Systems' Security | *Segurança de Sistemas*

Introduction to Cryptography

Miguel Frade

POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Overview

Learning Objectives

After this chapter, you should be able to:

1. Explain the difference between cryptographic algorithms categories
2. Present an overview of the main concepts of symmetric cryptography
3. Explain the difference between cryptanalysis and brute-force attack

Introduction

Cryptography

is the study of techniques for secure communication in the presence of third parties called adversaries

Cryptography

is the study of techniques for secure communication in the presence of third parties called adversaries

Cryptography aims to develop techniques and algorithms to provide these security services:

- confidentiality
- data integrity
- authentication
- non-repudiation

Cryptographic algorithms can be characterized into **three independent dimensions**:

1. type of operations used for transforming plaintext to ciphertext:
    - substitution
    - transposition
    - both substitution and transposition

Cryptographic algorithms can be characterized into **three independent dimensions**:

1. type of operations used for transforming plaintext to ciphertext:
   - substitution
   - transposition
   - both substitution and transposition

2. number of keys used:
   - one key: symmetric, single-key, secret-key, or conventional encryption
   - two keys: asymmetric, two-key, or public-key encryption

Cryptographic algorithms can be characterized into **three independent dimensions**:

1. type of operations used for transforming plaintext to ciphertext:
    - substitution
    - transposition
    - both substitution and transposition

2. number of keys used:
    - one key: symmetric, single-key, secret-key, or conventional encryption
    - two keys: asymmetric, two-key, or public-key encryption

3. way in which the plaintext is processed:
    - block cipher
    - stream cipher

Cryptographic algorithms can be divided into three families

- Symmetric algorithms
- Asymmetric (or public key) algorithms
- Hash algorithms | *algoritmos de síntese*

Symmetric Encryption

Symmetric encryption:

- also referred to as conventional encryption or single-key encryption
- the only type of encryption until the 1970s (before the development of public-key encryption)

Symmetric encryption ingredients:

1. **plaintext** – the original intelligible message or data that is fed into the algorithm as input

Symmetric encryption ingredients:

1. **plaintext** – the original intelligible message or data that is fed into the algorithm as input
2. **encryption algorithm** – algorithm that performs various substitutions and transformations on the plaintext

Symmetric encryption ingredients:

1. **plaintext** – the original intelligible message or data that is fed into the algorithm as input
2. **encryption algorithm** – algorithm that performs various substitutions and transformations on the plaintext
3. **secret key** – another input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time.

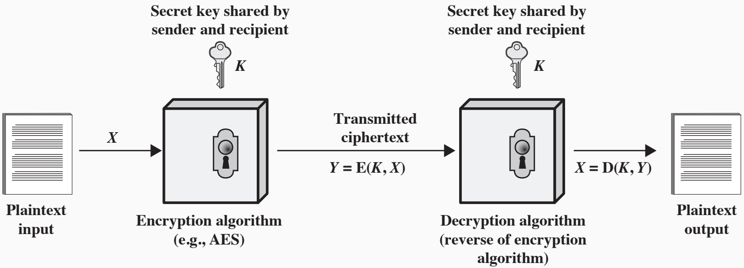
Symmetric encryption ingredients:

1. **plaintext** – the original intelligible message or data that is fed into the algorithm as input
2. **encryption algorithm** – algorithm that performs various substitutions and transformations on the plaintext
3. **secret key** – another input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time.
4. **ciphertext** – the scrambled message produced as output that depends on the plaintext and the secret key

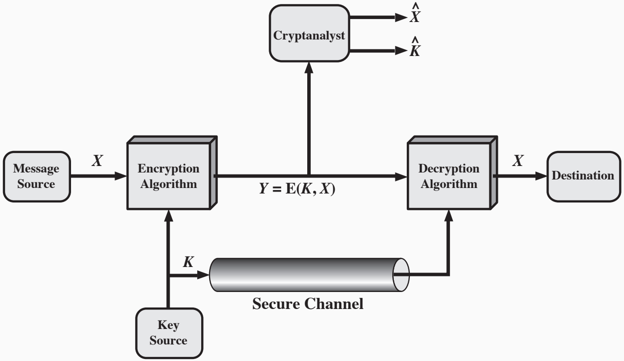
Symmetric encryption ingredients:

1. **plaintext** – the original intelligible message or data that is fed into the algorithm as input
2. **encryption algorithm** – algorithm that performs various substitutions and transformations on the plaintext
3. **secret key** – another input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time.
4. **ciphertext** – the scrambled message produced as output that depends on the plaintext and the secret key
5. **decryption algorithm** – takes the ciphertext and the secret key and produces the original plaintext.

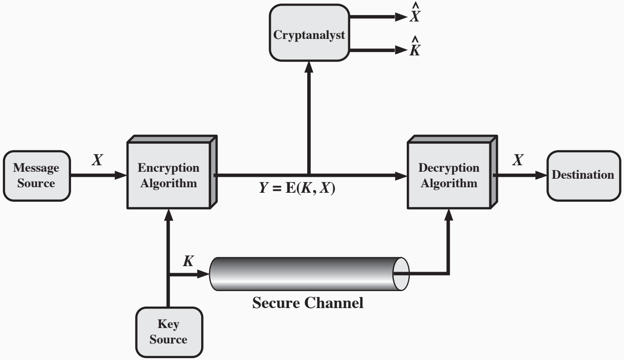
Simplified model of symmetric encryption



Model of symmetric encryption



Model of symmetric encryption



Short representation

$$A \rightarrow B : \quad E_{ks}(M)$$

There are two general approaches to attack a conventional encryption scheme:

1. **Cryptanalysis** – relies on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.

There are two general approaches to attack a conventional encryption scheme:

1. **Cryptanalysis** – relies on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.
2. **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
   - The analyst must be able to recognize plaintext as plaintext.

open algorithms – reverse the encryption process must rely only on the key value and not the fact of knowing the used algorithm

open algorithms – reverse the encryption process must rely only on the key value and not the fact of knowing the used algorithm

An encryption scheme is said to be computationally secure against brute-force attacks if

- **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

open algorithms – reverse the encryption process must rely only on the key value and not the fact of knowing the used algorithm

An encryption scheme is said to be computationally secure against brute-force attacks if

- **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

These principles also apply to other cryptographic algorithms.

Advantages

- Computationally efficient

Disadvantages

Advantages

- Computationally efficient
- Easy to implement in hardware

Disadvantages

Advantages

- Computationally efficient
- Easy to implement in hardware

Disadvantages

- Difficult to distribute keys safely

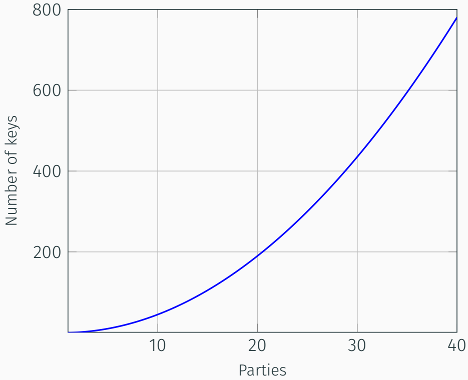
Advantages

- Computationally efficient
- Easy to implement in hardware

Disadvantages

- Difficult to distribute keys safely
- Number of keys required for independent communication between N parties

Number of keys required for independent communication between N parties



$$\text{Number of keys} = \frac{N(N-1)}{2}$$

Asymmetric Ciphers

Asymmetric algorithms

- use a key pair:
  - Private key (k_R)
  - Public key (k_U)

Asymmetric algorithms

- use a key pair:
  - Private key (k_R)
  - Public key (k_U)
- the k_R and k_U keys are mathematically related
  - but it's computationally infeasible derive the private key from the public key

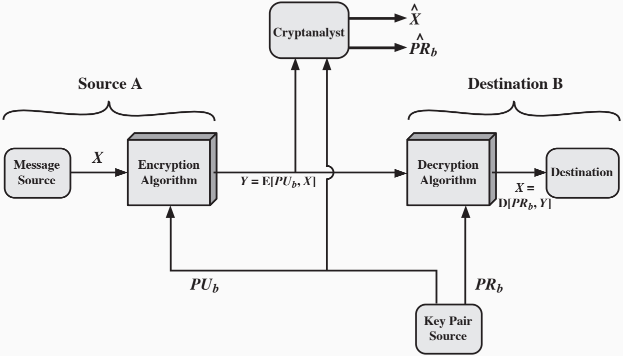
Asymmetric algorithms

- use a key pair:
  - Private key (k_R)
  - Public key (k_U)
- the k_R and k_U keys are mathematically related
  - but it's computationally infeasible derive the private key from the public key
- these keys are used to perform complementary operations:
  - encryption / decryption
  - signature generation / signature verification

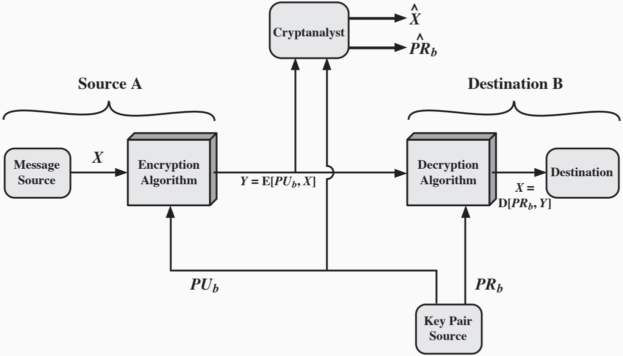
Asymmetric algorithms solve some problems found in symmetric algorithms:

- confidentiality is not required for key distribution of public keys
  - these algorithms are used to safely distribute symmetric keys
- perform digital signatures to ensure non-repudiation

Model of asymmetric encryption – **confidentiality**



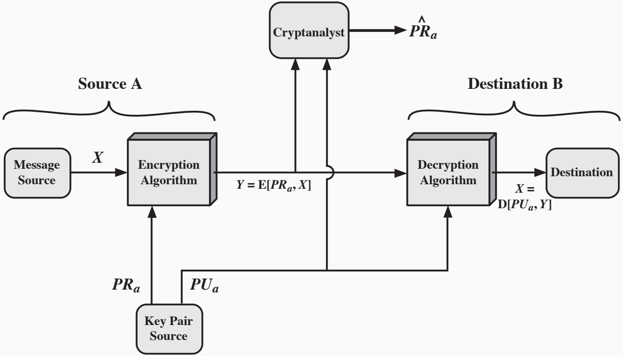
Model of asymmetric encryption – **confidentiality**



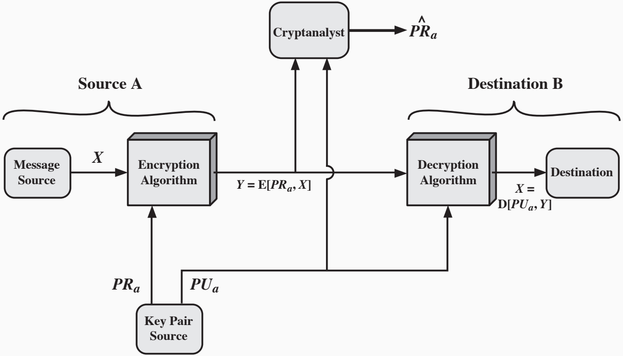
Short representation

$$A \rightarrow B: \quad E_{k_{U_B}}(M)$$

Model of asymmetric encryption – **non-repudiation**



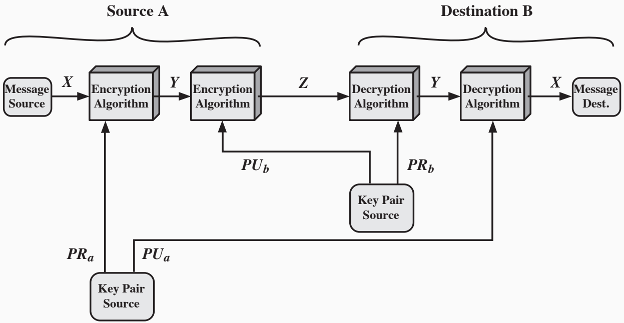
Model of asymmetric encryption – **non-repudiation**



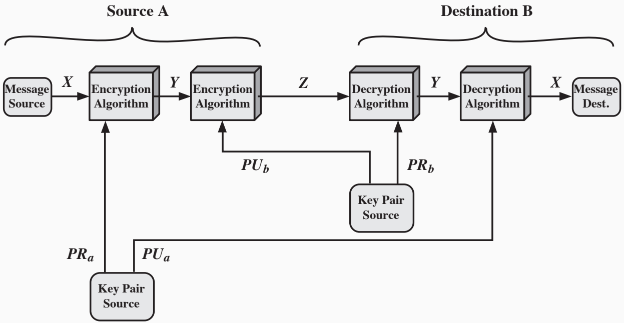
Short representation

$$A \rightarrow B: \quad E_{k_{R_A}}(M)$$

Model of asymmetric encryption – **confidentiality and non-repudiation**



Model of asymmetric encryption – **confidentiality and non-repudiation**



Short representation

$$A \rightarrow B: \quad E_{k_{U_B}}(E_{k_{R_A}}(M))$$

Advantages

- public keys can be shared over insecure channels

Disadvantages

Advantages

- public keys can be shared over insecure channels
- ideal to protect secret keys of symmetric ciphers (key agreement)

Disadvantages

Advantages

- public keys can be shared over insecure channels
- ideal to protect secret keys of symmetric ciphers (key agreement)
- implementation of non-repudiation (digital signatures)

Disadvantages

Advantages

- public keys can be shared over insecure channels
- ideal to protect secret keys of symmetric ciphers (key agreement)
- implementation of non-repudiation (digital signatures)

Disadvantages

- slower and more complex than symmetric ciphers

Advantages

- public keys can be shared over insecure channels
- ideal to protect secret keys of symmetric ciphers (key agreement)
- implementation of non-repudiation (digital signatures)

Disadvantages

- slower and more complex than symmetric ciphers
- infrastructure to certify owners of the public keys is complex

Advantages

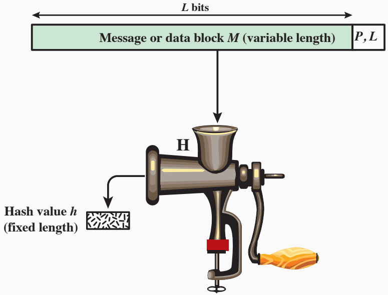
- public keys can be shared over insecure channels
- ideal to protect secret keys of symmetric ciphers (key agreement)
- implementation of non-repudiation (digital signatures)

Disadvantages

- slower and more complex than symmetric ciphers
- infrastructure to certify owners of the public keys is complex
- encryption is too slow, so key encapsulation is used to overcame this limitation

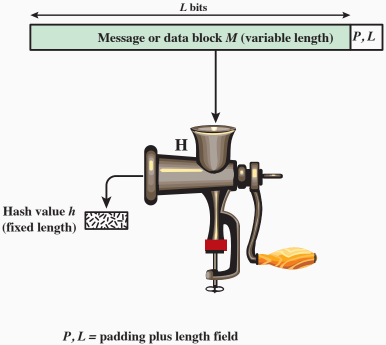
Hash Algorithms | *Algoritmos de Síntese*

Hash algorithms, or hash functions



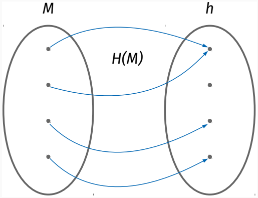
P, *L* = padding plus length field

Hash algorithms, or hash functions

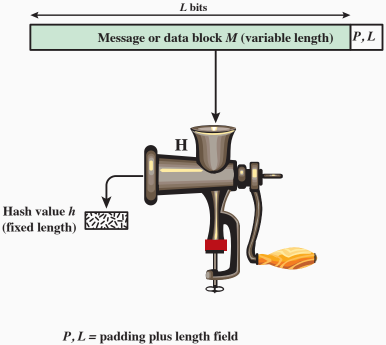


Many-to-one function

- input – a message M of variable length
- output – a value h of fixed length, *e. g.* 256 bits

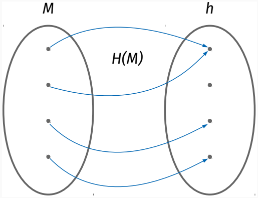


Hash algorithms, or hash functions



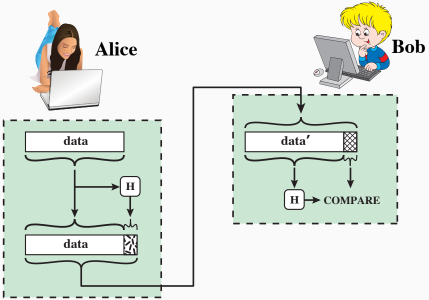
Many-to-one function

- input – a message M of variable length
- output – a value h of fixed length, *e. g.* 256 bits



- size of the messages M universe $= \infty$
- size of the hash values h universe $= 2^{n\ bits}$

Use of hash algorithms

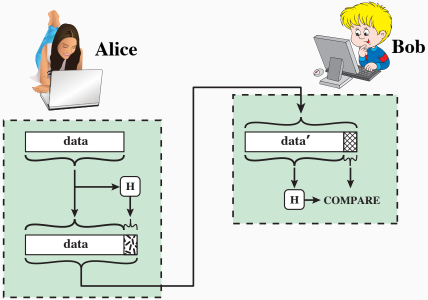


$H(\text{Hello}) = $ `0x8b1a9953c4611296a827abf8c47804d7`

Short representation

$$A \rightarrow B : \quad M \parallel H(M)$$

Use of hash algorithms



$H(\text{Hello}) = $ `0x8b1a9953c4611296a827abf8c47804d7`

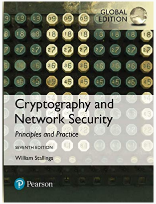
Short representation

$$A \rightarrow B: \quad M \parallel H(M)$$

Prone to attacks

1. the hash algorithm is not enough to guarantee data integrity
2. the hash value must be protected somehow

Questions?



Chapters 3.1, **9.1** and **11.1** of

William Stallings, Cryptography and Network Security: Principles and Practice, Global Edition, 2016