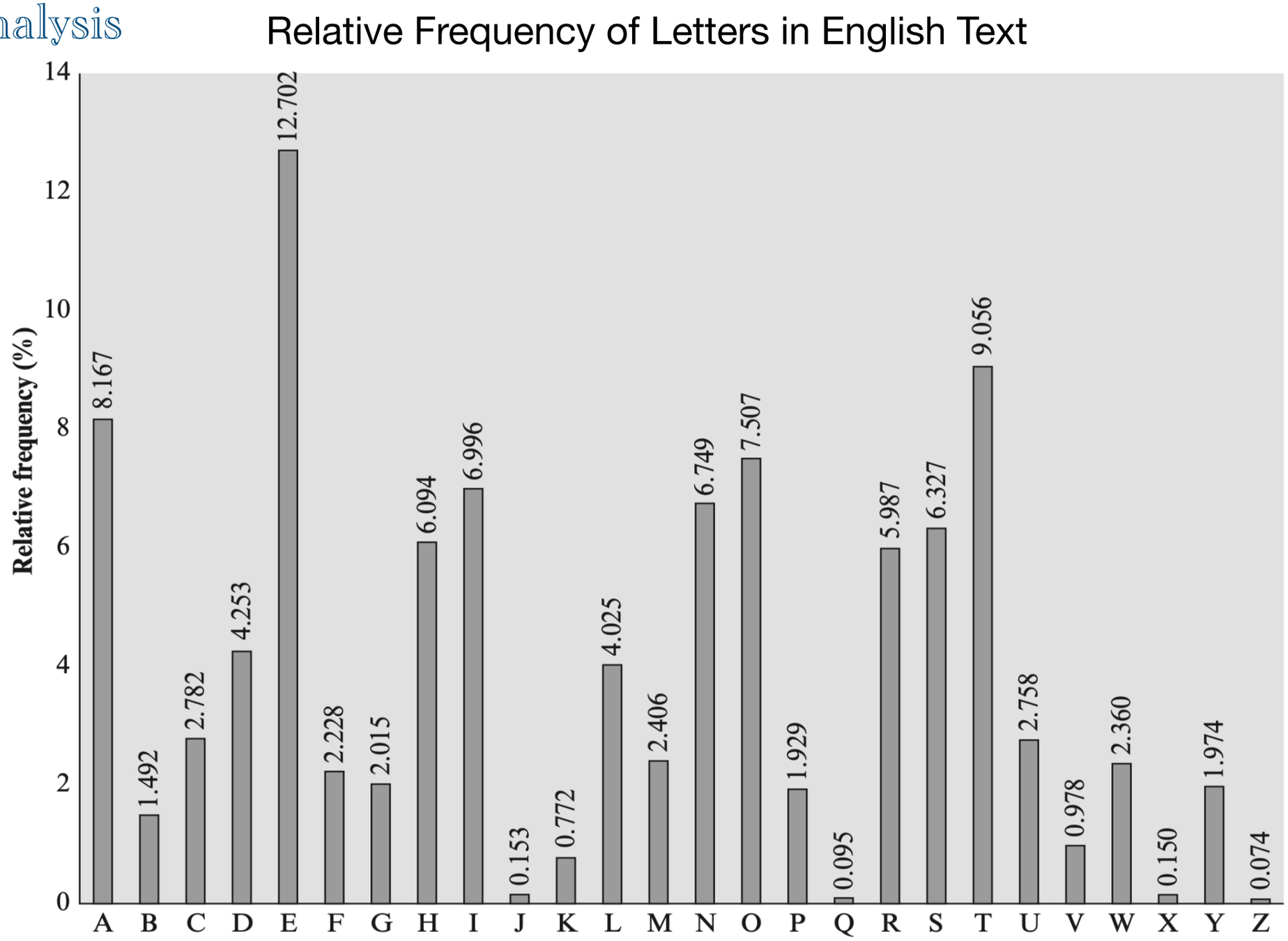
**In general, there are $n!$ Permutations of a set of $n$ elements**

$26! > 4 \times 10^{26}$ → Strong against brute-force cryptanalysis

# Classical Techniques

## Monoalphabetic Ciphers

Cryptanalysis




Source: Stallings, W. & Brown, L. Cryptography and Network Security: Principles and Practice

# Classical Techniques

## Monoalphabetic Ciphers

### Cryptanalysis - Example

**Ciphertext**

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
```

**Bigrams and trigrams are also useful**

The relative frequencies of the letters in the ciphertext

| | | | | |
|---|---|---|---|---|
| P 13.33 | H 5.83 | F 3.33 | B 1.67 | C 0.00 |
| Z 11.67 | D 5.00 | W 3.33 | G 1.67 | K 0.00 |
| S 8.33 | E 5.00 | Q 2.50 | Y 1.67 | L 0.00 |
| U 8.33 | V 4.17 | T 2.50 | I 0.83 | N 0.00 |
| O 7.50 | X 4.17 | A 1.67 | J 0.83 | R 0.00 |
| M 6.67 | | | | |

Use this to recover the plaintext!

Source: Stallings, W. & Brown, L. Cryptography and Network Security: Principles and Practice

# Classical Techniques

## Monoalphabetic Ciphers

### Cryptanalysis - Result

**Ciphertext**

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
```

**Plaintext**

```
it was disclosed yesterday that several informal but
direct contacts have been made with political
representatives of the viet cong in moscow
```

# Classical Techniques

## Transposition

Transposition ciphers generate ciphertext by performing some sort of permutation on the plaintext letters.

### Example - A simple transposition scheme

**Original plaintext:** **attackatnoon**

**Key:** **4213** (this is the permutation order)

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| a | t | t | a |
| c | k | a | t |
| n | o | o | n |

← **Read the matrix following the key order**

**Ciphertext:** **ATATTKCANONO**

### Cryptanalysis

The ciphertext preserves the frequency of the letters from the plaintext.

# Classical Techniques

## One-Time Pad

**An unconditionally secure cipher, that:**

- **Uses a key as long as the message (*i.e.*, plaintext);**
- **The key is randomly generated;**
- **Each key is used to encrypt and decrypt a single message.**

**It produces random output that bears no statistical relationship to the plaintext.**

**It usually operates over a binary alphabet using *XOR* operations.**

$$C_i = p_i \ XOR \ k_i \qquad p_i = C_i \ XOR \ k_i$$

In practice, its security is entirely due to the randomness of the key. If the stream of characters (or bits) in the key is truly random, then so are the characters (or bits) in the ciphertext.

# Classical Techniques

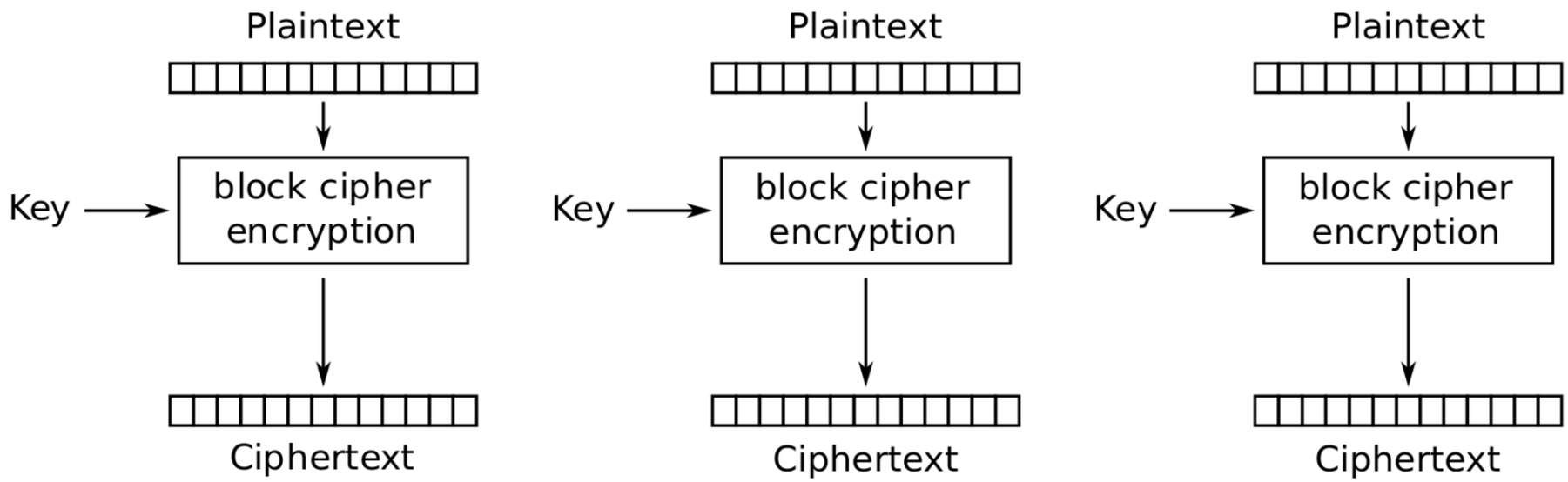
## One-Time Pad

**Despite offering unconditional security, the one-time pad has two fundamental drawbacks:**

- **Generating large numbers of random keys is a practical problem;**
- **Key distribution is an even more challenging problem since each message to be sent requires a new key of the same size that must be shared with the receiver.**

# Block cipher

The cryptographic algorithm operates on fixed-length blocks of the input (*e.g.,* plaintext) with the resulting output block (*e.g.,* ciphertext) of the same size.




- Typical block size: 64, 128 and 256 bits
- The plaintext is divided into blocks of equal size
- It is often required to make the plaintext a multiple of the block length - this is done by padding the last block
  - Bit padding
  - Byte padding
  - PKCS#5 and PKCS#7
  - ANSI X9.23

# Block cipher

## Diffusion & Confusion

Introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system.

*In a strongly ideal cipher, all statistics of the ciphertext are independent of the particular key used.*

**Confusion: each bit of the cryptogram must be a complex function of the bits of the cleartext.**

In this way, it will be difficult to discover unknown parts of the plaintext, even knowing some pieces, and it will also be difficult to deduce parts or all of the key used.

# Block cipher

## Diffusion & Confusion

Introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system.

*In a strongly ideal cipher, all statistics of the ciphertext are independent of the particular key used.*

**Diffusion: each bit of the cleartext must affect the highest number of bits in the corresponding ciphertext.**

In this way, any small change in the cleartext leads to major changes in the ciphertext, making it difficult to establish relationships between similar original texts and similar cryptograms.

The statistical structure of the plaintext is dissipated into long-range statistic of the ciphertext.

# Block cipher

## Feistel Chiper Structure

A practical application of Shannon's principle to develop a product cipher that alternates *confusion* and *diffusion* functions. In particular, it alternates *substitutions* and *permutations*.

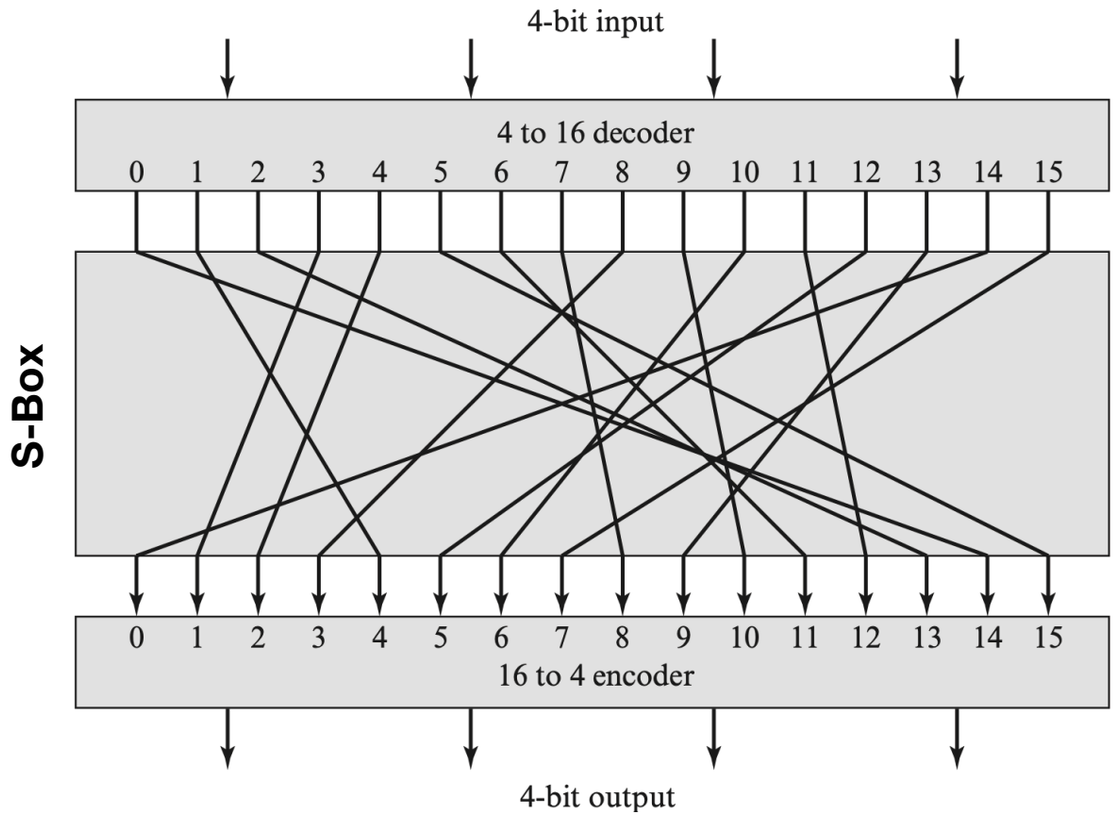
**Substitution:** Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements.

**Permutation:** A sequence of plaintext elements is replaced by a permutation of that sequence. No elements are added, deleted or replaced in the sequence.

# Block cipher

## Substitution & Permutation

### Substitution




- When operating on binary words, it can be seen as a permutation (exchange of wires) between a decoder/encoder pair
- The mapping itself constitutes the key

Example

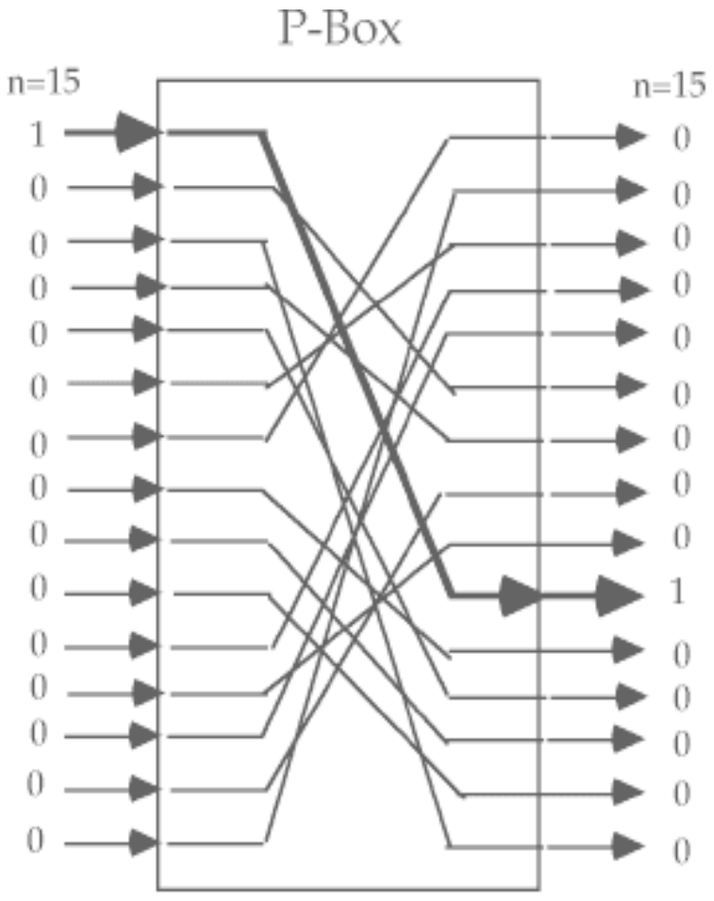
| Plaintext | Ciphertext |
|---|---|
| 0000 | 1110 |
| 0001 | 0100 |
| 0010 | 1101 |
| 0011 | 0001 |

Source: Stallings, W. & Brown, L. Cryptography and Network Security: Principles and Practice

# Block cipher

## Substitution & Permutation

### Permutation



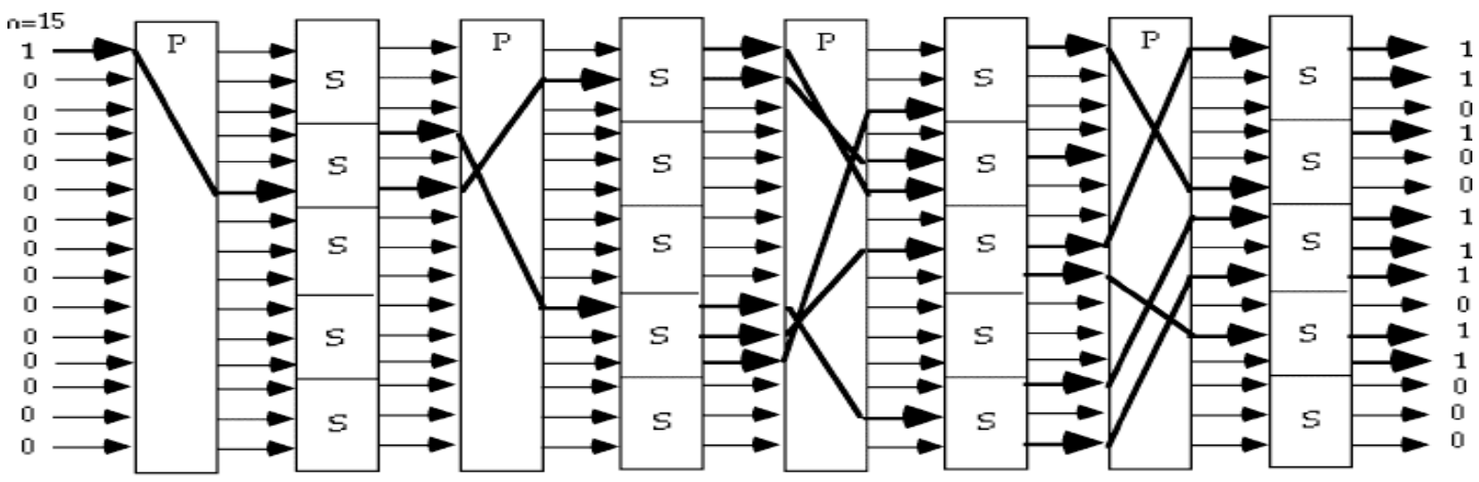

- The elements in the output are the same as in the input
- Its implementation is quite simple in hardware
  - Not so much in software
- Transpositions are simply an “exchange of wires” between the input and output bits

# Block cipher

## Substitution & Permutation

### S-P Networks

- Substitutions and permutations alone are not enough to build secure ciphers.
  - They are both idempotent operations
- However, by combining both techniques, it is possible to construct a non-idempotent *product cipher* called *round*
- One round alone is not secure enough. But when iterated, the desired security level can be achieved




It is impractical since it requires duplicated devices or programs for decryption.

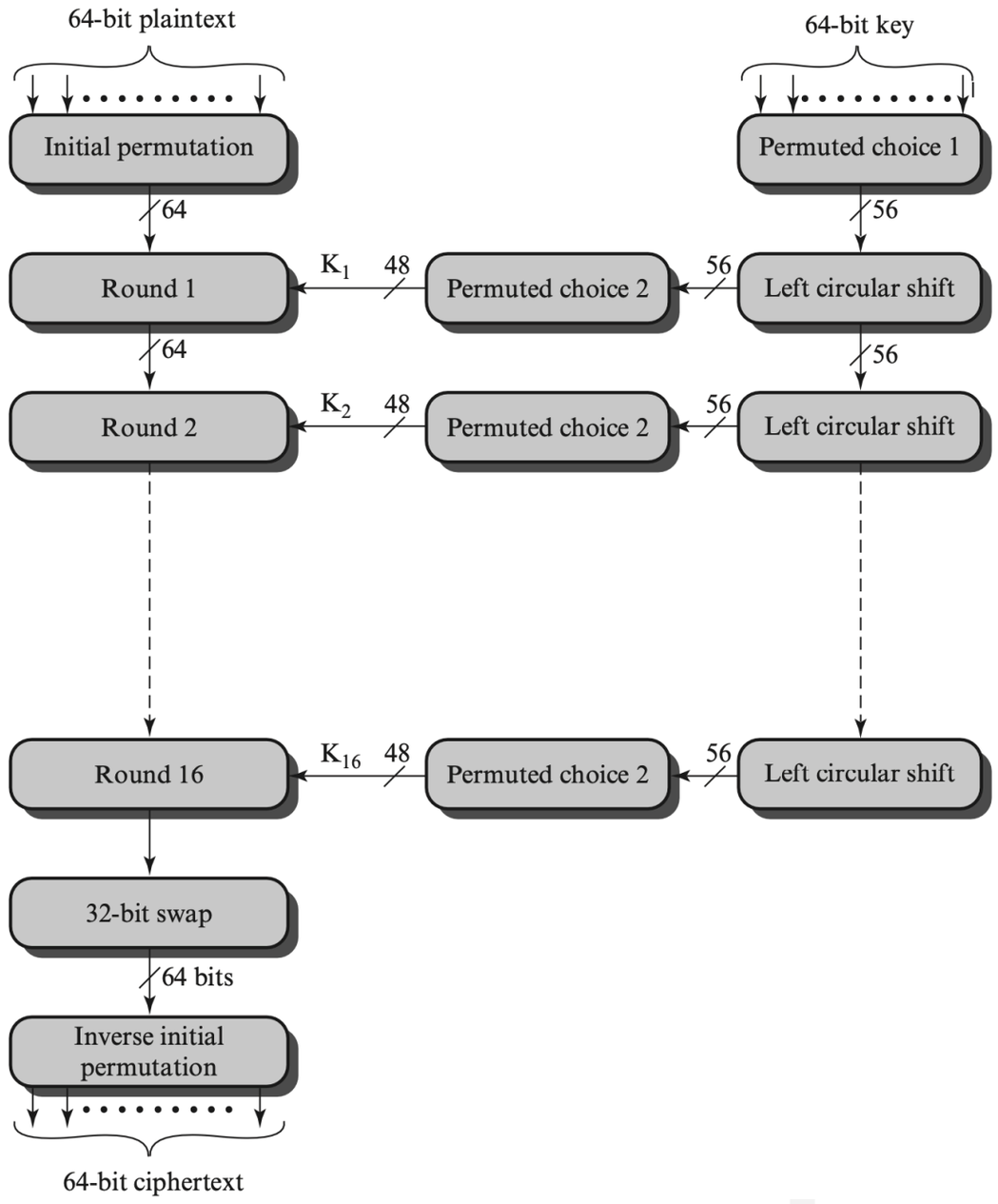
# Block cipher

## Data Encryption Standard (DES)

- Standardized in 1975
- An `initial permutation` is applied to the 64-bit block and then split into two 32-bit sub-blocks.
- This is followed by 16 `rounds`, according to the Feistel circuit.
- The 56-bit `key` is processed to produce 16 round keys of size 48 bits.
- The algorithm concludes by reversing the initial permutation.

DES is an outdated cipher with a choice of parameters that is not compatible with current security requirements (*e.g.*, key and block size).

- 3DES enhances the cryptographic strength of DES. However, it is still less secure than AES.



# Block cipher

## Advanced Encryption Standard (AES)

- Announced by the NIST as US FIPS PUB 197 (FIPS 197) in 2001
  - Intended to replace DES and 3DES with an algorithm that is more secure and efficient
    - Proposed by ***Rij****men*-***Dae****men* (*Rijndael*)
- Its design has a strong mathematical foundation on *Finite Fields / Galois Field*
- All operations can be executed by `XOR` and `lookup` tables
  - Efficient for software and hardware implementation

# Block cipher

## Modes of operation

To apply a block cipher in various application types, five operation modes have been defined by NIST.

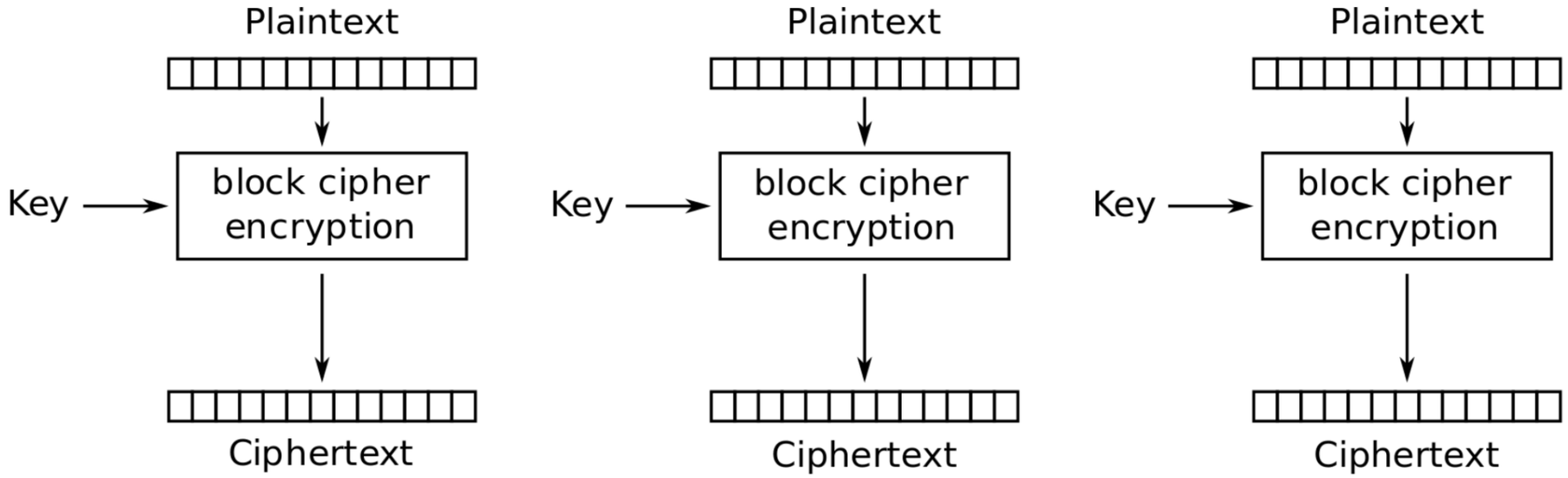
- **Electronic Codebook (ECB)**
- **Cipher Block Chaining (CBC)**
- **Cipher Feedback (CFB)**
- **Output Feedback (OFB)**
- **Counter (CTR)**

# Block cipher

## Modes of operation

### Electronic Codebook (ECB)

**Each block of the plaintext bits is encoded independently using the same key.**




#### Typical application

- Secure transmission of single values.

# Block cipher

## Modes of operation

### Electronic Codebook (ECB)

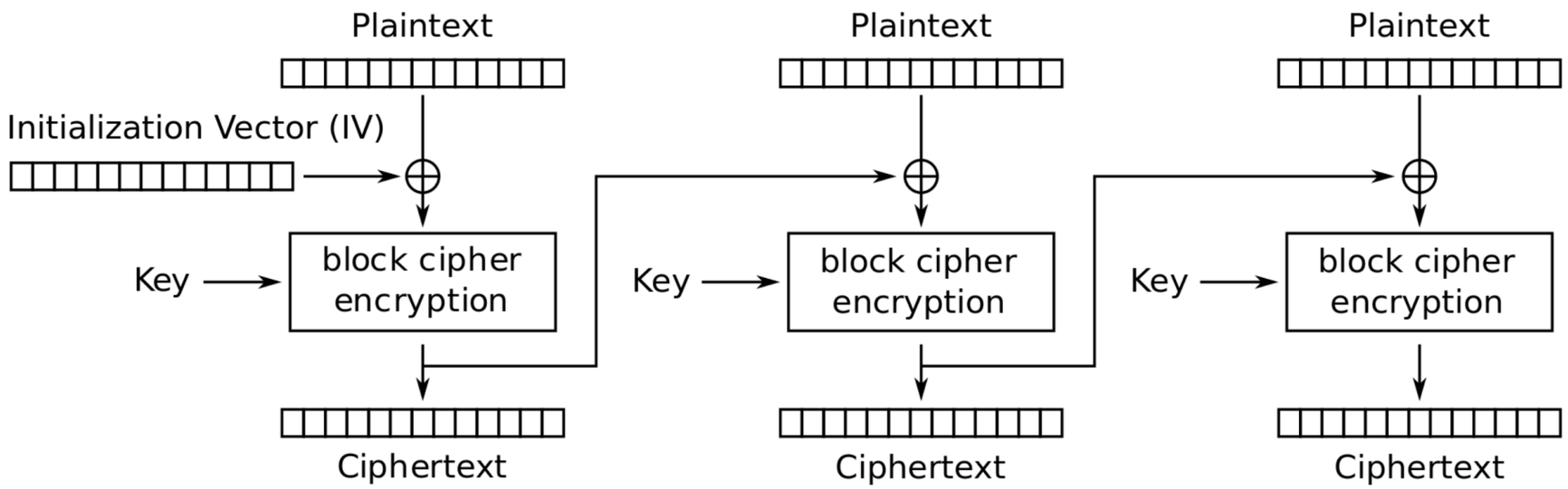
- For lengthy cleartext, the ECB mode may not be secure
- Repeated blocks are detected (*code book attack*)
  - Two identical blocks in the cleartext produce two identical blocks in the ciphertext
- Vulnerable to replay and substitution attacks
- A one-bit error in one block of the cryptogram affects a single block of the plaintext after decryption
- It should only be used to encrypt messages in a single block (or a few)

# Block cipher

## Modes of operation

### Cipher Block Chaining (CBC)

**The input to the encryption algorithm is the XOR of the next block of plaintext and the preceding block of ciphertext.**




### Typical application

- General-purpose block-oriented transmission
- Authentication

# Block cipher

## Modes of operation

### Cipher Block Chaining (CBC)

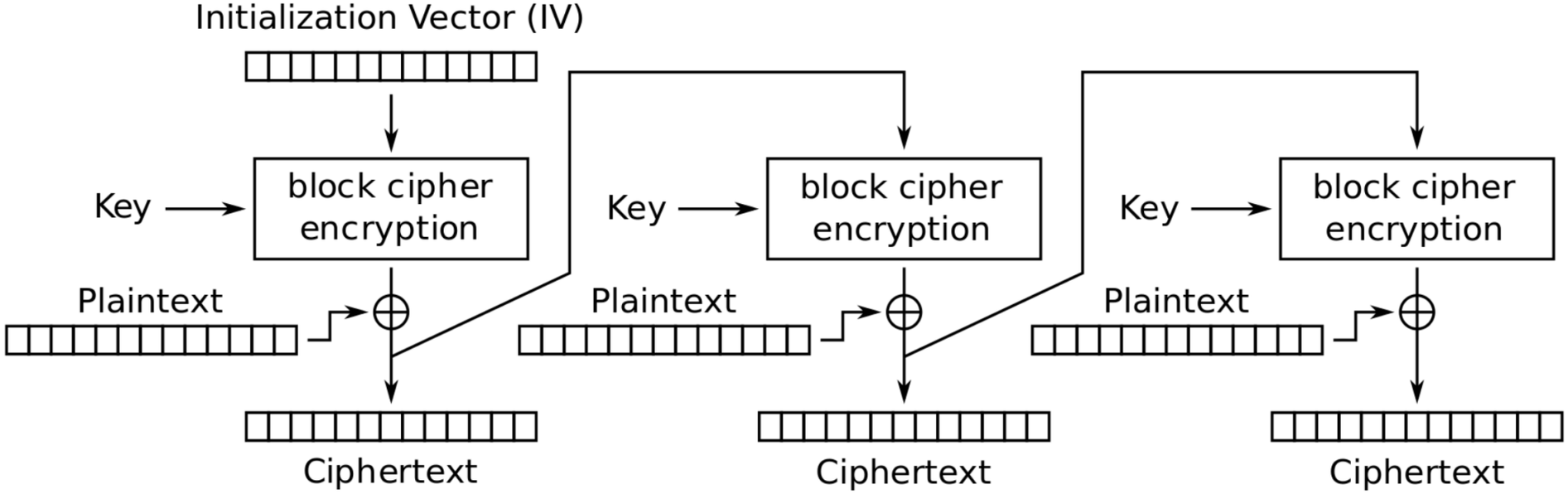
- Its main objective is to prevent the repetition patterns of ECB
- The value of the Initialization Vector (IV) used to process the first block can be secret or not
  - If the IV is sent in cleartext, an attacker can change bits of the first block by changing the respective bits of the IV
  - If the IV is fixed, the ciphertext of two messages with a common prefix will preserve this property
- An error in one block of the ciphertext corrupts two blocks after decryption
- The chaining process means that encrypting a block depends on all the blocks that precede it
  - It can be used in message authentication

# Block cipher

## Modes of operation

### Cipher Feedback (CFB)

**The input is processed $n$ bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with the plaintext to produce the next unit of ciphertext.**




Typical application

- General-purpose stream-oriented transmission
- Authentication

# Block cipher

## Modes of operation

### Cipher Feedback (CFB)

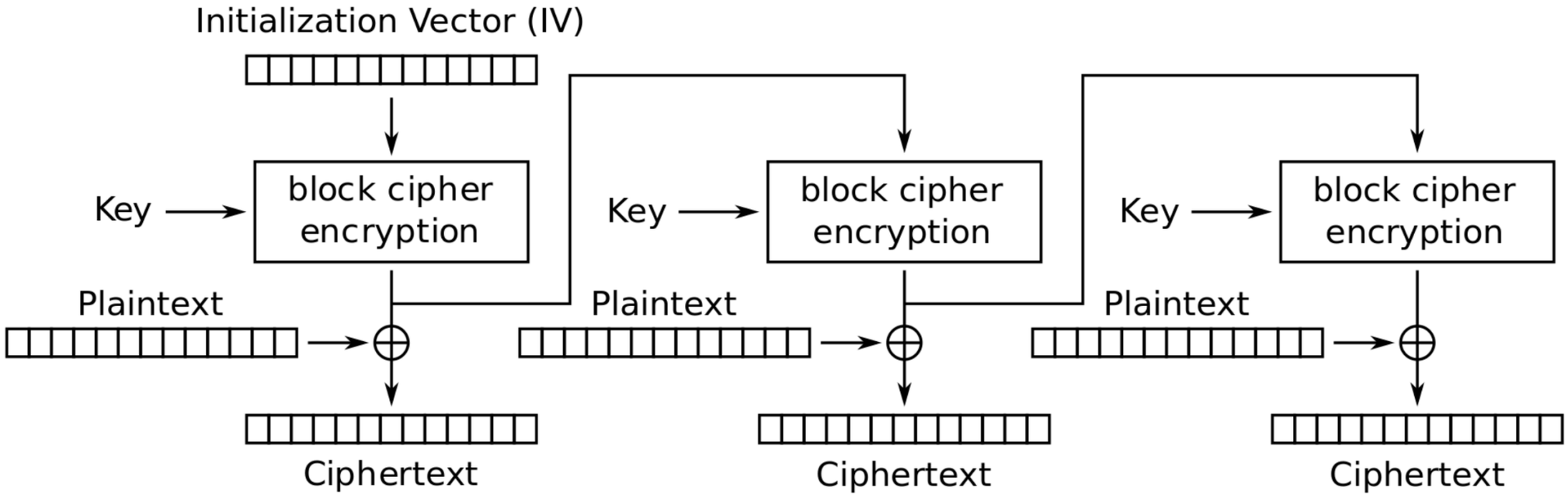
- A mode that transforms a block cipher into a stream cipher
- The number of bits in the feedback is variable (*i.e.*, $n$=`1`, `8` or `64`)
  - The feedback transfers the $n$ most significant bits to the least significant bits (with a shift of the rest)
- The IV must be unique for each use and can be sent in cleartext
- The key sequence depends on the IV, the cipher key, and all the cleartext already encrypted
- An error in one bit of the ciphertext affects the corresponding bit in the block and all bits in the next block

# Block cipher

## Modes of operation

### Output Feedback (OFB)

**Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.**




### Typical application

- Stream-oriented transmission over noisy channels, e.g., satellite communications

# Block cipher

## Modes of operation

### Output Feedback (OFB)

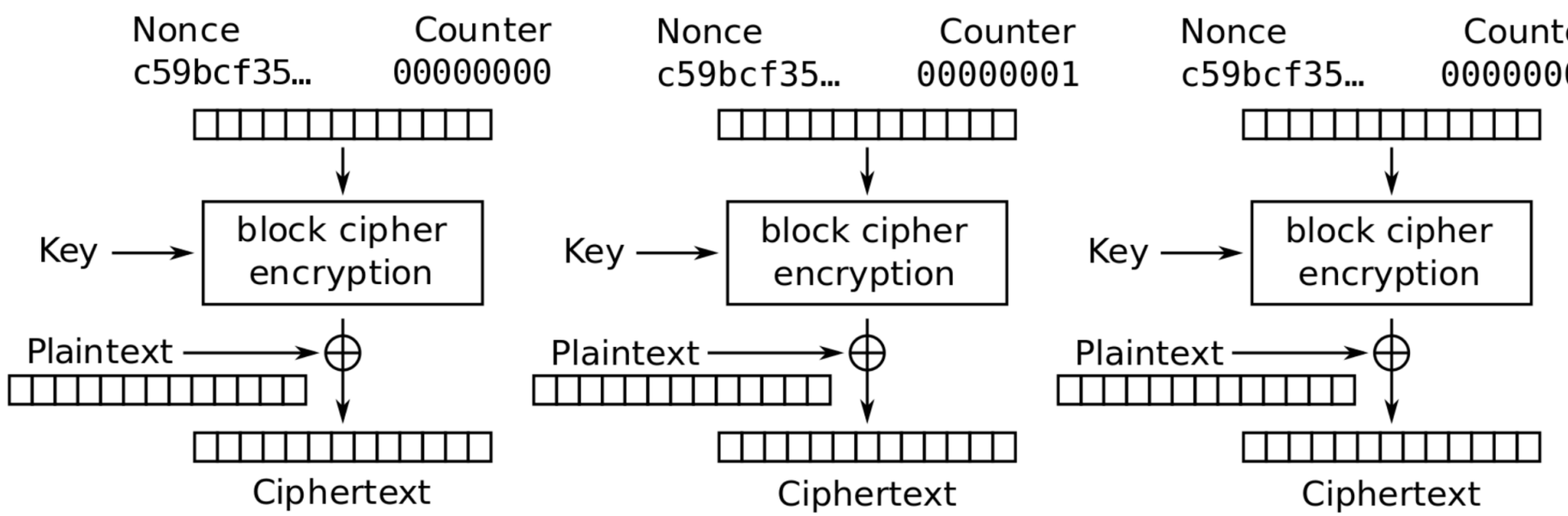
- A mode that transforms a block cipher into a stream cipher
- The key sequence is obtained by iterating the cipher over an initial block, *i.e.*, the IV
- The key sequence is independent of the message, so it can be processed regardless of the message availability
- Bit errors in the ciphertext only affect the respective bits in the decrypted plaintext
  - An advantage over CFB, since bit errors do not propagate
  - However, it is more vulnerable to message stream modification attack

# Block cipher

## Modes of operation

### Counter (CTR)

**Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.**




### Typical application

- General-purpose block-oriented transmission
- Useful for high-speed requirements

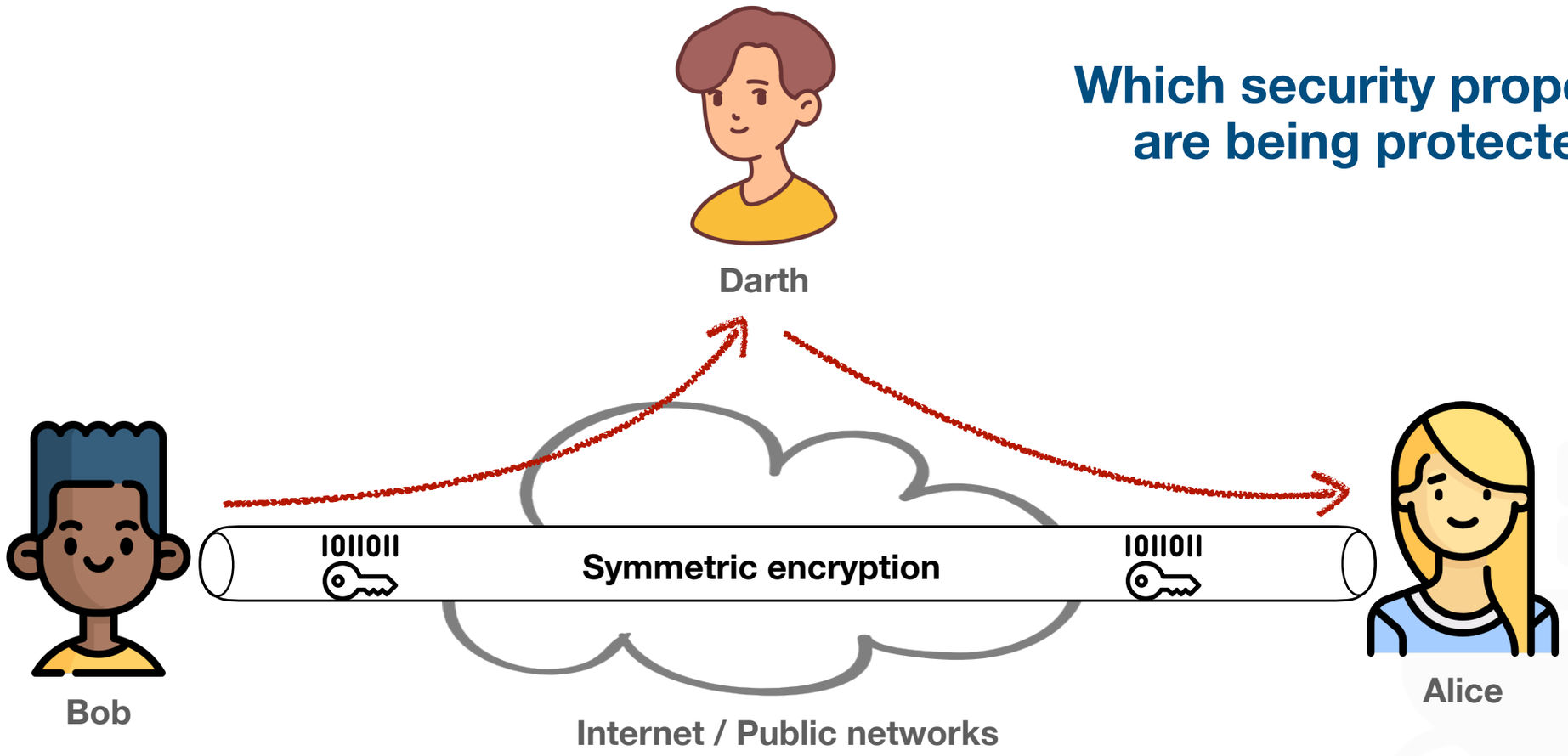
# Block cipher

## Modes of operation

### Counter (CTR)

- Efficient hardware and software implementations
- Simple and secure
- The *Nonce* (IV) and *Counter* can be combined in different ways (concatenated, `XOR`ed, etc.)
- The requirement of the *Counter* is to produce different values for all the blocks
  - A counter is the simplest way to do it
- There is no dependency between processing blocks (they can be processed in parallel, randomly, etc.)

# Cryptographic Hash Functions




**Which security properties are being protected?**

- **Confidentiality**
- **Integrity**
- **Availability**
- **Authenticity**
- **Non-repudiation**

# Cryptographic Hash Functions

## Hash functions

A ***hash function*** **`H`** accepts a variable-length block of data **`M`** as input and produces a fixed-size hash value **`h`.**

$$h = H(M)$$

Desired properties:

- The results of applying it to a large set of inputs will produce outputs that are evenly distributed and apparently random;
- A change to any bit in **`M`** results, with high probability, in a change to **`h`.**

### Cryptographic Hash Functions

Hash functions for which is computationally infeasible to find:

- A data object that maps to the same hash result;
- Two data objects that map to the same hash result.

Hash functions are often used to determine whether data has changed. **`<Integrity>`**

# Cryptographic Hash Functions

## Applications of Cryptographic Hash Functions

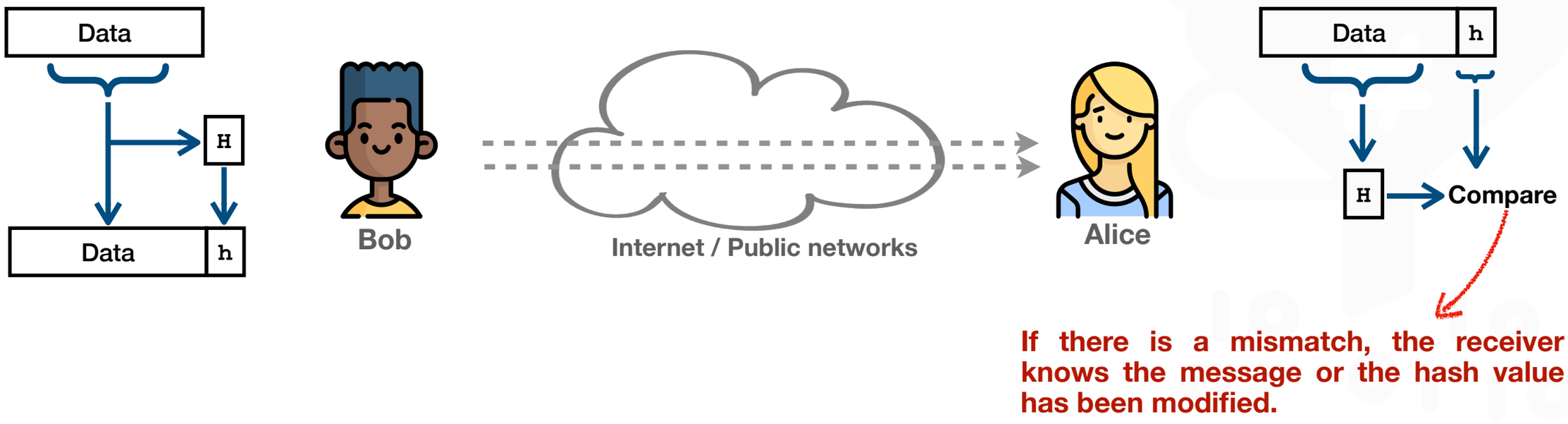
- Message Authentication
- Digital Signatures
- One-Way Password File
- Intrusion Detection
- Virus Detection
- Pseudorandom Functions (PRF)
- Pseudorandom Number Generator (PRNG)

# Cryptographic Hash Functions

## Message Authentication

**A mechanism or service used to verify a message's integrity.**

When used to ensure the sender's identity, the hash function value is also referred to as a *message digest*.

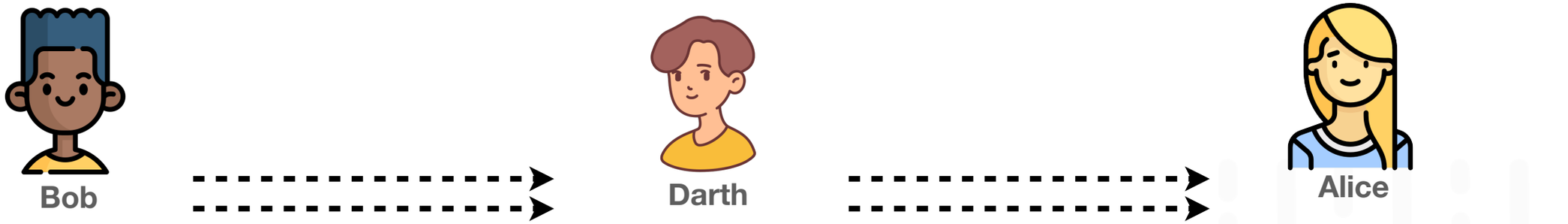


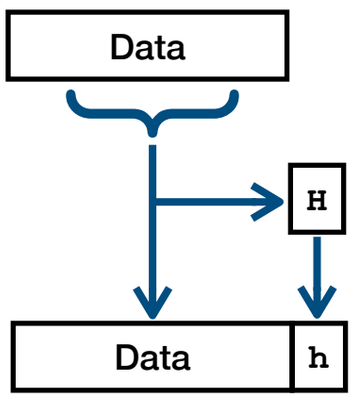
# Cryptographic Hash Functions

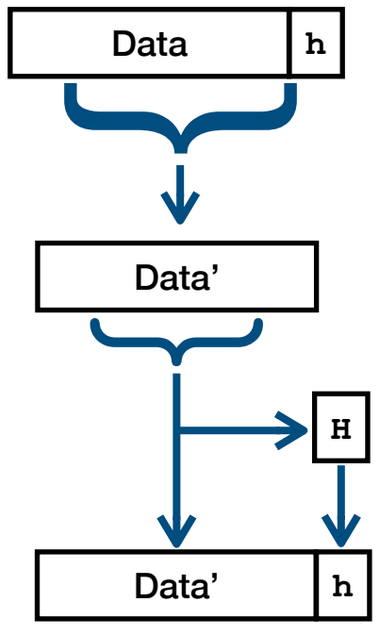
## Message Authentication

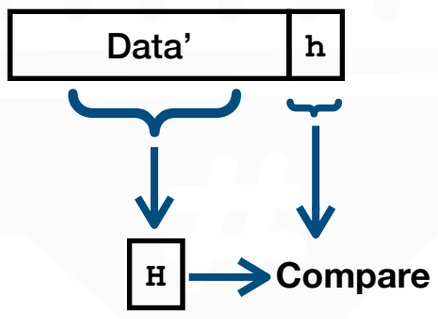
### Attack against Hash Function

**The hash value must be transmitted securely.**



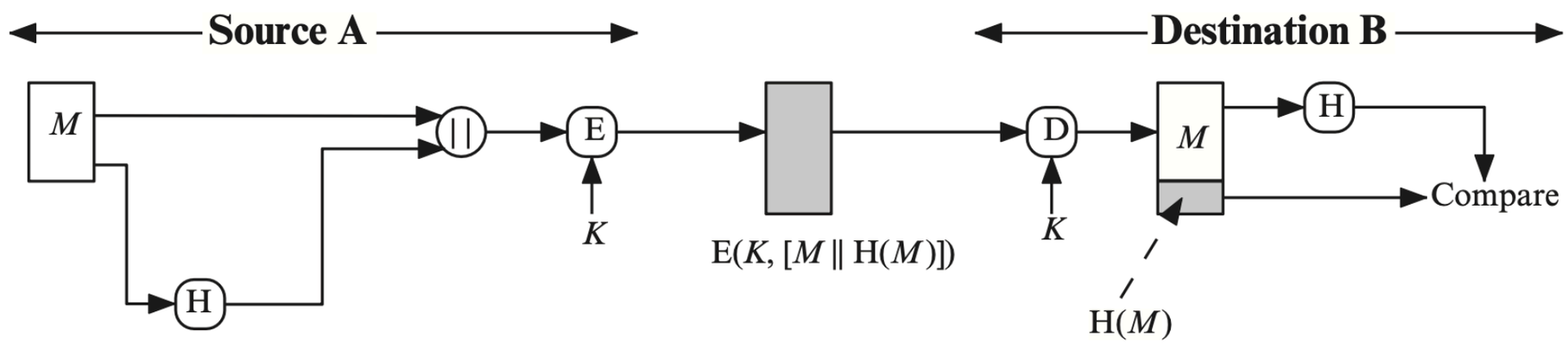




Man-in-the-middle attack

# Cryptographic Hash Functions

## Message Authentication

Using hash code to provide message authentication



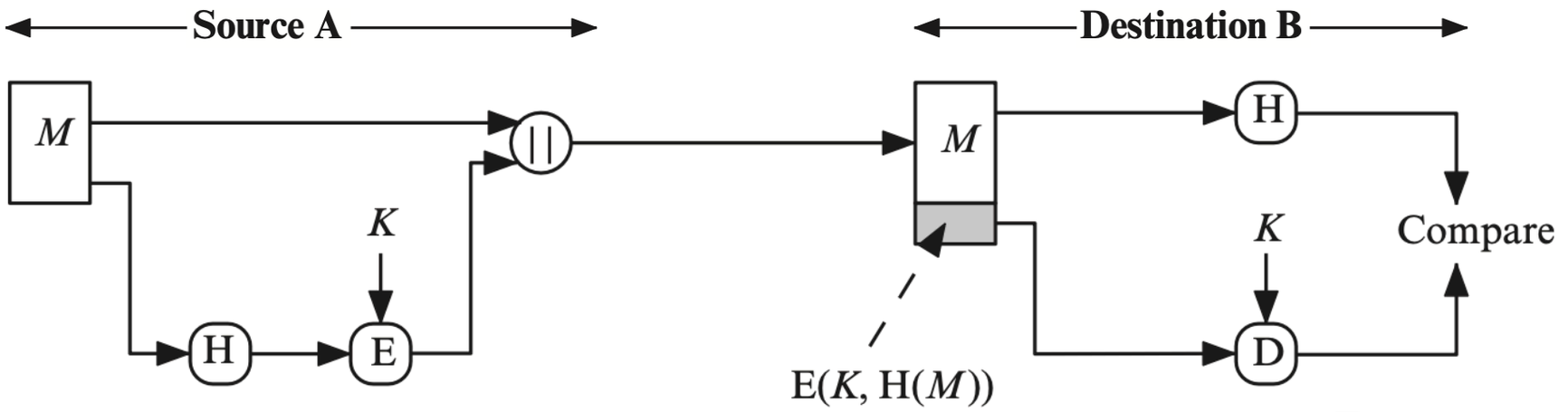

The message plus the hash code is encrypted using symmetric encryption.

- As A and B share the key, the message must have come from A and has not been altered.
- By encrypting the entire message plus the hash code, confidentiality is also provided.

# Cryptographic Hash Functions

## Message Authentication

Using hash code to provide message authentication



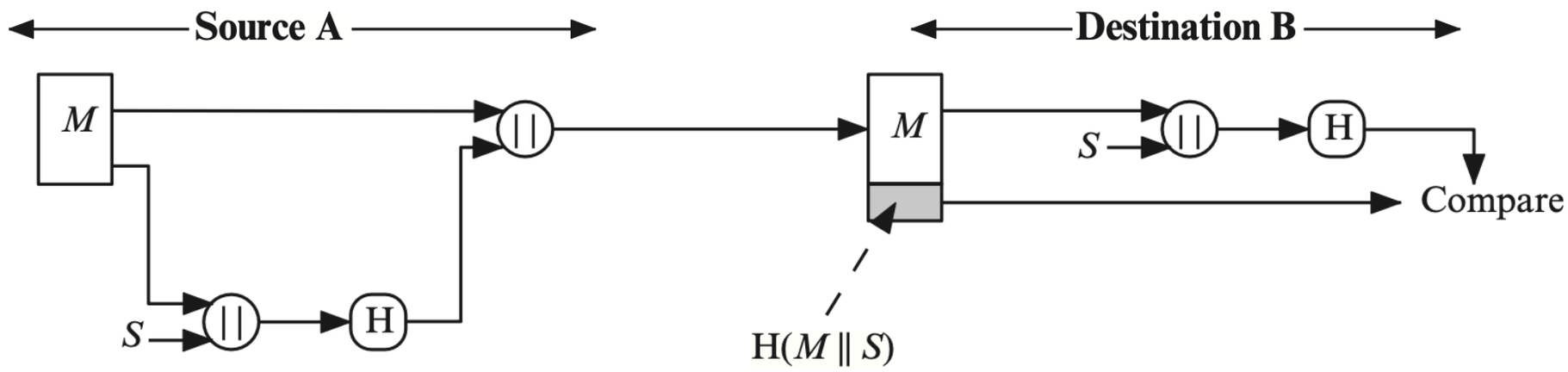

Only the hash code is encrypted using symmetric encryption.

- This reduces the processing burden.
- It does not provide confidentiality.

# Cryptographic Hash Functions

## Message Authentication

Using hash code to provide message authentication



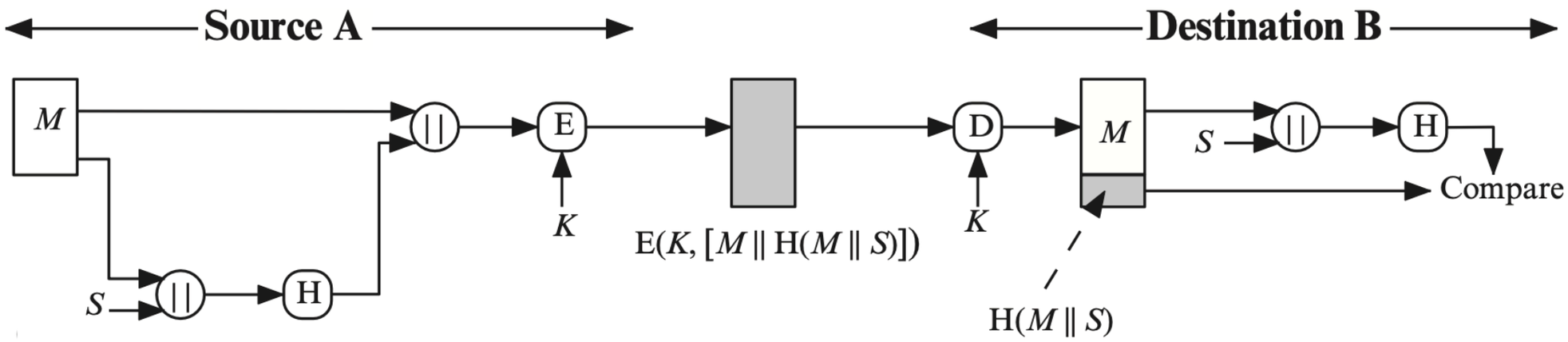

It uses a hash function for message authentication without encryption.

- It requires that two communicating parties share a common secret value **`S`**.
- The hash value is computed over the concatenation of **`M`** and **`S`**.

# Cryptographic Hash Functions

## Message Authentication

Using hash code to provide message authentication




It adds confidentiality to the previous approach by encrypting the entire message plus the hash code.

# Cryptographic Hash Functions

## Requirements & Security

### Terminology

For a hash value **`h = H(x)`**, **`x`** is called the **preimage** of **`h`**.

A **collision** occurs when **`x ≠ y`** and **`H(x) = H(y)`**.

- Since **`H`** is a many-to-one mapping, for any given **`h`**, there will be multiple possible preimages.
  - It maps a variable-length block (`n bits`) to a fixed-size value **`h`** (`b bits`). As typically `n > b`, on average, each `h` corresponds to $2^{n-b}$.
- For data integrity, collisions are undesired.

# Cryptographic Hash Functions

## Requirements & Security

### Security requirements for cryptographic use of hash functions

- **Variable input size:** `H` can be applied to a block of data of any size;
- **Fixed output size:** `H` produces a fixed-length output;
- **Efficiency:** `H(x)` is relatively easy to compute for any `x` (for software and hardware implementations);
- **Preimage resistant:** For any given hash value `h`, it is computationally infeasible to find `y` such that `H(y)=h`;
- **Second preimage resistant:** For any given block `x`, it is computationally infeasible to find `y≠x` with `H(y)=H(x)`;
- **Collision resistant:** It is computationally infeasible to find any pair `(x,y)` with `x≠y`, such that `H(x)=H(y)`;
- **Pseudorandomness:** Output of `H` meets standard tests for randomness and unpredictability.

# Cryptographic Hash Functions

## Brute-Force Attacks

### Preimage and Second Preimage Attacks

#### Objective

Find a value `y` such that `H(y)` is equal to a given hash value `h`.

#### Brute-force method

Pick values of y at random and try each value until a collision occurs.

- For an `m`-bit hash value, the level of effort is proportional to $2^m$;
- On average, an adversary would have to try $2^{m-1}$ values of `y` to find one that generates a given hash value `h`.

# Cryptographic Hash Functions

## Brute-Force Attacks

### Collision resistant attacks / Birthday attack

#### Objective

Find two messages or data blocks, `x` and `y`, that yield the same hash function:

**`H(x)=H(y)`**

#### Brute-force method

It relies on the birthday paradox in probability theory:

*In a group of 23 people, there is a 50% chance that at least two of them have the same birthday. Increasing the group to 50 people, the probability is over 97%.*

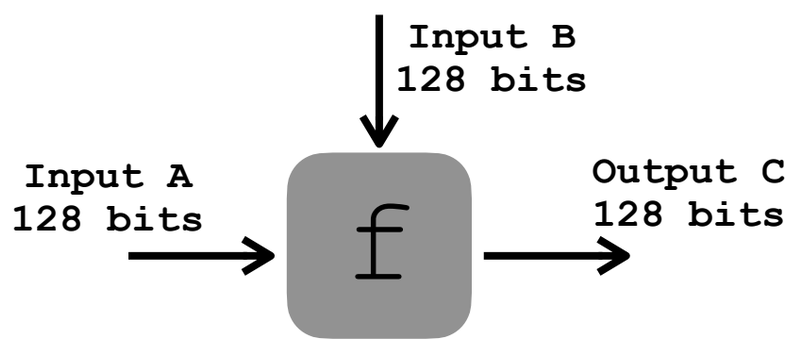
- When choosing random variables from a uniform distribution in the range `0` through *`N - 1`*, the probability that a repeated element is encountered exceeds `50%` after *`sqrt(N)`* choices have been made.
- Thus, for an m-bit hash value, by picking blocks randomly, it is expected to find two blocks with the same hash value within $2^{m/2}$ attempts.
  - For `128-bit` and `512-bit` hash functions, an attack should test $2^{64}$ and $2^{256}$ messages.

# Cryptographic Hash Functions

## Design

### Compression function

A widely used component in secure hash function design. It consists of taking two inputs and producing an `n-bit` output.




They are one-way functions that:

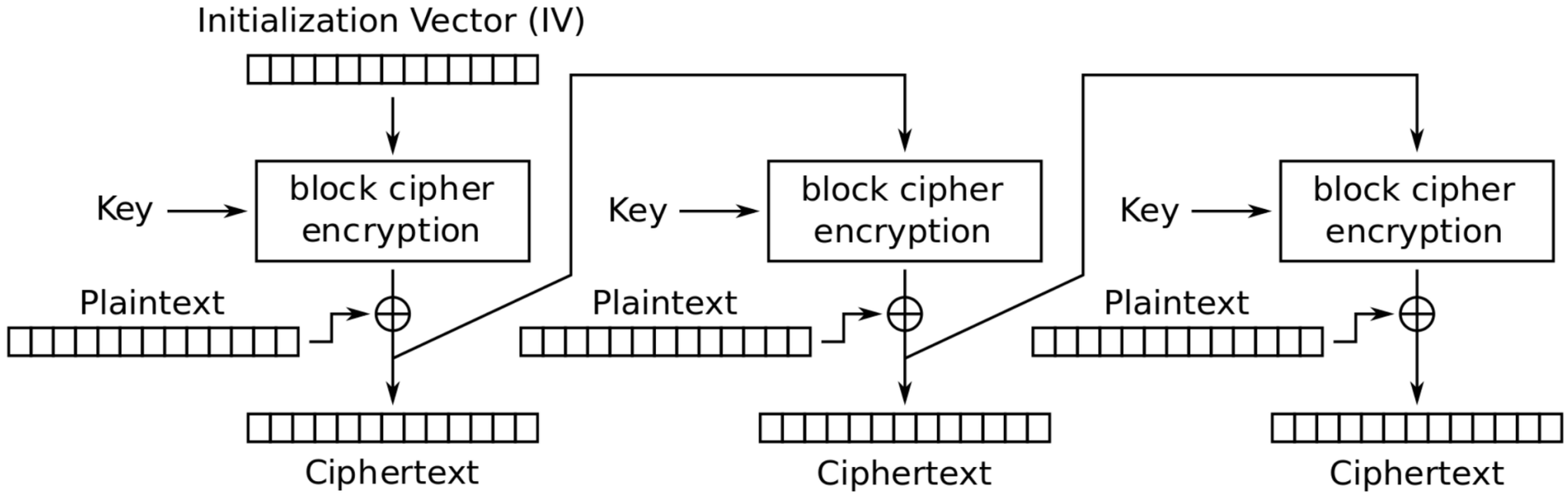
- Having the two inputs, it is easy to compute the output;
- It is infeasible to discover any of the inputs from the output;
- Having the output and one of the inputs, it is infeasible to find the second input;
- They are collision-resistant.

# Cryptographic Hash Functions

## Design

### Compression function

**Can we use chained block ciphers as hash functions?**



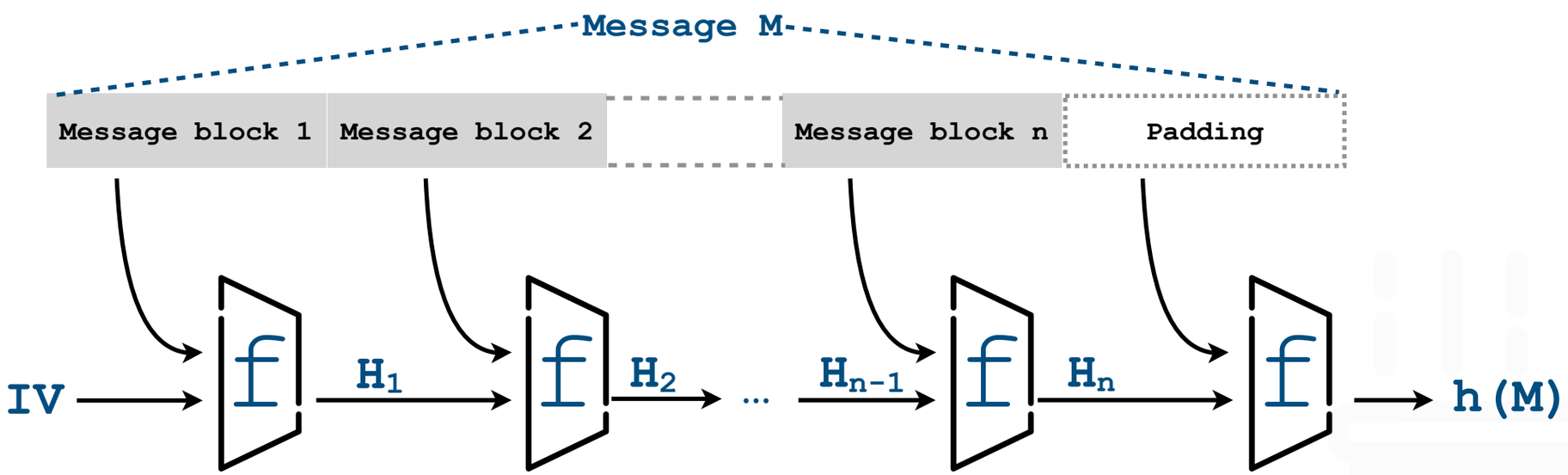

**Cipher Feedback (CFB)**

# Cryptographic Hash Functions

## Design

### *Merkle-Damgård* structure

Used by most of the current cryptographic hash functions, including **SHA**.




- The compression function chaining propagates the previous message hash;
- The `IV` is a fixed value defined by the algorithm;
- The padding should include information about the message size;
- If the compression function `f` is collision-resistant, so is the hash function.

# Cryptographic Hash Functions

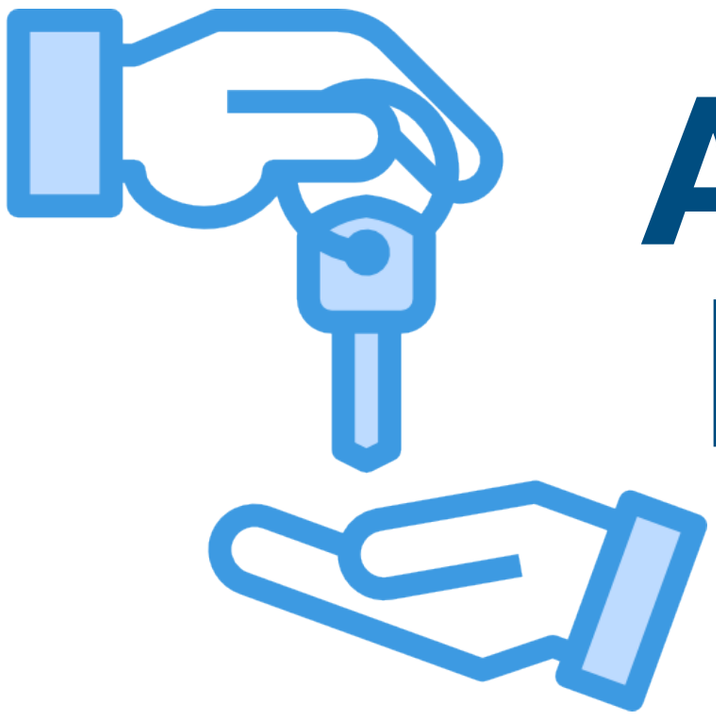
## Examples

Not recommended for applications with high-security requirements due to collisions already encountered.

| Algorithm | Block size (bits) | Rounds | Digest size (bits) |
|---|---|---|---|
| MD5 | 512 | 4 | 128 |
| SHA-1 | 512 | 4 | 160 |
| SHA-224 | 512 | 64 | 224 |
| SHA-256 | 1024 | 64 | 256 |
| SHA-384 | | 80 | 384 |
| SHA-512 | | | 512 |
| SHA3-224 | 1152 | 24 | 224 |
| SHA3-256 | 1088 | | 256 |
| SHA3-384 | 832 | | 384 |
| SHA3-512 | 576 | | 512 |

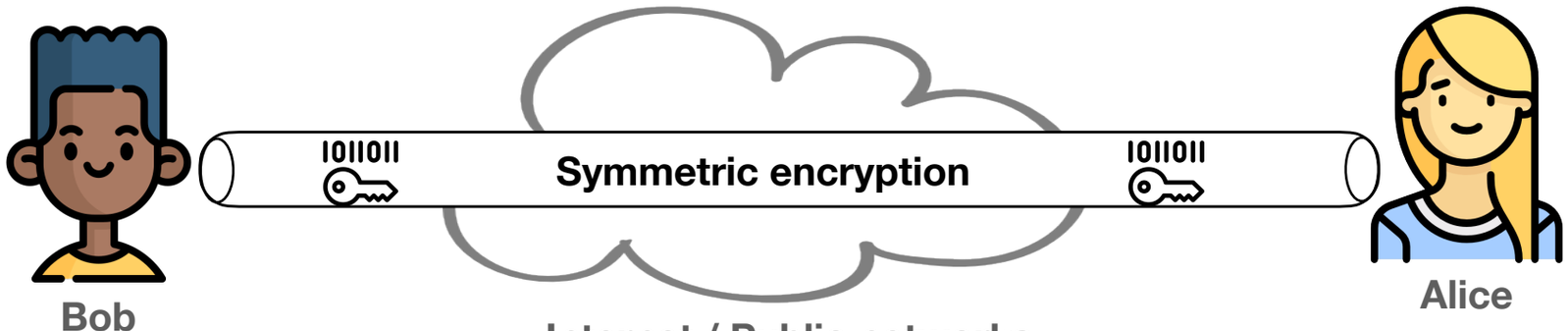
- SHA-2 Standard
- Considered secure for cryptographic application

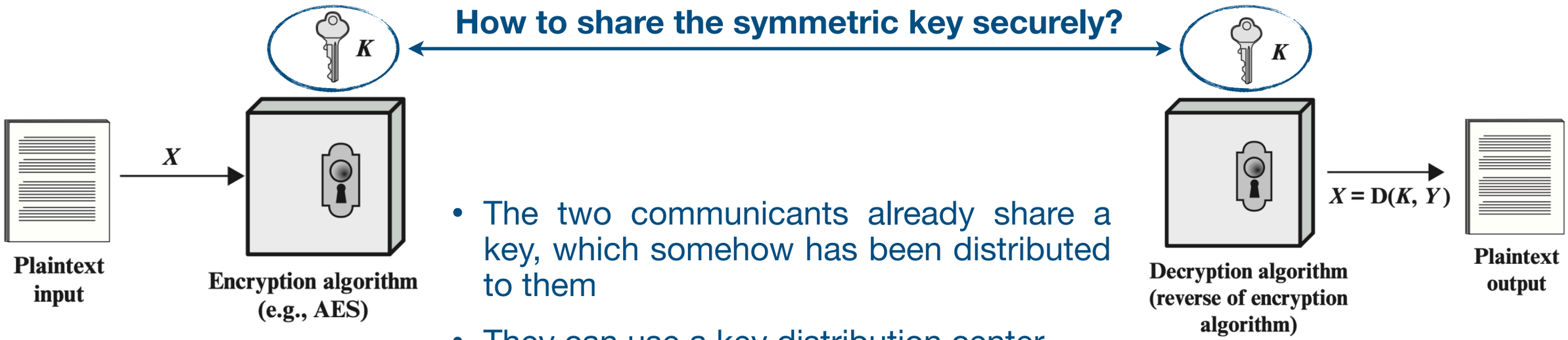
- New standard from 2015;
- Uses Keccak's algorithm;
- Based on the sponge-construction with a wide range of configuration parameters that allow to adjust the level of security/efficiency.



# Asymmetric Encryption

# Asymmetric Encryption




- The two communicants already share a key, which somehow has been distributed to them
- They can use a key distribution center

A third party has access to the key that should be known only by the two communicants.

# Asymmetric Encryption

## Principles

- Asymmetric algorithms rely on one key for encryption and a different but related key for decryption. They have two main characteristics:
  - It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key;
  - Either of the two related keys can be used for encryption, with the other used for decryption.
- A tighter specification is used for *one-way functions with secret*: **`Trapdoor Permutation:`**
  - Injective function that has an efficient calculation algorithm;
  - Whose the inverse calculation is an intractable problem;
  - But anyone with additional information (*i.e.*, the secret) can calculate its inverse.

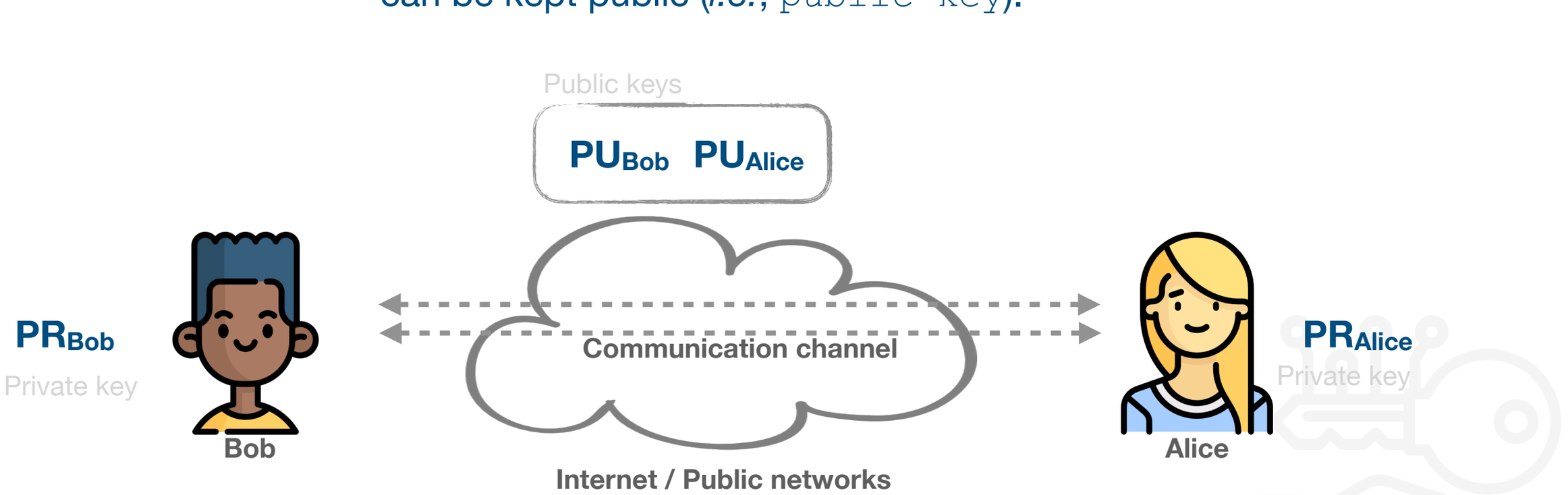
Much of the theory of public-key crypto systems is based on number theory.

# Asymmetric Encryption

## Principles

By using different keys for the encryption and decryption operation, it is possible to overcome the problem of pre-distribution of keys.

One key must be kept secret (*i.e.*, `private key`), while the second can be kept public (*i.e.*, `public key`).




It also enhances confidentiality and authentication-related operations.

# Asymmetric Encryption

## Terminology

**Also known as public key cryptography.**

**Asymmetric cryptography:** A cryptosystem that uses separate keys, one to encrypt or digitally sign data and one for decrypting the data or verifying the digital signature.

**Asymmetric keys:** Two related keys, a public and a private key, that are used to perform complementary operations, e.g., encryption and decryption or signature and signature verification.

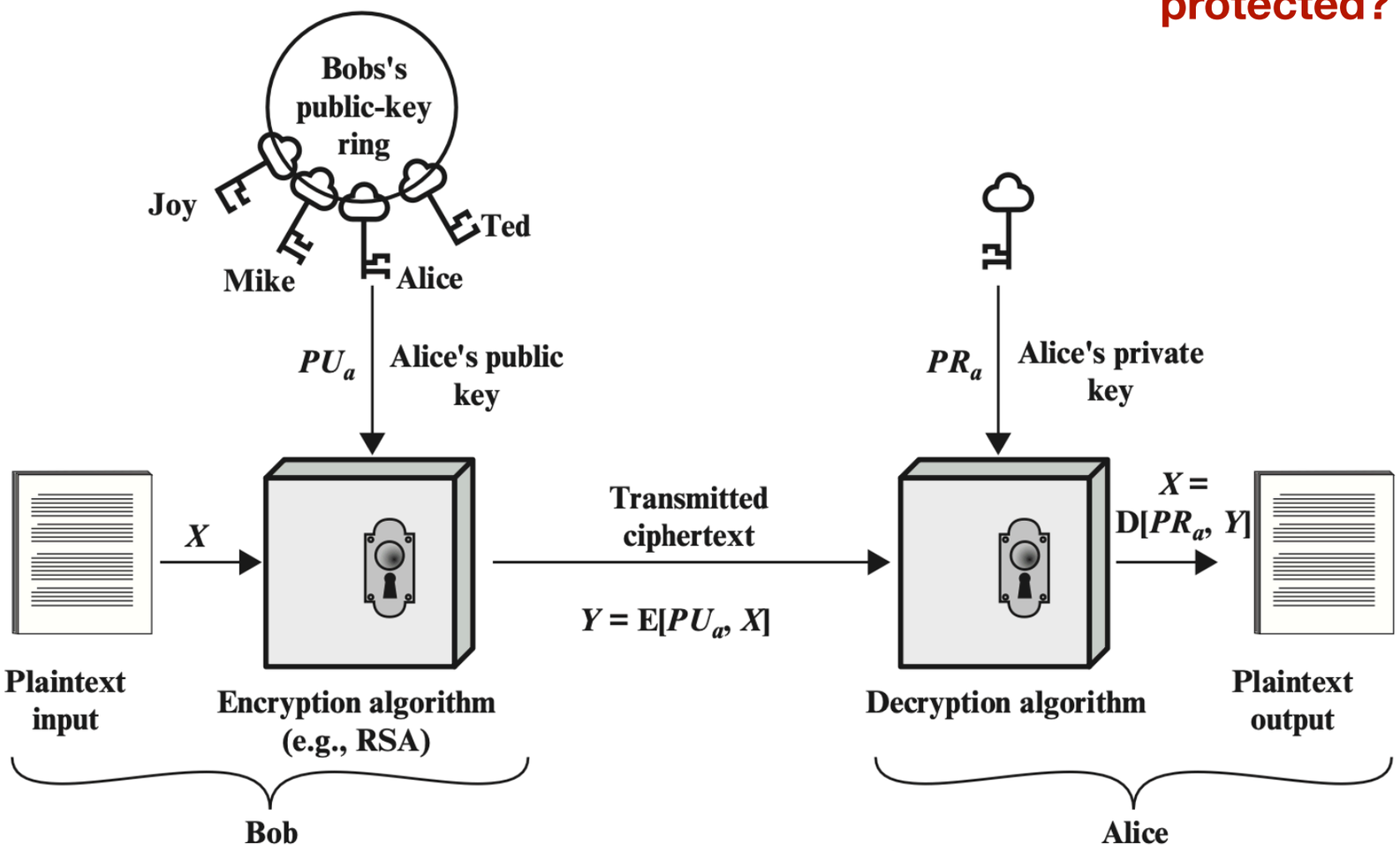
**Public key certificate:** A digital document issued and digitally signed by the private key of a Certification Authority that binds the name of a subscriber to a public key.

**Public key Infrastructure (PKI):** A set of policies, processes, server platforms, and software used to issue, maintain, and revoke certificates and public-private key pairs.

# Asymmetric Encryption

## Encryption with public key

**Which security properties are protected?**

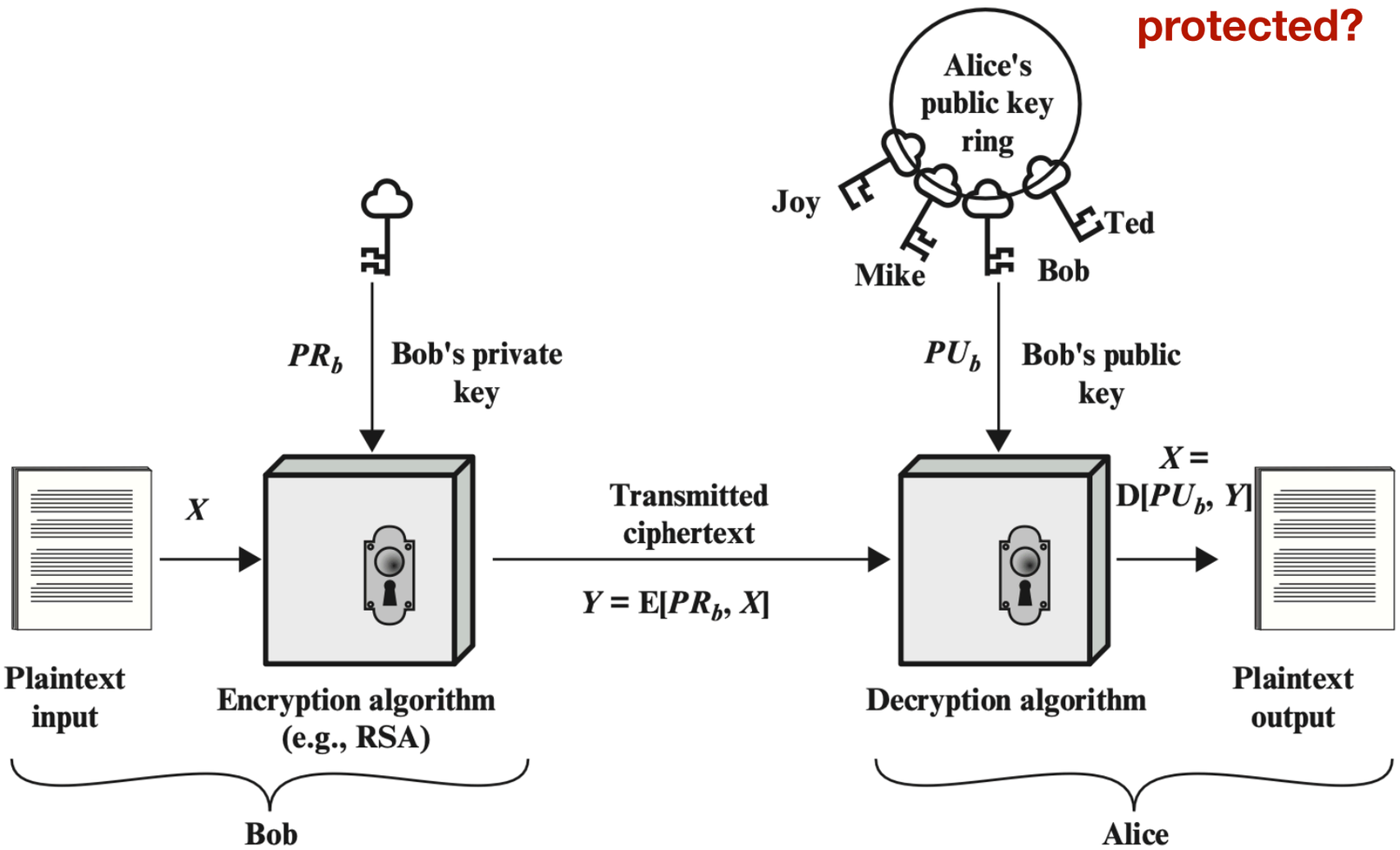



Source: Stallings, W. & Brown, L. Cryptography and Network Security: Principles and Practice

# Asymmetric Encryption

## Encryption with private key

**Which security properties are protected?**




Source: Stallings, W. & Brown, L. Cryptography and Network Security: Principles and Practice

# Asymmetric Encryption

## Properties

- Although better than symmetric cryptosystems, public-key cryptography requires protocols for key distribution, generally involving a central agent.

- One common misconception is that public-key encryption is more secure from cryptanalysis than symmetric encryption.

- Another misconception is that asymmetric encryption made symmetric encryption obsolete.

  - For the same level of security, asymmetric encryption is several orders of magnitude less efficient than symmetric encryption.

  - For this reason, they are typically used in conjunction and not alternatively.

- Public-key algorithms are based on mathematical functions rather than on substitution and permutation.

# Asymmetric Encryption

## Requirements

- It is computationally easy for a party **B** to generate a key pair ($\mathbf{PU_b}$, $\mathbf{PR_b}$)

- It is computationally easy for a sender **A**, knowing the public key and the message to be encrypted, **M**, to generate the corresponding ciphertext.

$$\mathbf{C = E(PU_b, M)}$$

- It is computationally easy for the receiver **B** to decrypt the resulting ciphertext using the private key to recover the original message.

$$\mathbf{M = D(PR_b, C) = D[PR_b, E(PU_b, M)]}$$

- It is computationally infeasible for an adversary, knowing the public key, $\mathbf{PU_b}$, to determine the private key, $\mathbf{PR_b}$.

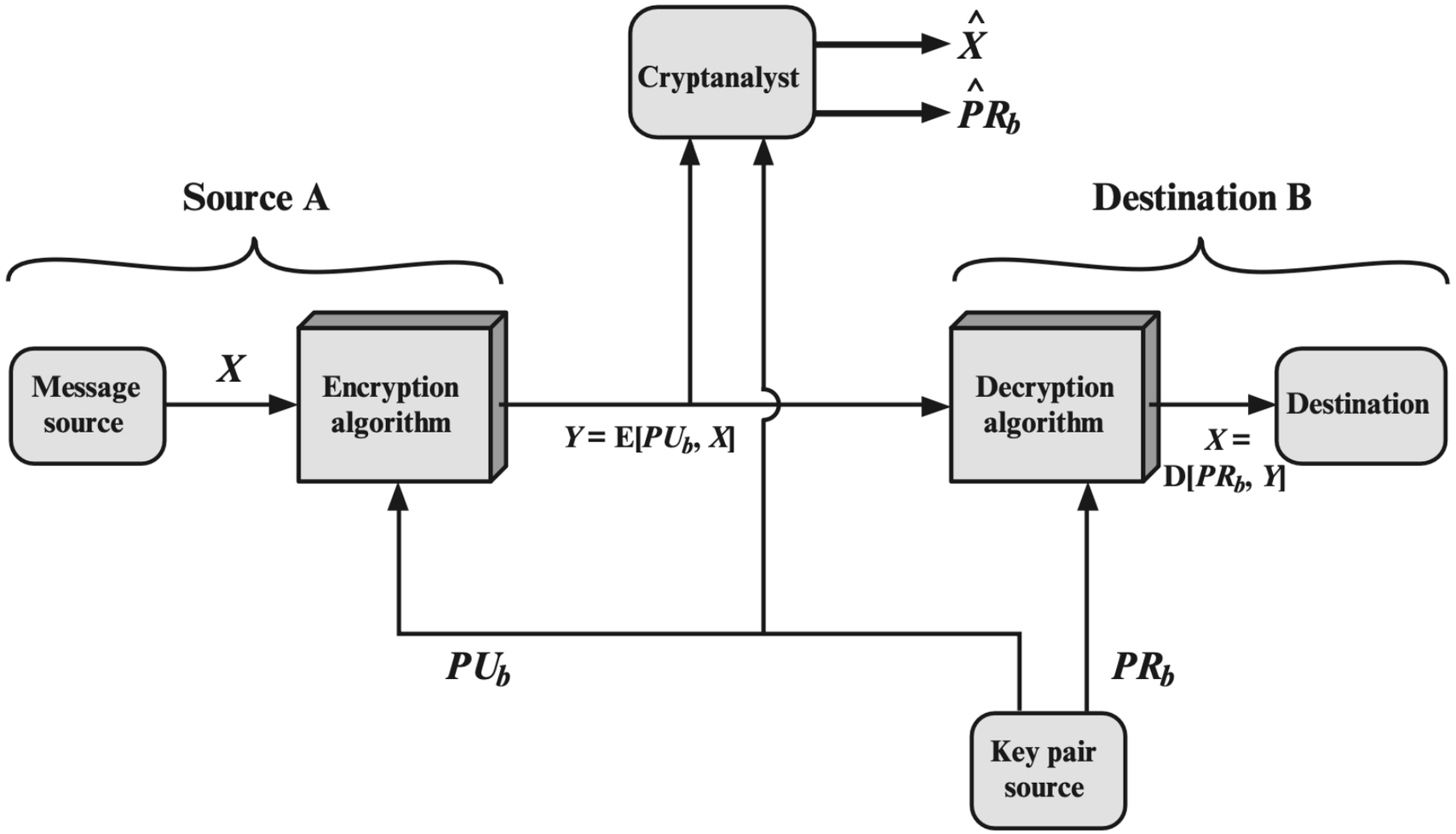
- It is computationally infeasible for an adversary, knowing the public key, $\mathbf{PU_b}$, and a ciphertext, **C**, to recover the original message, **M**.

- The two keys can be applied in either order.

# Asymmetric Encryption

## Applications

### Public-Key Cryptosystem: Confidentiality

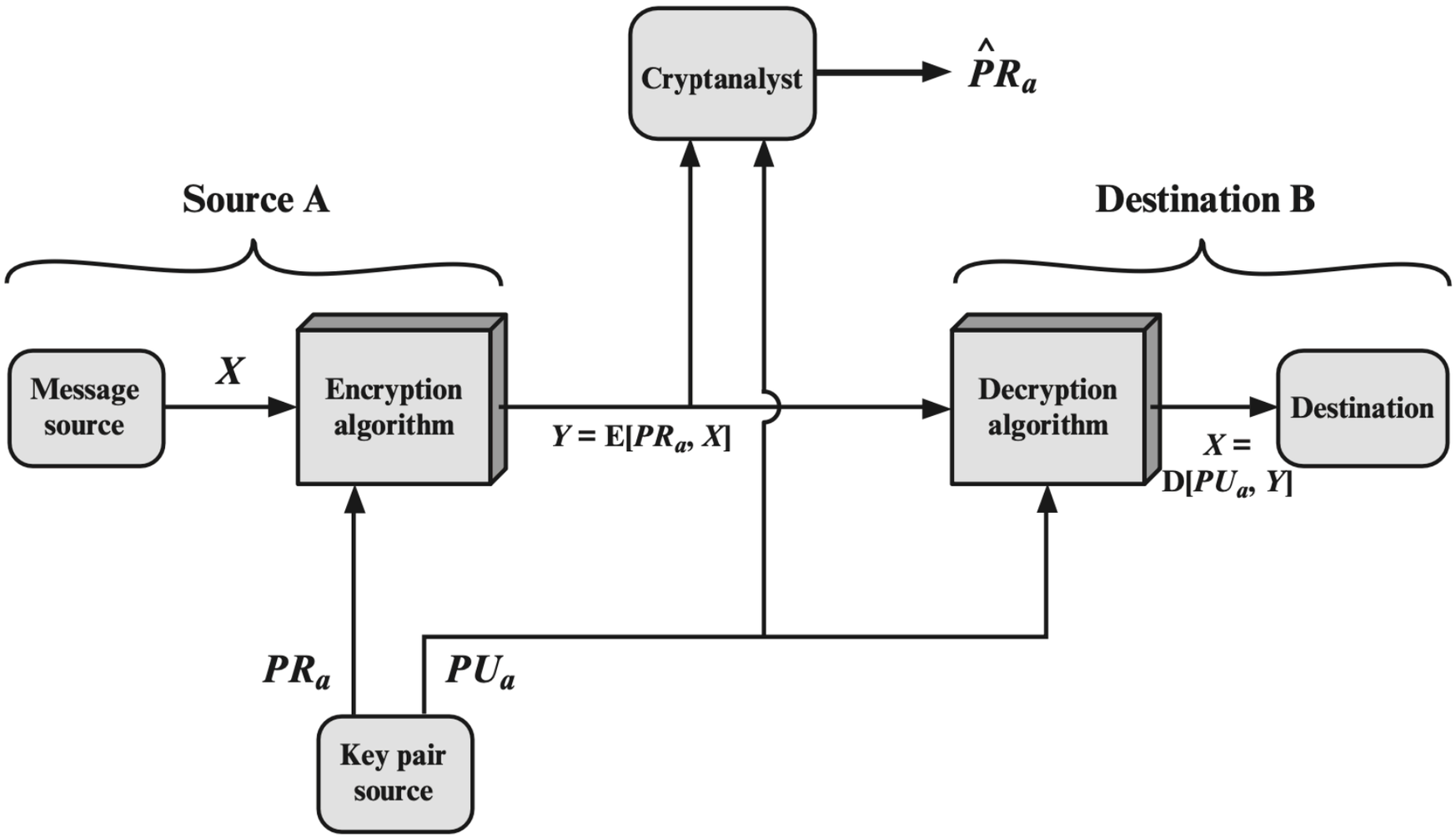



Source: Stallings, W. & Brown, L. Cryptography and Network Security: Principles and Practice

# Asymmetric Encryption

## Applications

### Public-Key Cryptosystem: Authentication

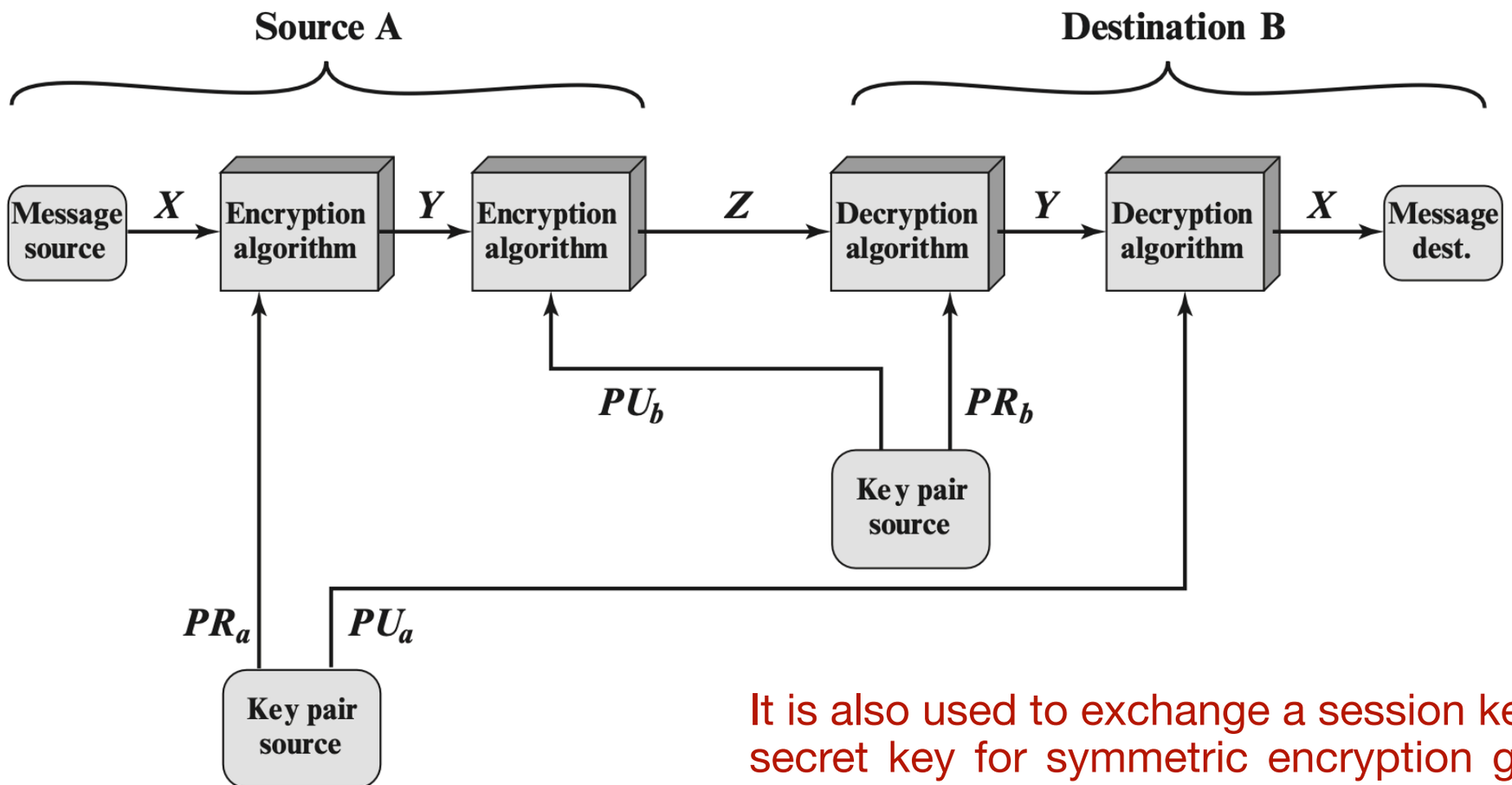



Source: Stallings, W. & Brown, L. Cryptography and Network Security: Principles and Practice

# Asymmetric Encryption

## Applications

### Public-Key Cryptosystem: Authentication & Confidentiality



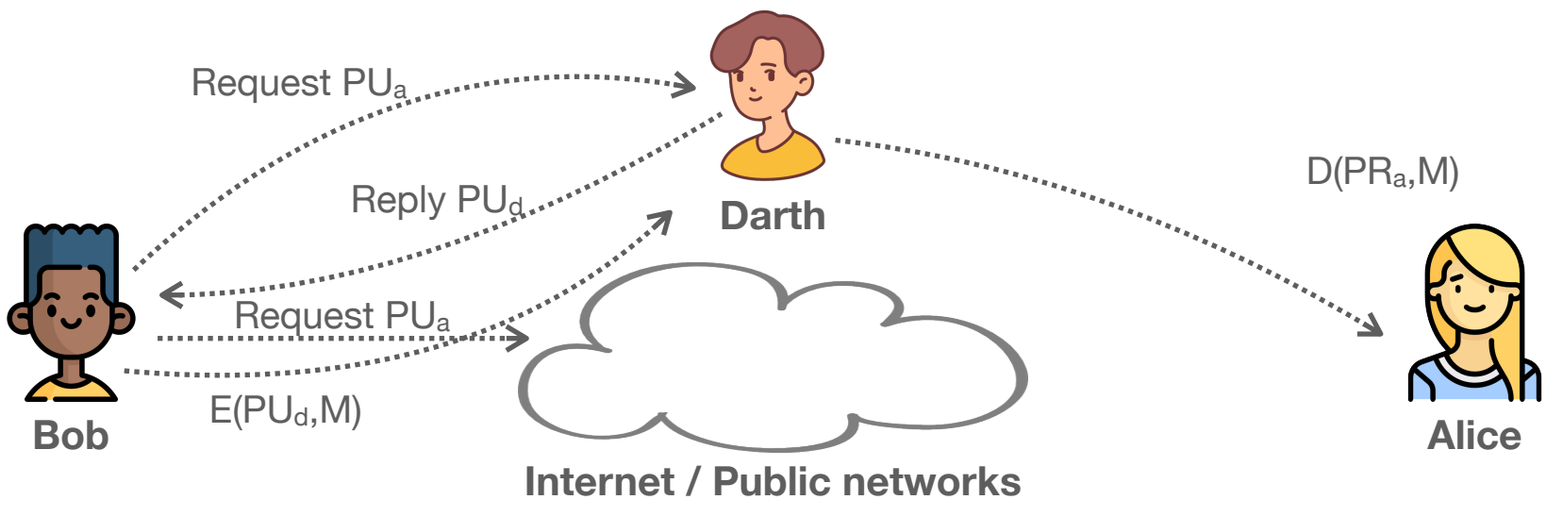

It is also used to exchange a session key, which is a secret key for symmetric encryption generated for use in a particular transaction or session.

Source: Stallings, W. & Brown, L. Cryptography and Network Security: Principles and Practice

# Asymmetric Encryption

## Man-in-the-Middle (*MitM*) attack

Bob wants to send Alice an encrypted message.




1. Bob sends a request for Alice's public key
2. Darth intercepts the request and replies with his own public key
3. Darth intercepts the message encrypted with $PU_d$ and reads it using $PR_d$
4. Then, encrypts the message with $PU_a$ and sends it to Alice
5. Alice decrypts the message with $PR_a$ without the knowledge that Darth also had access to the message

It requires a trust mechanism for the association between key pairs and identities.

# Asymmetric Encryption

## Applications

### Digital signature

The main contribution of asymmetric cryptography was to allow the definition of a digital analog of the concept of signing a document.

In general, a digital signature can be identified as a “supplement” to the message that allows for verifying:

- **Integrity**: the message is not modified after signing;
- **Authenticity**: the signer’s identity can be confirmed;
- **Non-repudiation**: it is possible to prove the signer’s identity.

A signature scheme comprises two operations:

- **Signature production**: the process by which the *signer* generates the signature $x=Sig^s(M)$, which he attaches to the message
  - the signed message consists of a pair $(M,x)$;
- **Signature verification**: the process in which the verifier confirms that the origin of the message $M$ is $S$, i.e., $Ver^s(M',x)==true$.

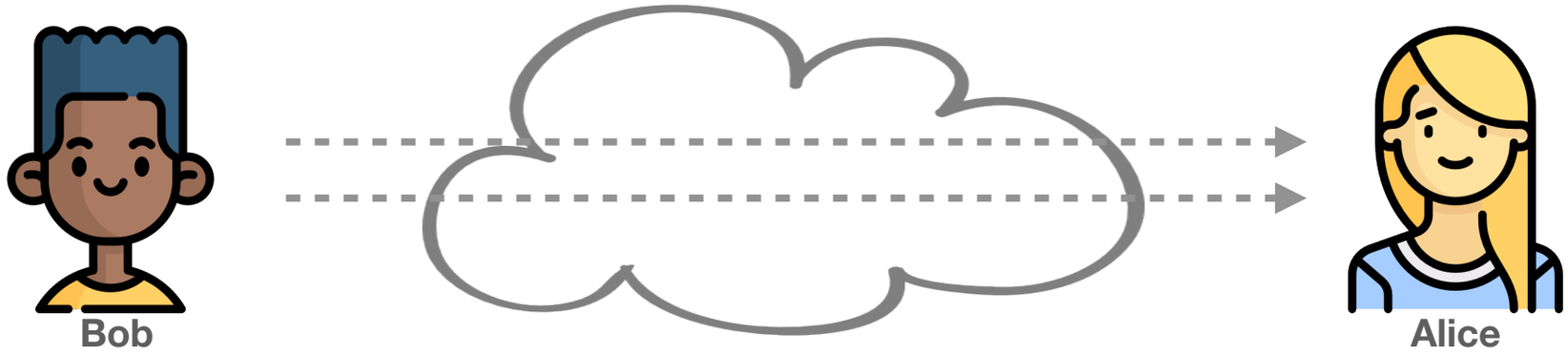
# Asymmetric Encryption

## Applications

### Digital signature - a practical example

**Without certification of the relationship between the keys and the signer's identity, this scheme is also vulnerable do MitM attacks.**

- Public key: `PUb`
- Private key: `PRb`




- Public key: `PUa`
- Private key: `PRa`

For a message `M`:

- `H = hash(M)`
- Signature `S = E(PRb,H)`
- Send `C = E(PUa, M||H)`

By using the message's hash, the transmitted data is reduced, and Alice can verify its integrity.

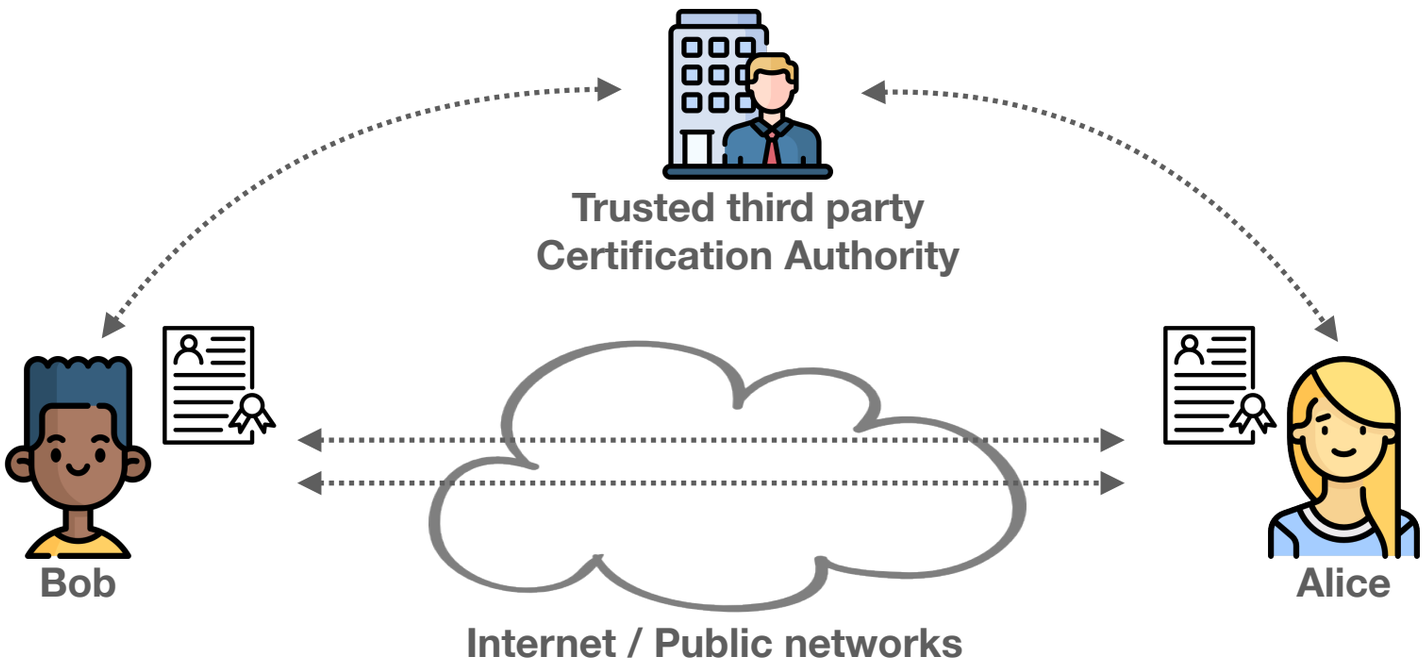
For the ciphertext `C`:

- `D(PRa, C) = (M||H)`
- Verify `S:` `D(PUb,H)== h(M)`

# Asymmetric Encryption

## Applications

### Digital signature - key certification




- All agents have the public key of a trusted agent, i.e., the Certification Authority (CA), obtained via a secure channel.
- The CA guarantees (by digital signing) the association between the public key and the agent's identity via a public key certificate.
- Any agent can verify a certificate's signature, thus attesting to the identity's validity.

# Digital signatures
## DSS/DSA

- *Digital Signature Standard (DSS)*: The NIST standard for digital signature algorithm introduced in 1991,
  - Based on the ElGamal scheme.

- *Digital Signature Algorithm* (DSA)
  - Uses the Secure Hash Algorithm (SHA) to generate a digest of variable-size messages.
  - Designed to provide only the digital signature function
  - Designed to be more efficient than the RSA signature scheme.
    - Suitable for constrained devices, *e.g., smartcards.*
    - However, the verification process is comparatively heavier.

# Public Key Cryptography

## Key Distribution

### Public Announcement of Public Keys

- Any participant can send its public key to any other participant or broadcast the key to the community at large.


- Although convenient, it has a major weakness:

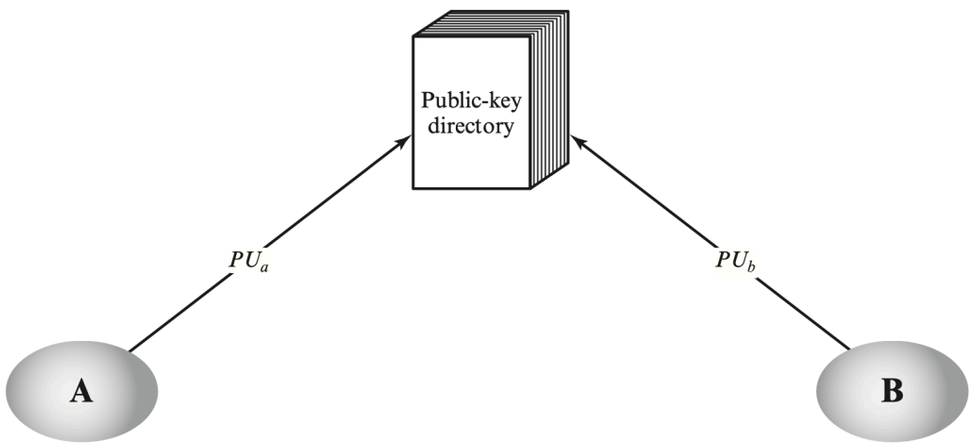
**Anyone can forge such a public announcement and impersonate a participant.**

# Public Key Cryptography

## Key Distribution

### Publicly Available Directory

- A publicly available dynamic directory maintained by a trusted entity or organisation.
- The directory stores a `{name, public key}` entry for each participant.
- The participants register or replace their register using a secure channel.




- It enhances security. However,  it still has a critical weakness:

**An adversary might tamper with the records kept by the central entity**

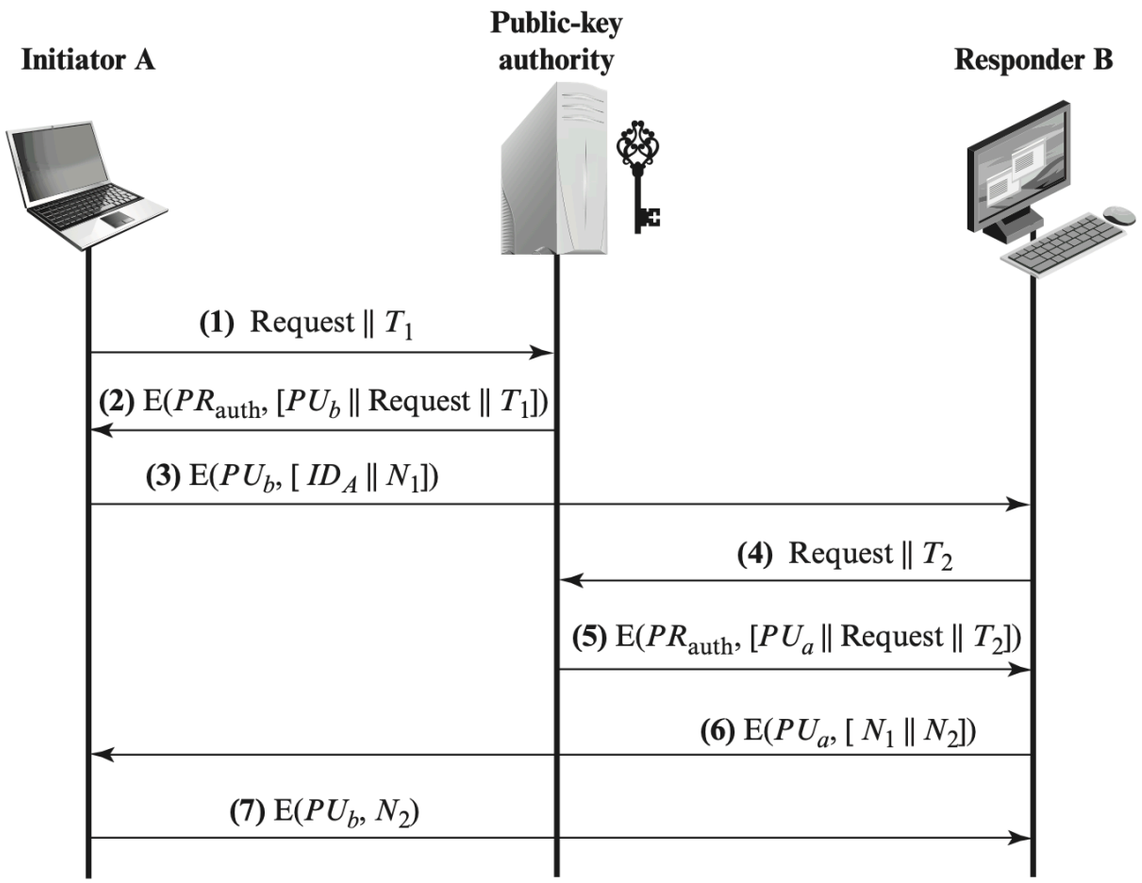
# Public Key Cryptography

## Key Distribution

### Publicly-Key Authority

- Provides tighter control over the distribution of public keys from the directory.
  - The central authority maintains a dynamic directory of public keys of all participants.
  - Each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key.

N is a *nonce* used to identify a message uniquely.




Source: Stallings, W. & Brown, L. Cryptography and Network Security: Principles and Practice

The initial five messages need to be used only infrequently since a participant can save other's public key for future use, *i.e.*, caching.

**The public authority must be contacted to provide a public key for every other user to be contacted.**

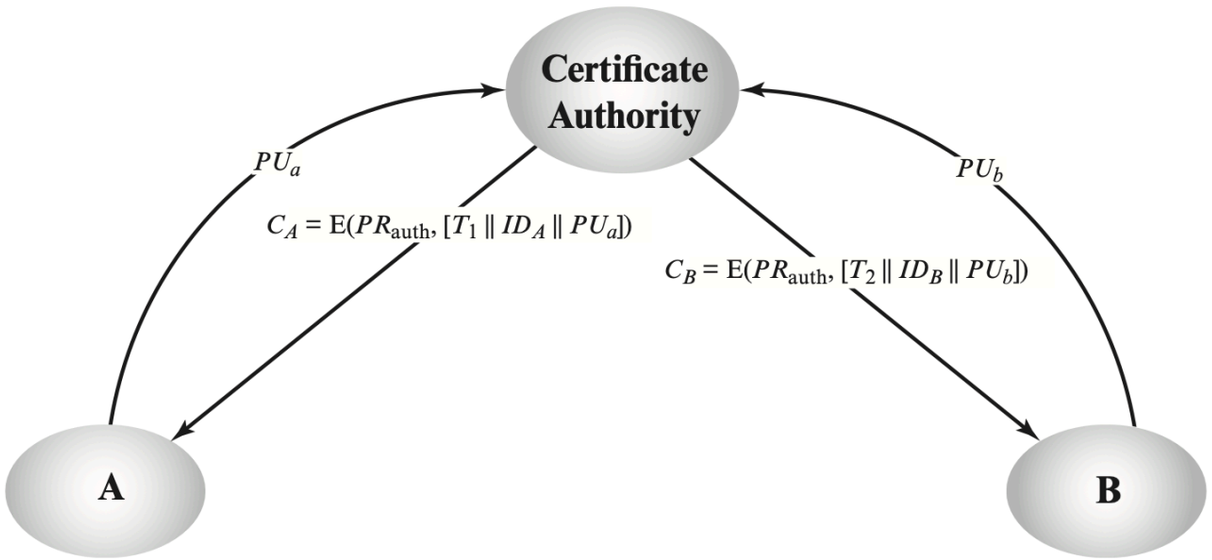
# Public Key Cryptography

## Key Distribution

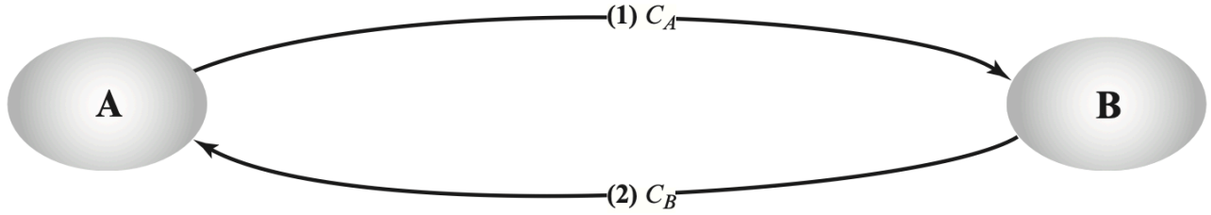
### Public-Key Certificates

- Used to reliably exchange keys among participants without contacting a public-key authority.
- Basically, a certificate consists of a public key, an identifier of the owner, and the whole block signed by a trusted third party.

**Obtaining certificates from CA**




**Exchanging certificates**




Source: Stallings, W. & Brown, L. Cryptography and Network Security: Principles and Practice

# Public Key Cryptography

## Digital Certificates

An electronic document or container file that contains a key value and identifying information about the entity that controls the key.

- They are often issued and digitally signed by with the private key of a third party, usually a Certificate Authority (CA).
- A digital signature is attached to the certificate's container file to certify its origin and integrity.
- Unlike digital signatures, which are used to authenticate messages' origin, digital certificates authenticate the cryptographic key that is embedded in the certificate.
- The X.509 Standard is the most used format of public key certificates.
  - Issued by the International Telecommunication Union (ITU), it is currently in version 3.
  - Used by:
    - IP Security (IPSec)
    - Transport Layer Security (TLS)
    - Secure/Multipurpose Internet Mail Extension (S/MIME)

# Public Key Cryptography

## Digital Certificates

| X.509 v3 Certificate structure | |
|---|---|
| Version | |
| Certificate Serial Number | → An integer value unique within the issuing CA |
| • Algorithm ID<br>• Algorithm ID<br>• Parameters | → The algorithm used to sign the certificate and the associated parameters |
| Issuer Name | |
| • Validity<br>• Not Before<br>• Not After | → Two dates defining its the period of validity |
| Subject Name | → The name of the entity to which this certificate refers |
| Subject Public-Key Information<br>• Public-Key Algorithm<br>• Parameters<br>• Subject Public Key | → The subject's public key, plus the algorithm identifier, and its parameters |
| Issuer Unique Identifier (Optional) | |
| Subject Unique Identifier (Optional) | |
| Extensions (Optional)<br>• Type<br>• Criticality<br>• Value | |
| Certificate Signature Algorithm | |
| Certificate Signature | → The CA signature for the entire certificate |



Source: Whitman, M. & Mattord, H. Principles of Information Security

# Public Key Cryptography

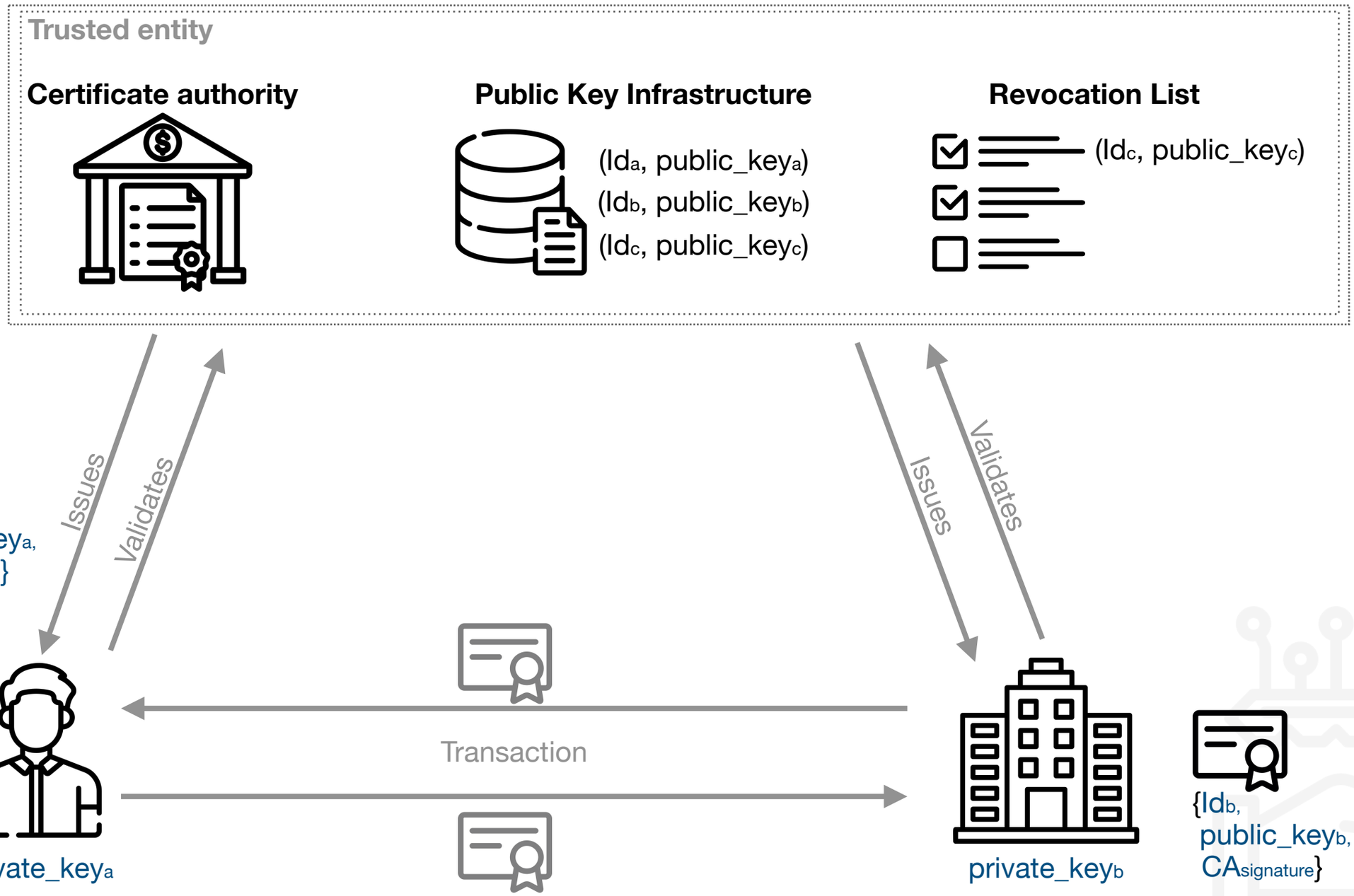
## Public Key Infrastructure (PKI)

> The architecture, organisation, techniques, practices, and procedures that collectively support the implementation of a certificate-based public key cryptographic system.
>
> NIST SP 800-53 Rev.5

- It allows applications to implement mechanisms that support:
  - Authentication: Individuals, organisations, and WEB servers can validate the identity of each party in a remote transaction.
  - Integrity: Content signed by the certificate is known not to have been altered while in transit from host to host or server to client.
  - Confidentiality: Information can be protected from being intercepted during transmission.
  - Authorisation: The validated identity of users and programs can enable authorisation rules that remain in place during a transaction.
  - Non-repudiation: By using digital signatures, parties of a digital transaction can be held accountable for their actions.

# Public Key Cryptography

## Digital Certificates



University of Minho
School of Engineering

# Distributed Data Processing Environments

## Bachelor in Data Science

**João Marco Silva**

Department of Informatics
joaomarco@di.uminho.pt

2024/2025