Systems and data security

M. Abderrezak RACHEDI

Full Professor (Professeur des Universités)

University Gustave Eiffel (UGE)
Gaspard Monge Computer Science Lab. (LIGM UMR8040)
Email : abderrezak.rachedi@univ-eiffel.fr

Outline

Part I

- Introduction to security
- Information security
- Cryptology : cryptographie and Cryptanalysis
- Symetric (secret key) cryptography
  - DES and AES protocols
- Asymetric (public key) cryptography
  - RSA and ElGamal protocols
- Multi-key public key cryptography
- Hash Function and Message authentication code
- Electronic signature and blind signature
  - DSA
- Digital certificates and trust model
  - PKI, PGP

Introduction

Application fields

- physical security
- personal safety
- procedural security
  - security audit, IT procedures...
- security of operating systems
- network & communications security
- …

Definition

- *The security of a system corresponds to the non-occurrence of events that could negatively impact the integrity of the system and its environment, during the entire duration of the system's activity.*

- The difference between safety and security
  - **Safety:** Protection of computer systems against *unintentional threats*
  - **Security:** Protection of computer systems against *intentional malicious actions (attacks).*

Reminder - Terminology

- *Vulnerabilities:* weaknesses, security holes that may or may not be exploitable
- *Exploits:* they represent the ways to exploit a vulnerability. There can be several attacks for the same vulnerability
- *Counter-measures*: these are the procedures or techniques used to resolve a vulnerability or to face a specific attack.
- *Threats*: these are determined adversaries capable of create an attack exploiting a vulnerability.

Examples of vulnerabilities

- Operating system vulnerabilities = application level bugs

  - **Buffer overflow**:
    Very common in programs written in C/C++.
  - **Multi-tasking environment**:
    One process dynamically generates other processes
    => complex management / administration
  - **Suspicious handling**:
    Theft of information via a malicious program
  - **Unprocessed input**: SQL Injection, ..
  - …

The main aim of security

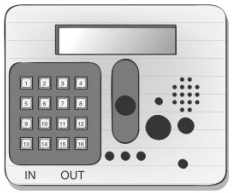
- Prevent unauthorized disclosure of data
- Prevent unauthorized modification of data
- Prevent unauthorized use of network / computer resources in general
- Ensuring the availability of services

Security challenges

- The principle of computer security:
  "Risk 0 does not exist”
- The most reliable computer systems suffer from attacks, especially internal and external attacks.
  "80% of the attacks are internal, such as abuse of privilege"
- The cost of security in terms of money and resources
  Computing capacity, memory, energy, ...
- Security with performance and service availability

Features of a secured system

- **Authentication**
  - The first step in protecting IS resources.
  - Based on the following principle:
    - **Something you have (**a key or a card**),**
    - **Something you know (**password or code**)**
    - and **something you are** (biometrics).

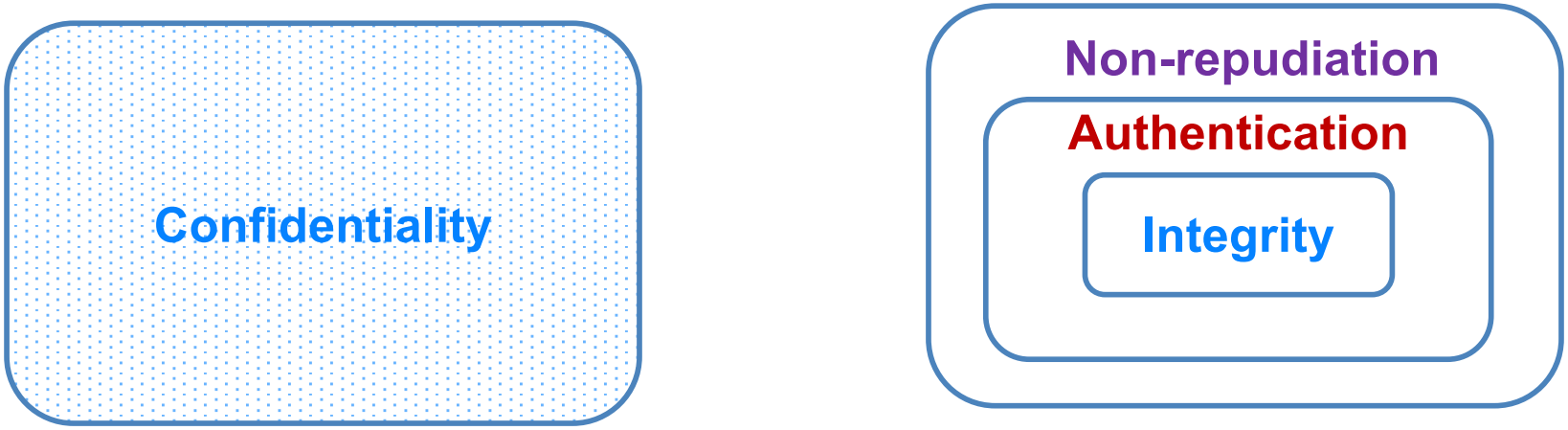


- **Confidentiality**
  - Protection against threats of unauthorized disclosure of information
  - The use of cryptography to implement privacy in a computer system

Features of a secured system

- **Availability**
  - Protection against the threat of disruption of IS (DoS), the implementation of this protection is based on the principle of *fault tolerance*
- **Integrity**
  - Protection against unauthorized modification of data (abuse of privilege)
- **Non-repudiation:**
  - Preventing a person from denying the fact that he or she has performed an operation (example: sending a message, placing an order, ...)

Relationship between security services



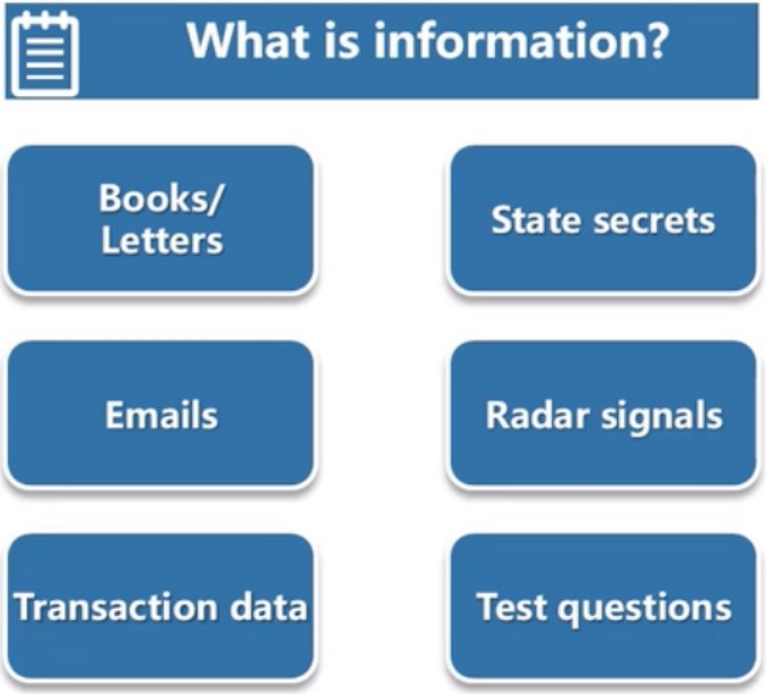
Information Security

Basic concept of information security

- *Information security* is the process of ensuring safe data communication and preventing issues such as information leackage, modification and distruption

- Objectives of this part
  - Describe the definition and characrestics of information security
  - Explain the characteristics and differences of security models
  - Differentiate between security risks

Basic concept of information security



- Information created, received and maintained as evidence
- information by organization or people, in persuance of leagale oligations or transaction of business

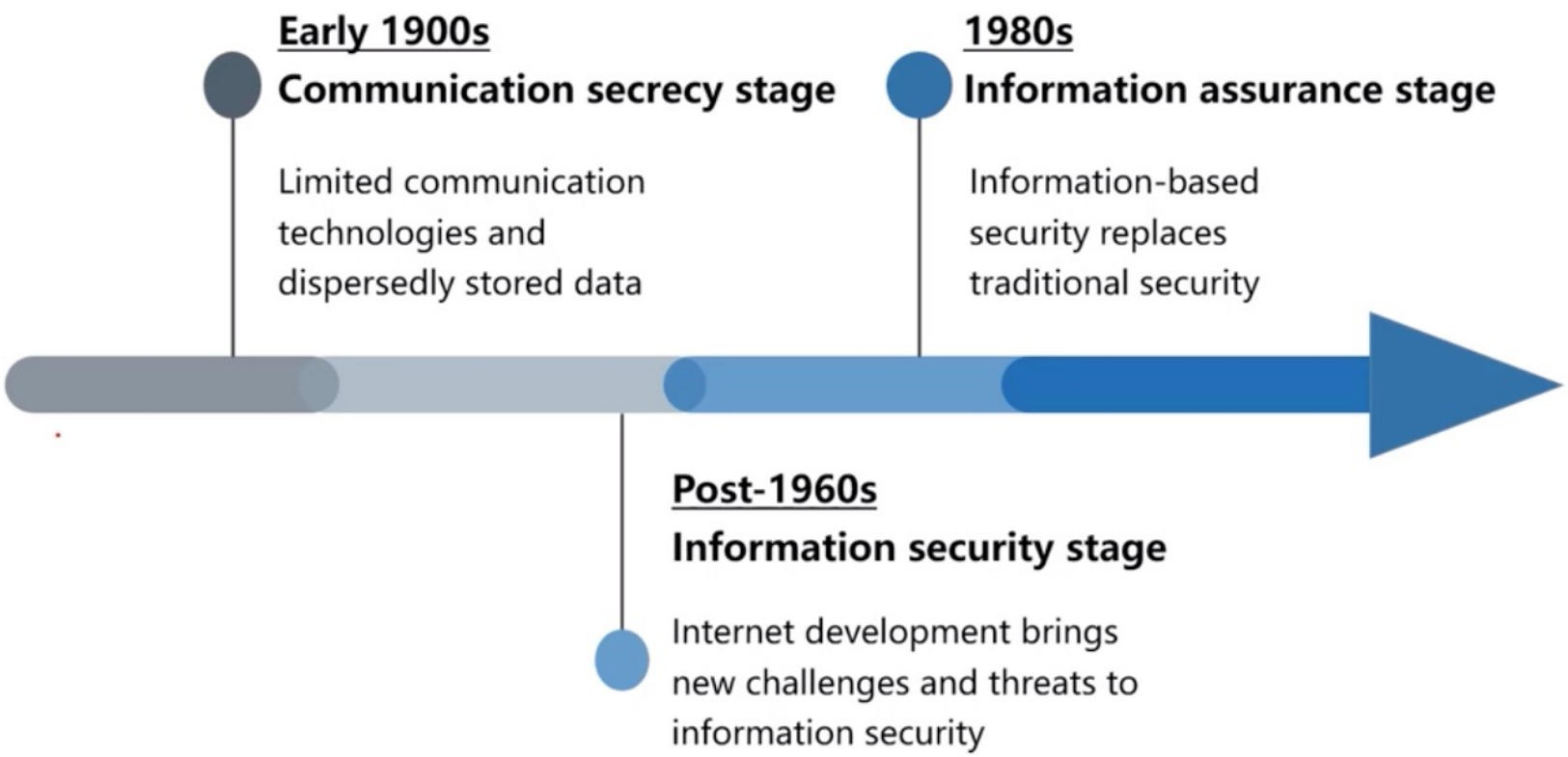
--ISO/IEC Guidelines for the Management of IT Security

Information security

- Information security refers to preservation of Confidentiality, Integrity, and Availability (CIA) of data through security technologies
- These technologies include computer software and hardware, network and key technologies
- Organizanional management measure throughout information lifecycle (generation, transmission, exchange, processing, and storage) are also essential
- The following will be affected if information assets are damaged:

- The aim of information security is to protect data against threats through technical means and effective management

Information Security developpement



Communication secrecy stage

- In the early 1900s, communication technologies were underdevelopped
- Information system security was limited to physical security of information
- As long as information was in a relatively secure place and unauthorized users were prohibited from accessing the information

Information security stage

- Since the 1990s, the Internet technologies have devlopped rapidly, and information leaks have increased
- As results, in addition to classical security services (confidentiality, integrity and avalability), the information security began to focus controllability and non-repudiation

Examples of threats and attacks

- **Case – WannaCry**
  - In 2017, the WannaCry ransomeware cryptoworm, propagated through *EternalBlue*, infected over 100,000 computers and causing a loss of US$ 8billion
  - **EternalBlue** is a cyberattack exploit developed by the NSA. It was leaked by the Shadow Brokers hacker group on April 14, 2017

Examples of threats and attacks

- Case – OceanLotus (APT32)
  - It is a hacker group associated with the government of Vietnam.
  - It is aim is *Cyberespionage* of political dissidents, businesses and foreign officials that have ties to Vietnam.
  - In 2020, it had targeted China's Ministry of Emergency Management and the Wuhan municipal government in order to obtain information about the COVID-19 pandemic.
  - In 2020 Kaspersky researchers disclosed that it had been using the *Google Play Store* to distribute malware.

APT: Advanced Persistent Threat

Cryptology

What is cryptology ?

- Cryptology = **Cryptography + Cryptanalysis**
- Cryptography is the study of mathematical techniques that can be used to provide security services (e.g. Encryption, Authentication, Integrity, etc).
- Cryptanalysis is the study of mathematical techniques that can be used to achieve security objectives.
- Cryptography is formal and theoretical

The goal of cryptography

- Provide a number of security services:
  - Confidentiality: information is accessible only to authorized parties.
  - Authentication:
    - Authentication of entities: the entity is who it claims to be
    - Data authentication: data comes from an authorized party
  - Integrity: any unauthorized data changes are detected
  - Access control: only authorized parties can use specific resources
  - Availability: resources are accessible to authorized parties
  - Non-repudiation (repudiation): the inability (ability) to prohibit communication

Cryptography and its Uses

- Cryptography is only a tool to achieve security objectives
  - others include software, hardware, physical security, etc.
- However, it is very powerful and has many uses
- The need to use cryptographic techniques often arises when parties need to communicate remotely
- Among the well known uses of cryptographic techniques are
  - encryption, digital signatures, integrity control, certificates, etc.

Cryptography and its Uses

- Cryptography also allows :

- To secure e-voting systems and auctions
- To use virtual money (ex. Bitcoin)
- Negotiate and sign contracts (smart contracts)
- Ensure anonymous authentication (via hidden identities and/or hidden policies)
- Modify critical data and computing methods
- To use storage capacity (encrypted data searches) or questionable computing performance (uncheatable grid computing)
- To ensure the protection of private data (including the extraction of private information and the exploitation of data protecting private information)

Attack models

- Security attacks can be passive or active.
  - A passive attacker does not modify the data, but only intercepts / controls the communication.
  - An active attacker can interfere with communications by altering, deleting, inserting data, and even corrupting and controlling certain participants.

  - External attacker: the attacker does not have access to the internal data of the system.
  - Internal attacker: the attacker is part of the system and has access to internal information (e.g., access to cryptographic keys).

- A cryptographic system, in general :
  - Defines the objectives and behavior of an attacker
  - Is formally resilient with such adversarial behavior.

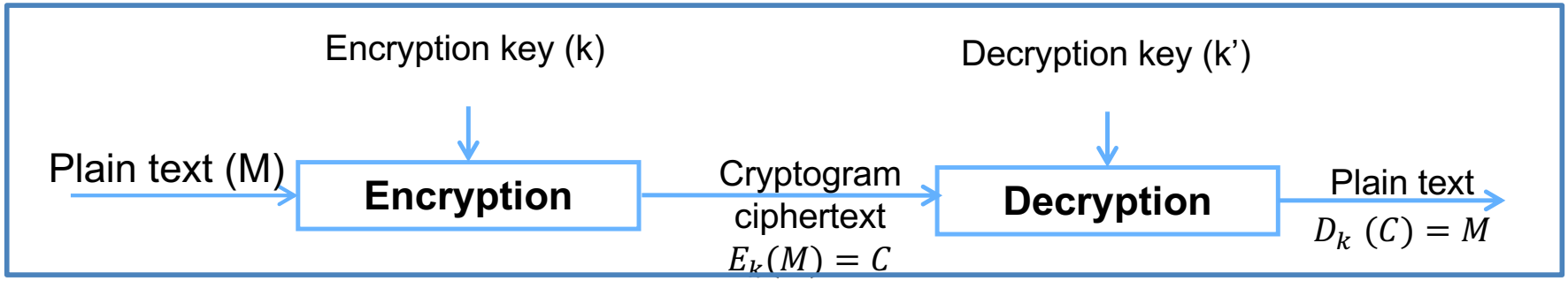
Terminologies and notation

- Suppose the cryptosystem (M, C, K, E, D)
  - M: messages in clear text (without any encryption)
  - C: cryptogram (the result of encryption process)
  - K: set of possible encryption keys
  - E: M x K -> C Encryption process
  - D: C x K -> M Decryption process

- If $k \in K$: Encryption process $\mathrm{E_k(M_1)} = C_1$
  and decryption process $\mathrm{D_k(C_1')} = M'_1$
- $\mathrm{D_k(E_k}(M_1)) = M_1$

Cryptographic algorithms

- A cryptographic algorithm is based on mathematical functions used for encryption and decryption.
- The cryptography is based on a key (k) for encryption and decryption
- The key can take one of a large number of possible values (key space)



Kerckhoff principle

- No secret must reside in the algorithm
  - The secret lies in the key!
- It is necessary to distinguish *Secret* and *Robustness* from the algorithm
- Without K, it is impossible to find M from C
- If we know K, decryption is an easy process

Cryptanalysis

- Cryptanalysis is the science of reconstructing plaintext without knowing the key.
- It can provide either the plain text or the key to the text.
- It can test and highlight the weaknesses of a cryptosystem.

- An attempt at cryptanalysis is called an "attack".

- One of the fundamental axioms of cryptography is that cryptanalysis has all the details of the algorithm and only lacks the key used for encryption.

Cryptanalysis

- There are 4 generic types of cryptanalytic attacks
  - Encrypted text-only (cryptogram) attack
    - The cryptanalysis has the cryptogram of several messages
    - The objective is to find the clear messages or better to find the key(s).
    - Input: $C_1 = E_k(M_1), C_2 = E_k(M_2), \dots, C_i = E_{k(M_i)}$
    - Output: $M_1, M_2, \dots, M_i$ Or key *k, Or an algorithm enables to get* M_{i+1} from $C_{i+1} = E_k(M_{i+1})$
  - The known plain-text attack
    - The cryptanalysis has not only the cryptogram of several messages but also their associated plain-text
    - The purpose is to find the used key(s)
    - Input: $\{M_1, C_1 = E_k(M_1)\}, \{M_2, C_2 = E_k(M_2)\}, \dots., \{M_i, C_i = E_k(M_i)\}$
    - Output: key *k, Or an algorithm enables to get* M_{i+1} from $C_{i+1} = E_k(M_{i+1})$

Cryptanalysis

3. **The plain text attack chosen:**
   - Not only the cryptanalyst has access to both ciphertext and plain text, but he can also choose the plain text to be encrypted.
   - Choosing specific plain text texts that will give more information about the key(s) can help the cryptanalyst

4. **Adaptive plain text attack chosen:**
   - This is a special case of the clear text attack chosen
   - Not only can the cryptanalyst choose the plain text, but he can adapt his choices according to the previous ciphertext.
   - In a chosen plaintext attack, the cryptanalyst is just allowed to choose a large block of plaintext to start with. On the other hand, in an adaptive plaintext attack, he chooses a smaller initial block and then he can choose another block depending on the result of the first one.

Cryptosystems security

- The several algorithms have different levels of security (more or less difficult to break)
- An algorithm is *probably safe* if :
  - the cost to break it exceeds the value of the encrypted information
  - the time needed to break the algorithm is longer than the time needed for the information to remain secret
- Classification of methods to break an algorithm
  - **Complete breakdown:** the encryption- key is found
  - **Global obtention:** a *replacement algorithm* that makes it possible to retrieve the plaintext message without knowing the key
  - **Local obtention:** the plain text of an encrypted message is found
  - **Obtaining information:** get some information about the plain text or the key

Cryptosystems security

- An algorithm is unconditionally safe if :
  - There is not enough information to retrieve the plaintext no matter how much ciphertext is encrypted (e.g. disposable mask)
  - Most other cryptosystems are vulnerable to exhaustive attack (Try all possible keys)
- An algorithm is computationally invulnerable if it cannot be broken with the resources available now and in the future.

- The complexity of an attack can be measured in the following ways:
  - **Information complexity**: the amount of information needed as input
  - **Complexity in time**: the time required to complete the attack
  - **Spatial complexity**: the amount of memory required for the attack
- The complexity of an attack = *min(Information, Time, Space)*

Published vs. secret cryptosystem

Published algorithm

- The only reliable way to assess the security
- Prevents backdoors hidden by developers
- Large number of achievements = low price + high performance
- No need for reverse-engineering protection
- Software implementations
- Local and international standardization

Secret algorithm

- The cryptanalysis must include the recovery of the algorithm
- Smaller number of users = smaller motivation to break
- Robustness and reliability is not guaranteed.
- Not available for other countries

Steganography

- *Steganography is the art of concealment:*
  - *The objective of steganography is to make a message go unnoticed in another message.*
  - *Secret messages can be hidden in images*
  - *Replace the last significant bit of each point of the image with that of a message (without the image changing appreciably)*
  - *This way a 64 kilobyte message can be stored in a 1024x1024 image.*

- *Peter Wayner's mimetic functions allow to divert messages*
- *They modify a message so that its statistical profile looks like something else.*

Summary

- Cryptography is essential to ensure the security of workstations and communications.

- To summarize, the cryptographic concept implies that:

  - You know your objectives
  - You know what your opponent is capable of doing
  - You use known and secure tools as a solution or demonstrate security

Classic Cryptography

Classic Cryptography

- Encryption by substitution
  - Jules Caesar algorithm
  - Vigenère algorithm
- Encryption by transposition
  - ADFGX Encryption
- Enigma machine (with rotors)
- XOR algorithm
- One-time pad (The disposable mask)

Encryption by substitution

- Consists of replacing each character in the plain text with another character in the ciphertext.
  - Associates each letter with a number between 0 and 25, i.e., A = 0, B = 1, etc.
  - The key space K = {1, ..., 25}

- ***Encryption:*** either a k key, change the right letter of k positions, i.e.,

$$E_k(M) = (m + k) \; mod \; 26$$

- ***Decryption:*** either a k key, change the left letter of k positions, i.e.

$$D_k(C) = (C - k) \; mod \; 26$$

Encryption by substitution

- 4 types of substitution :
  - ***Simple substitution***: each character of the plain text is replaced by a corresponding character in the ciphertext.
  - ***Homophonic substitution (multiple representation)***: each character in the plain text is matched with several characters in the ciphertext. (A = 5, 13, 25 or 56)
  - ***Simple substitution by polygrams***: characters are encrypted in blocks. (e.g. ABA = RTQ; ABB=SLL)
  - ***Polyalphabetic substitution***: is a composition of several simple substitution ciphers.

Simple substitution

- The position of the letters in ciphertext is given by
  $$C(letter) = E_k(intialPos) = (initialPos + k)\ modulo\ 26$$
  - The initial position of the letter in alphabet is an incremental sequence. For instance, ***initialPos(A)=0, initialPos(M)= 12***
  - Each character of the plain text is replaced by the one that is *k* places further on in the *modulo 26* alphabet.
  - Based on a circular rotation of the alphabetical sequence
  - In the case of Jules Cesar algorithm ***k=3 (named ROT3)***

0	1	2	3	4	5	6	7	8	9	10	11	12
3	4	5	6	7	8	9	10	11	12	13	14	15
13	14	15	16	17	18	19	20	21	22	23	24	25
16	17	18	19	20	21	22	23	24	25	0	1	2

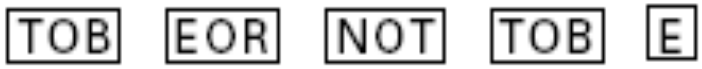
A	B	C	D	E	F	G	H	I	J	K	L	M
D	E	F	G	H	I	J	K	L	M	N	O	P
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Q	R	S	T	U	V	W	X	Y	Z	A	B	C

  - ROT13 is a program used on Unix systems : Ex. ***C= ROT13(M)***
- Advantage: It is a simple substitution encryption
- Disadvantage: Unsecure encryption system because the key space contains only 26 elements.
- Improvements : we substitute each letter by any other letter (26! = 4,1026 possible combinations)

Vigenère Encryption

- The message is divided into blocks of length d∈ [0.25]
- Each block is transformed according to Caesar's algorithm. So, we need a key vector of size "d": K=(k1, k2, ..., kd)
  - Ex. d = 3, (k1, k2, k3) = (7, 0, 12)

«**TO BE OR NOT TO BE**»
Cyphertext: «**AO NL OD UOF AO NL**»



- The cryptanalyst must find:
  - The size of the key and then the key itself
  - The method is based on the frequency of letters

Encryption by transposition / Transposition ciphering

- The characters of the plain text remain unchanged but their respective positions are modified.
- To obtain the cryptogram, we write the text in clear text horizontally squared of fixed width (the key) and we read the ciphertext vertically.
- Example (with a key size 9)
  Plain text : « L'ASSASSIN EST LE DOCTEUR MATRIX REGARDEZ DERRIERE L'HORLOGE »

L	A	S	S	A	S	S	I	N
E	S	T	L	E	D	O	C	T
E	U	R	M	A	T	R	I	X
R	E	G	A	R	D	E	Z	D
E	R	R	I	E	R	E	L	H
O	R	L	O	G	E			

- Ciphertext (cryptogram):

« LEERE OASUE RRSTR GRLSL MAIOA EAREG SDTDR ESORE EICIZ LNTXD H »

Transposition ciphering

- Example: we have the key: $B_1I_4D_2U_6L_5E_3$

  - Plain text:
    ACHETER_DEUX_MILLE_ACTIONS_DE_LA_SOCIETE_PEUGEOT
  - The key size = 6 and size of the text to be encrypted = 48 so the size of the matrix = 6*8
  - Encrypted text:
    AR__NLIEEELTDSEEC_MASAEUEXEO_CPTTULIEO_OHDIC__TG

- **Disadvantages:**
  - As the letters of the cryptogram are the same as those of the plain text, a statistical analysis of the frequency of the letters gives an important clue to the cryptanalyst.

Encryption by substitution & transposition

Statistical attack techniques

- Statistical analysis of encrypted texts
- Determination of symbol appearance frequencies
- Comparison with typical language frequencies

- The problem is to have :
  - computing power
  - enough text for the length of the keys used

Affine encryption

- The encryption function is only the affine function of type

$$\mathbf{y} = (\boldsymbol{ax} + \boldsymbol{b})\ \boldsymbol{mod}\ \mathbf{26}$$

  - Where *a* and *b* are constants, and where *x* and *y* are numbers corresponding to the letters of the alphabet (A=0,B=1,...)
  - If a=1, then we find the Caesar encryption and b is the offset.

- How it works
  - Key = $(\mathrm{k}_1 = a, k_2 = b)\ \mathrm{k}_1, \mathrm{K}_2 \in [0, 25]; \gcd(k_1, 26) = 1$
  - Ciphering (encryption porcess):

$$\boldsymbol{C_i} = \boldsymbol{f}(\boldsymbol{M_i}) = (\boldsymbol{k_1}.\boldsymbol{M_1} + \boldsymbol{k_2}\ )\ \boldsymbol{mod}\ \mathbf{26}$$

  - Deciphering (decryption process):

$$\mathbf{M_i} = \mathbf{f}^{-1}(\mathbf{C_i}) = \frac{\boldsymbol{C_i} - \boldsymbol{k_2}}{\boldsymbol{k_1}}\ \boldsymbol{mod}\ \mathbf{26}$$

Affine encryption

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

q	r	s	t	u	v	w	x	y	z
16	17	18	19	20	21	22	23	24	25

- **Example -1-**
  - Key = (9,2), gcd(9, 26)=1
  - y= 9.x+2 mod 26
    Plain text: M=« **affine** » = « 0 5 5 8 13 »
  - Give the cryptogram of this message
  - Decipher this cryptogram with the previous method

Correction -1-

- Encryption
  y(0)=2 => C ; y(5)=21 => V ; y(8)=22 =>W ; y(13)=15 => P ; y(4)=12 => M
  C= f(M)=**CVVWPM**
- Decryption

$$\mathrm{M} = \mathbf{f}^{-1}(C) = \frac{C-2}{9} \, mod\ 26 = \frac{1}{9}(C-2) mod\ 26$$

$$27 = 9\,.\,3 = 1\ mod\ 26 \Rightarrow 3 = \frac{1}{9}\ mod\ 26$$

$$\Rightarrow\ M = 3(C-2) mod\ 26$$

C= CVVWPM = «2 21 21 22 15 12 »

$$\mathbf{f}^{-1}(2) = 0\ mod\ 26 = 0 \Rightarrow A\ ; f^{-1}(21) = 3(21-2) mod 26 = 5 \Rightarrow F$$

Affine encryption

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

q	r	s	t	u	v	w	x	y	z
16	17	18	19	20	21	22	23	24	25

- Example -2-
  - Key : (13, 4) ; $\mathbf{y(x) = 13\,x + 4}$
  - Encrypt both messages: M1="input"; M2="alter".
  - What do you notice?
  - Is it possible to decipher these messages? Why
  - How many keys are possible with this encryption method?

Correction -2-

- Encryption of message M1 = « input » = « 8 13 15 20 19 »
  y(8)= 4 => E ; y(13)=17 => E ; y(15)=17 => E ; y(20)=4 => E ; y(19)=17 => R
  F(M1)= ERRER
- Encryption of message M2 = « alter » = « 0 11 19 4 17 »
  y(0)=4 => E ; y(11)=17 => R; y(19)=17 => R ; y(4)=4 => E ; y(17)=17 => R
  F(M2)= ERRER
- The two cryptograms are similar

It is not possible to decrypt both messages.
Because the condition $\boldsymbol{gcd(13, 26)=13\neq 1}$

- Keys must satisfy this condition $\boldsymbol{gcd(\alpha, 26)=1}$ knowing that $\alpha\in[0, 25]$
  we have 12 possibilities for α is the number of possible keys is: 12 x 26 =312 possibilities

Cryptanalysis of Affine encryption

- In case the cryptanalyst has :
  - Only the cryptogram:
    - It is necessary to look for the key in all the 312 possibilities
    - Use the technique of the *frequency of appearance* of letters in a given language
  - To know certain information from the plain text
    - Knowing two letters in the plain text and their equivalents in the cryptogram is enough to find the key
    - Example: Suppose we have managed to get the first two letters of the text in plain text "if" and the equivalent in the cryptogram is "PQ
      Reminder: y(x)= K1.x+K2 mod 26 ; the goal is to find K1 and K2

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

q	r	s	t	u	v	w	x	y	z
16	17	18	19	20	21	22	23	24	25

x= 8 (i) et C=y(8)= 15 (P)
x= 5 (f) et C=y(5)= 16 (Q)

$$\begin{cases} 8\,K_1 + K_2 \equiv 15\ (mod\ 26) \ \ldots\ldots (1) \\ 5\,K_1 + K_2 \equiv 16\ (mod\ 26) \ \ldots\ldots (2) \end{cases}$$

$(1) - (2) \Rightarrow 3\,K_1 \equiv -1\ mod\ 26 \Rightarrow \mathbf{3\,K_1 \equiv 25\ mod\ 26}\ \ , \mathbf{gcd(3,26) = 1}$

$\Rightarrow \mathbf{3\,K_1 - 25 = \alpha.26}\ \ \mathbf{avec}\ \boldsymbol{\alpha} \in \boldsymbol{Z}\ \Rightarrow \mathbf{K_1 = 17}$

From $(1) : 8.17 + K_2 \equiv 15\ (mod\ 26) \Rightarrow\ \mathbf{K_2 = 9}$

Cryptanalysis of Affine encryption

- The possibility to choose two letters in plain text
  - The most interesting choice is the letters: "ab" = "0 1".
  - y(0)=K_1. 0+K_(2)=K_2 and the second equation y(1)=K_1+K_2
  - Then the key is found
- The possibility to choose the letters in the cryptogram
  - AB" may be selected for the same reasons.

Playfair encryption

- Playfair was invented in 1854

- How it works
  - The key is a word whose repeated letters are deleted.
  - Example: the key is "playfair" it becomes: "playfir".
  - The remaining letters are used to form the 5x5 matrix:
  - The two letters "i" and "j" are treated as a single letter.

p	l	a	y	f
i	r	b	c	d
e	g	h	k	m
n	o	q	s	t
u	v	w	x	z

  - Assuming that the plain text to be encrypted is:
    « ***meet at the schoolhouse*** »
  - Delete spaces and group by two letters
  - If a letter appears twice in succession, the letter "x" is inserted between the two letters.
  - The text becomes: " **me et at th es ch ox ol ho us ex** ".

Playfair encryption

p	l	a	y	f
i	r	b	c	d
e	g	h	k	m
n	o	q	s	t
u	v	w	x	z

- We use the matrix to encrypt the text with the application of the following rules:

- ***Rule 1:*** If two letters are not in the same row or column, replace each letter by the letter that is in its row and that is in the column of the second letter (ex. ET => MN).

- ***Rule 2 :*** If both letters are on the same line, replace each letter with the letter on its right (if the letter is on the last column, go back to the first column) - (ex. me => EG)

- **Rule 3**: If the two letters are on the same column, replace each letter by the letter on the line below it (if the letter is on the last line go back to the first line) - (ex. ol => VR)

Give the cryptogram of this text: "me and at th es ch ox ol ho us ex".

The cryptogram is:
EG MN FQ QM KN BK SV VR GQ XN KU

ADFGX encryption

- It is a German process developed during the 1st World War.
- It uses a combination of both transposition and substitution ciphers.
- The choice of the letters ADFGVX is linked to Morse code symbols which are difficult to confuse (.-, -.., , ..-., ..-., --., -., -..-) to avoid transmission errors.

How it works

- The letters "i" and "j" are considred as a single letter.
- Put the letters of the alphabet in a 5x5 matrix
- The labels of the column and the row of the matrix are labeled with : A, D, F, G, X
- Each letter in the plain text is replaced by the label of its line and column.
- Example of the 5x5 matrix, (the letter s => FA; the letter z => DG)
  ESIPE => AG FA GG AA AG
- Another step based on a keyword that increases complexity
- This keyword is used to perform a substitution encryption.

	A	D	F	G	X
A	p	g	c	e	n
D	b	q	o	z	r
F	s	l	a	f	t
G	m	d	v	i	w
X	k	u	y	x	h

ADFGX encryption

- Example
  - Use ADFGX encryption to encrypt the following message: "**Kaiser Wilhelm**" with the keyword: "Rhein" and this 5x5 matrix

	A	D	F	G	X
A	p	g	c	e	n
D	b	q	o	z	r
F	s	l	a	f	t
G	m	d	v	i	w
X	k	u	y	x	h

Solution :

❑ Step 1: Le 1er cryptogram is:

C1= « XA FF GG FA AG DX GX GG FD XX AG FD GA »

R	H	E	I	N
X	A	F	F	G
G	F	A	A	G
D	X	G	X	G
G	F	D	X	X
A	G	F	D	G
A				

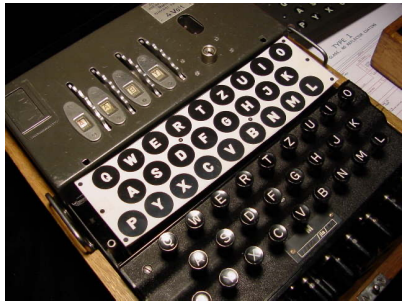
❑ Step 2:

- ❑ Put C1 in the matrix with the keyword like lebel
- ❑ Order the columns of the matrix according to the order of the alphabet
- ❑ he last cryptogram is the sequence of the columns of the ordered matrix:
  C2 = « FAGDFAFXFGFAXXDGGGXGXGDGAA »

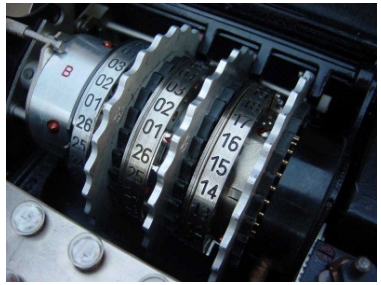
E	H	I	N	R
F	A	F	G	X
A	F	A	G	G
G	X	X	G	D
D	F	X	X	G
F	G	D	G	A
				A

Encryption automation: Enigma machine

- These are mechanical devices whose purpose is to automate encryption.
- A enigma (or rotor) machine has a keyboard and a set of rotors.
- Each rotor is an arbitrary permutation of the alphabet and performs a simple permutation
- Combination of all rotors makes the machine safer
- The best known rotor machine is the Enigma machine used by Germans during the second world war.



Keyboard



Rotors

Enigma Machine

Ou exclusif simple

- **XOR (exclusive OR)** is an operation (noted $\oplus$)
- It's has the following properties:
  - a $\oplus\, a = 0$
  - a $\oplus\, b \oplus\, b = a$

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

- If an encryption algorithm is faster than DES then there is a good chance that it is the XOR algorithm.
- It's a secret key algorithm
- The cryptogram is only the result of the "XOR" operation between the plaintext and the key.
  - M $\oplus\, K = C$
  - C $\oplus\, K = M \oplus\, K \oplus\, K = M$
- Disadvantages:
  - No real security, the algorithm can be broken in a few minutes.

The one-time pad mask

- Invented by AT&T in 1917
- The one-time mask is a long non-repetitive and random sequence of letters.
- The encryption algorithm adds the rank of the letter to be encrypted to the corresponding letter rank of the mask, the result modulo 26 gives the rand of the encrypted letter.
- Each letter of the mask is used only once, for a single message.
- Example :
  - Plain text: **MASQUEJETABLE**
  - Mask: **TBFRGFARFMIKL**
  - Encrypted text: **GCYIBKKWZNKWQ**
  -

M + T mod 26 = G
A + B mod 26 = C
S + F mod 26 = Y
Etc.

The one-time pad mask

- The one-time mask can be extended to binary data encryption
  - The mask is composed of bits instead of characters
  - The addition operation is replaced by XOR
- The use of the one-time mask is mainly for ultra-secret communication channels and at low flow rates
  - E.g. the red phone between the USA and the Soviet Union
- Advantages:
  - The combination of a clear text with a completely random key gives a random ciphered text.
  - Secure encryption regardless of computing power
- Disadvantages:
  - The length of the mask is equal to the length of the message to be encrypted.
  - The problem of mask distribution and its saving
  - Strong synchronization between transmitter and receiver (a one bit shift, the decrypted message will not make sense)

Modern Cryptography

Cryptographie moderne

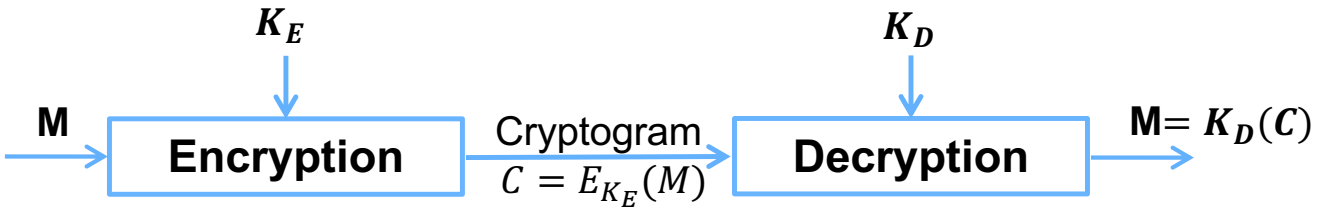
- Modern cryptography is the combination of:
  - theoretical computer science (performance evaluation of algorithms, complexity theory),
  - Algorithms (delicate programming of certain processes involving complex calculations),
  - electronics (realization of circuits implementing particular algorithms)
  - mathematics (algebra and number theory)

- There are two main encryption families
  - Symmetric or secret-key encryption
  - Asymmetric or public-key encryption

Modern cryptography

- Symmetric or secret key encryption
  - DES: Data Encryption Standard
  - AES: Advanced Encryption Standard
- Asymmetric or public-key encryption
  - RSA (Rivest Shamir Adleman)£
- One-way hash function
  - SHA-1
  - Message Authentication Code (MAC)
- Digital Signature
  - DSA (Digital Signature Algorithm)
  - Blind signature
- Digital Certificate

Symmetric or secret key encryption



- Characteristics:
  - The encryption and the decryption keys are the same $K_E = K_D = K$
  - Standard algorithms: DES, AES, ...
  - Key generation: Key randomly chosen in the key space.
  - Principle: Algorithms based on transposition and bit substitution operations of the clear text, depending on the key.
  - Size of the keys: (standard) 64 or 128 bits
  - Performance: Very fast encryption
  - Key distribution:
    - Critical operation
    - Must be done in a secure way (even manually).

Symmetric or secret key encryption

- The types of symmetric algorithms:
  - **Streaming cipher algorithms (stream cipher)**
    - Act on one bit at a time
    - The most common : RC4 (key length: 128 bits)
  - **Block cipher algorithms (block cipher)**
    - Act on plain text per block (generally 64bits)
    - DES (Data Encryption Standard): 56-bit 64-bit encryption key
    - Triple DES: use EDE mode (Encryption, Decryption, Encryption) with three separate keys (168bits) or only two (112bits)
    - IDEA, CAST (128bits), Blowfish (up to 448bits)
    - AES (Advanced Encryption Standard): Rijndael (128, 192, 256 bits)

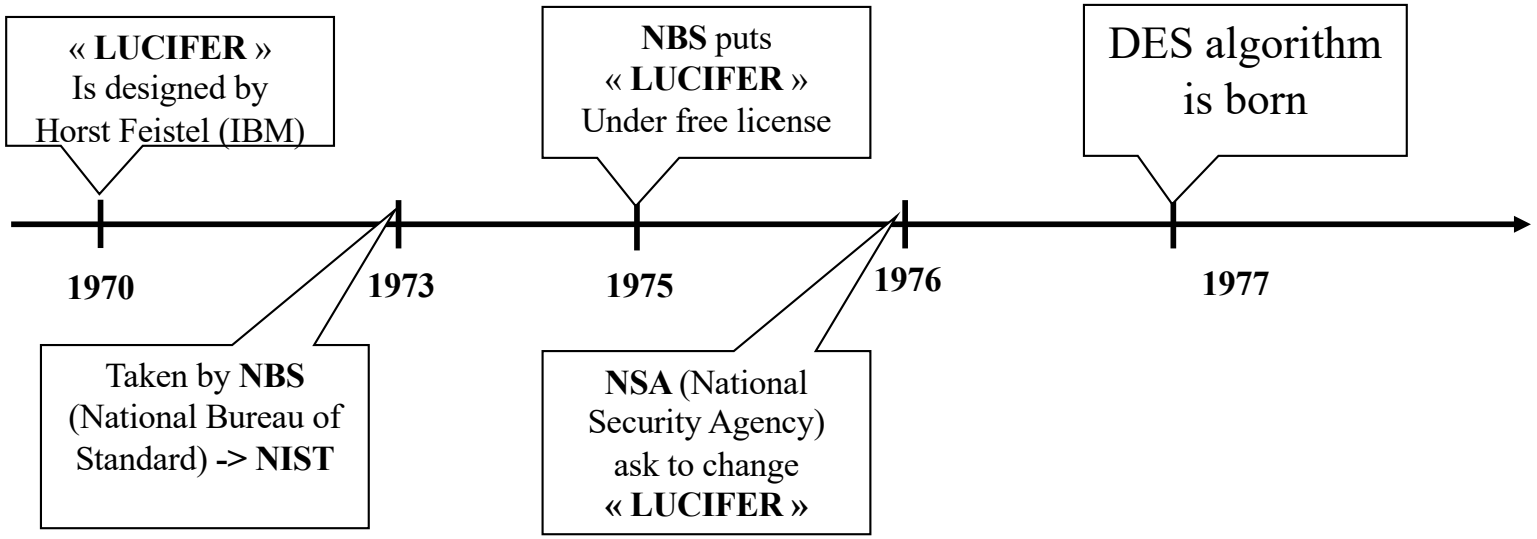
Symmetric or secret key encryption

- Advantages:
  - Simple and fast compared to asymmetric algorithms

- This kind of encryption has the following problems/limitations:
  - Keys have to be distributed/exchanged secretly
  - *Heavy key management*: If a different key is used for each pair of users, the total number of keys increases very quickly compared to the number of usersex: n users => the number of keys is $\boldsymbol{x = \frac{n(n-2)}{2}}$
  - If the key is compromised, the attacker can pretend to be one of the participants.

DES: Data Encryption Standard

- History

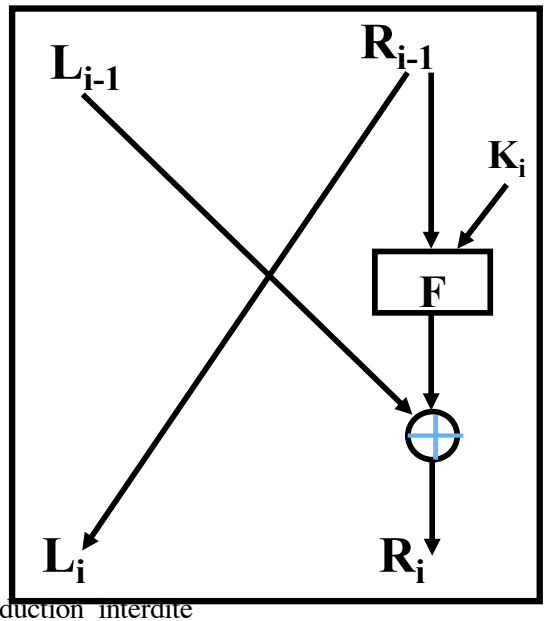


- Features:
  - DES easy to implement, software and hardware, and it is very fast
  - The DES key is a 64-bit string, but in fact only 56 bits are actually used to define the key.
  - There are 2^{56} possible keys, or about 72 million billion possibilities.

Simplified DES algorithm

- The iterative algorithm that provides block encryption
- The blocks are encrypted separately

- Assuming that we have a 12-bit M message with a single block
- **Encryption process**
  - *M* can be written as $\boldsymbol{L_0\ R_0}$ where L_0 is 6 first bits ($M = L_0\ R_0$)
  - The size of the K-key is 9 bits
  - The round (i) of the algorithm transforms the input $\boldsymbol{L_{i-1}}$ $\boldsymbol{R_{i-1}}$ with the use of the key K_i obtained from the key *K*
  - The main part is the function $\boldsymbol{f(R_{i-1}, K_i)}$ whose result is on 6 bits
  - $\boldsymbol{L_i = R_{i-1}}$ et $\boldsymbol{R_i = L_{i-1} \oplus f(R_{i-1}, K_i)}$
  - This operation is performed n times to get the cryptogram $\boldsymbol{C = [L_n][R_n]}$



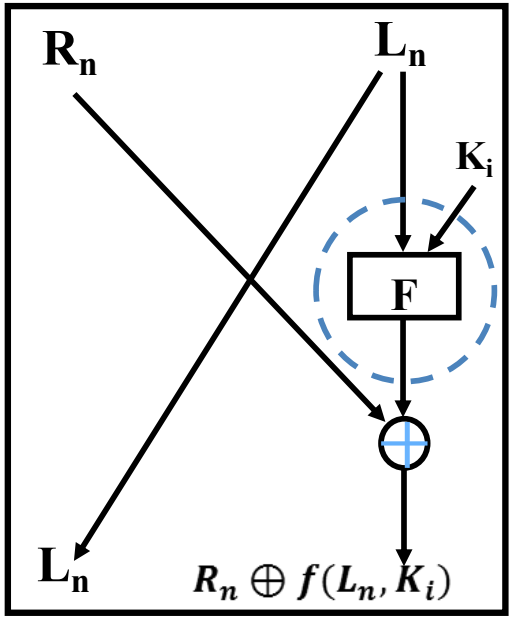
Simplified DES algorithm

Decryption process: $C = L_n R_n$

- It is a reverse encryption operation
  - Reverse use of keys ($K_n, \ldots, K_i, \ldots, K_1$)
  - Left to right block shift (switch): $[L_n][R_n] \Rightarrow [R_n][L_n]$
  - 1er iteration : $[R_n][L_n] \Rightarrow [L_n][R_n \oplus f(L_n, K_n)]$
  - According to the encryption operation $L_n = R_{n-1}$ and $R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$
  - $[L_n][R_n \oplus f(L_n, K_n)] = [R_{n-1}][L_{n-1} \oplus f(R_{n-1}, K_n) \oplus f(L_n, K_n)]$

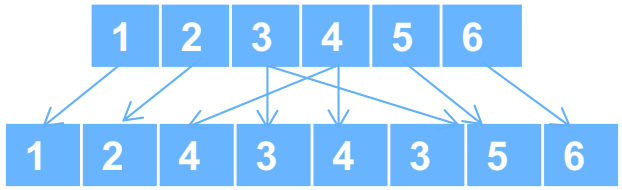
$$with\ L_n = R_{n-1}$$
$$f(R_{n-1}, K_n) \oplus f(R_{n-1}, K_n) = 0$$
$$(A \oplus A = 0)$$

  - $[L_n][R_n \oplus f(L_n, K_n)] = [R_{n-1}][L_{n-1}]$
  - ……. (n iterations)
  - The shift (switch) of the block from right to left
  - $[R_0][L_0] = plain\ text$

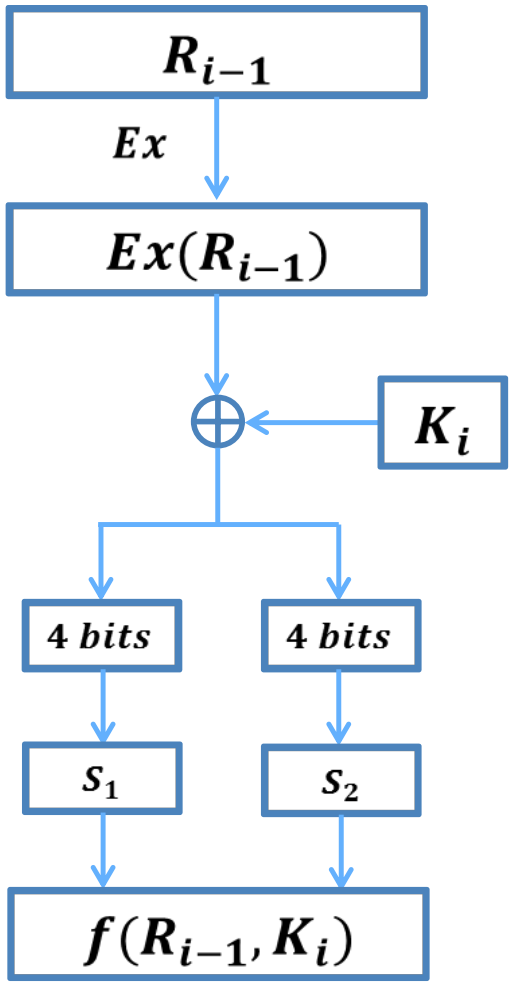


Simplified DES algorithm

- This is the ***"Expander"*** function
  - With 6 bits at the input and 8 bits at the output



  - Example : $Ex(011001) = 01010101$



- ***S-Box*** is a substitution function that contributes to "confusion" by making the original information unintelligible.
  - S1 and S2 are S-Boxes: with 4 bits in and 3 bits out.
    - The first bit indicates the number of the line (0: for the first line and 1: for the second line)
    - The 3 remaining bits represent the column number (000: the first column, 001: the second, ...)
    - Example : S1-Box(1010) = 110 (Second row and third column)

$$S_1 \quad \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 \quad \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}$$

erdite

Simplified DES algorithm

- The calculation of the key of ith iterations K_i (8 bits) from K (9 bits)
  - K_i is of 8 bits obtained starting with the ith bit of the key K
  - Example : $\boldsymbol{K = 010011001 \Rightarrow K_4 = 1100101}$

- Example :
  - Compute $\boldsymbol{f(R_{i-1}, K_i)}$ knowing that $\boldsymbol{R_{i-1} = 100110}$ et $\boldsymbol{K_i = 01100101.}$
  - $\boldsymbol{f(R_{i-1}, K_i) = Ex(R_{i-1}) \oplus K_i}$, $then\ the\ result$:
    The first 4 bits are sent to S1-box and the last 4 bits are sent to S2-box.
  - $\boldsymbol{Ex(100110)} \oplus K_i = 10101010 \oplus 01100101 = 1100\ 1111$
  - $S_1(1100) = 000$ et $S_2(1111) = 100$
  - $f(R_{i-1}, K_i) = 000100$

S_1

101	010	001	110	011	100	111	000
001	100	110	010	000	111	101	011

S_2

100	000	110	101	111	001	011	010
101	011	000	111	110	010	001	100

- Exercise:
  - Give the result of the next iteration L_iR_i knowing that the result of the previous iteration is t $L_{i-1}R_{i-1} = 011100100110$
  - $L_iR_i = 100110011000$

Differential Cryptanalysis

- Differential cryptanalysis was introduced by Biham and Shamir in 1990
- The idea is to compare the difference in the cryptograms to deduce the information about the key.
- It is based on the XOR operation to find the difference between two sets of bits
- In the simplified DES algorithm, the XOR between the key and $\boldsymbol{E}(\boldsymbol{R}_{i-1})$ is performed

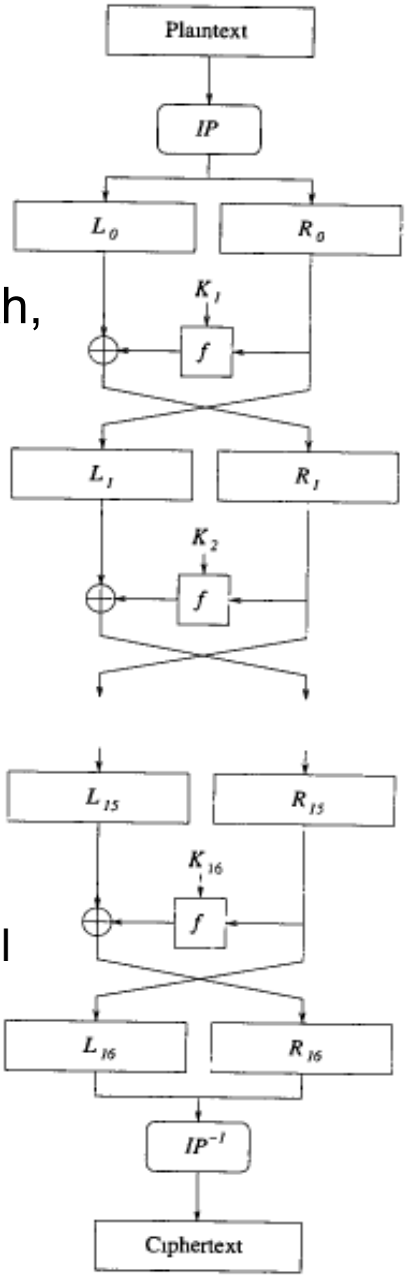
- **DES attacked!**
  - Attack of the RSA laboratories against DES (the key is found in 22 hours).
  - With the help of a dedicated machine named: Deep Cracker (250000 Dollars)
  - 100,000 PCs using distributed computing
- Changing the key size: **128 bits minimum**

DES: Data Encryption Standard

- This is 64-bit block encryption.
- The key is 56 bits but expressed by 64bits (because the 8th, 16th, 24th bits, ... are parity bits used to detect errors)

Encryption process

- Plain text is a 64-bit message ($\boldsymbol{m}$)
  1) $\boldsymbol{m}$ is initially swapped to $\boldsymbol{m_0 = IP(m) = L_0 R_0}$
     $\boldsymbol{L_0}$ and $\boldsymbol{R_0}$ $are\ blocks\ of\ 32\ bits$
  2) for $\boldsymbol{1 \leq i \leq 16}$ do :
     $$\mathbf{L_i = R_{i-1}}\ ;\ \boldsymbol{R_i = L_{i-1} \oplus f(R_{i-1}, K_i)};$$
     with $K_i\ is\ string\ of\ 48bits\ obtained\ from\ K$
  3) Invert the blocks to get $\boldsymbol{R_{16}L_{16}}$, then apply the inverse of the initial permutation $\mathbf{c = IP^{-1}(R_{16}L_{16})}$



DES: Data Encryption Standard

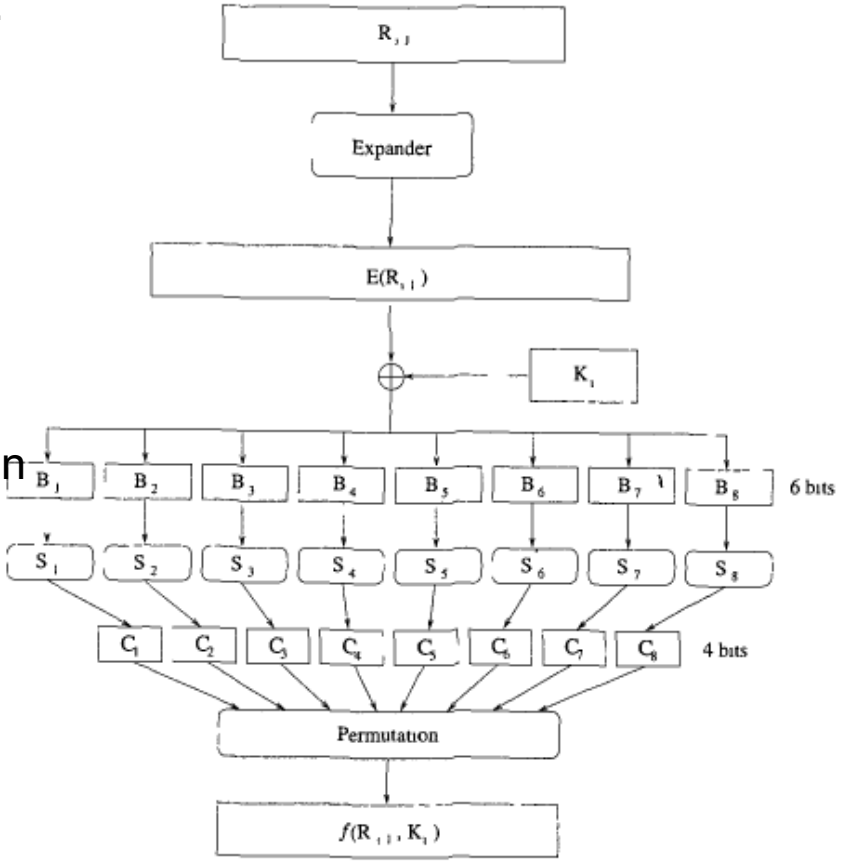
Initial Permutation

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

- Initial Permutation (IP)
  - 58th bit of the message $\boldsymbol{m}$ becomes the 1st bit of $\boldsymbol{m\_0}$, the 50th bit becomes the 2nd bit.
- Function $f(R_{i-1}, K_i)$
  - R_{i-1} of 32-bit size is extended to 48 bits by $E(R_{i-1})$

Expansion Permutation

32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1



  - Compute $E(R_{i-1}) \oplus K_i$, the 48-bit result is written in the following format $B_1 B_2 \ldots . B_8$ with B is a block of 6 bits
  - There are 8 S-Boxes $\{S_1, S_2, \ldots, S_8\}$, each block B_j is sent in a box S_j : $S_j(B_j) = C_j$ block of 4 bits
  - $B_j = b_1 b_2 b_3 \ldots b_6$, with $b_1 b_6$ gives the line number and $b_2 b_3 b_4 b_5$ gives the column number
    - Example : $B_3 = 001001$, 01 first line and 0100 the 5th column.

DES: Data Encryption Standard

- The $\{C_1 C_2 \ldots C_8\}$ are swapped according to the table below and the result is $f(R_{i-1}, K_j)$ (a 32-bit string)

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

- Obtaining the keys of each iteration $\{K_1, K_2, \ldots, K_{16}\}$ from the key K
  - Delete the parity bits (8th , 16th ,... 24th bits)
  - The rest of the bits are permuted according to the following table

Key Permutation

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

  - The permutation result is written in the format $\boldsymbol{C_0 D_0}$ withc $\boldsymbol{C_0}$ et $\boldsymbol{D_0}$ are 28bit blocks
  - For $\boldsymbol{1 \leq j \leq 16}$ do $C_i = LS_i(C_{i-1})$ and $D_i = LS_i(D_{i-1})$ with LS_i is a 1 or 2 bit shift function depending on the iteration number

Number of Key Bits Shifted per Round

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Shift	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

  - The 48 bits of the key $\boldsymbol{K_i}$ are chosen from $\boldsymbol{C_i D_i}$ according to the following table:

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

DES: Data Encryption Standard

- The S-Box is an important element of DES algorithm

- Each S-Box has 6 input and 4 output bits
- The outputs of the S-Box must not be linear in relation to the inputs
- Each line of the S-Box contains all the numbers from 0 to 15.
- If two inputs of the S-Box are different by only one bit the result is different by at least 2 bits.
- ….

S-Boxes

S-box 1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S-box 2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S-box 3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S-box 4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S-box 5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S-box 6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S-box 7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S-box 8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

DES: Data Encryption Standard

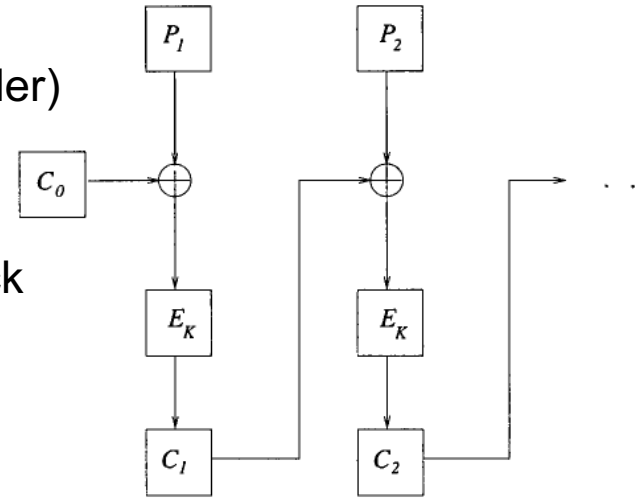
- DES is a block cipher algorithm
- Block encryption can be used with several modes of operation
  - Electronic Codebook (ECB)
  - Cipher block Chaining (CBC)
  - Cipher FeedBack (CFB)
  - Output FeedBack (OFB)
  - CounTeR (CTR)

- **Electronic Codebook (ECB)**
  - Consists of dividing the message to be encrypted into several blocks (of the same size)

    $P = [P_1, P_2, \dots, P_L]$ $C = [C_1, C_2, \dots, C_L]$ ou $C_j = E_k(P_j)$
  - Vulnerable to attacks (we can guess the message header)

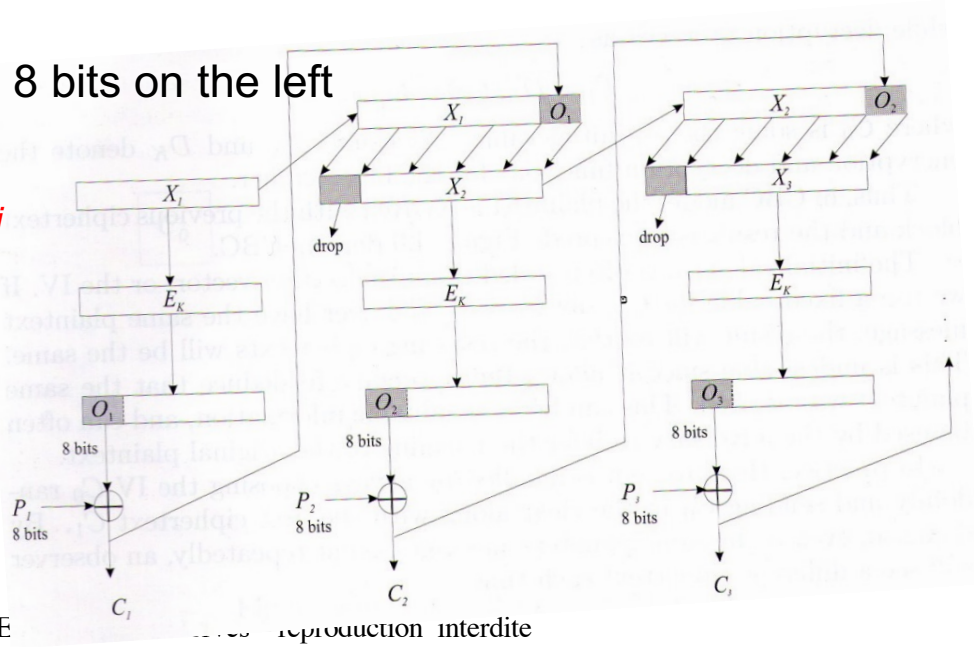
- **Cipher block Chaining (CBC)**
  - The encryption of a block depends on the previous block

    $C_j = E_K(P_j \oplus C_{j-1})$ $P_j = D_K(C_j) \oplus C_{j-1}$
  - C_0 is the random initialization vector (IV)



DES: Data Encryption Standard

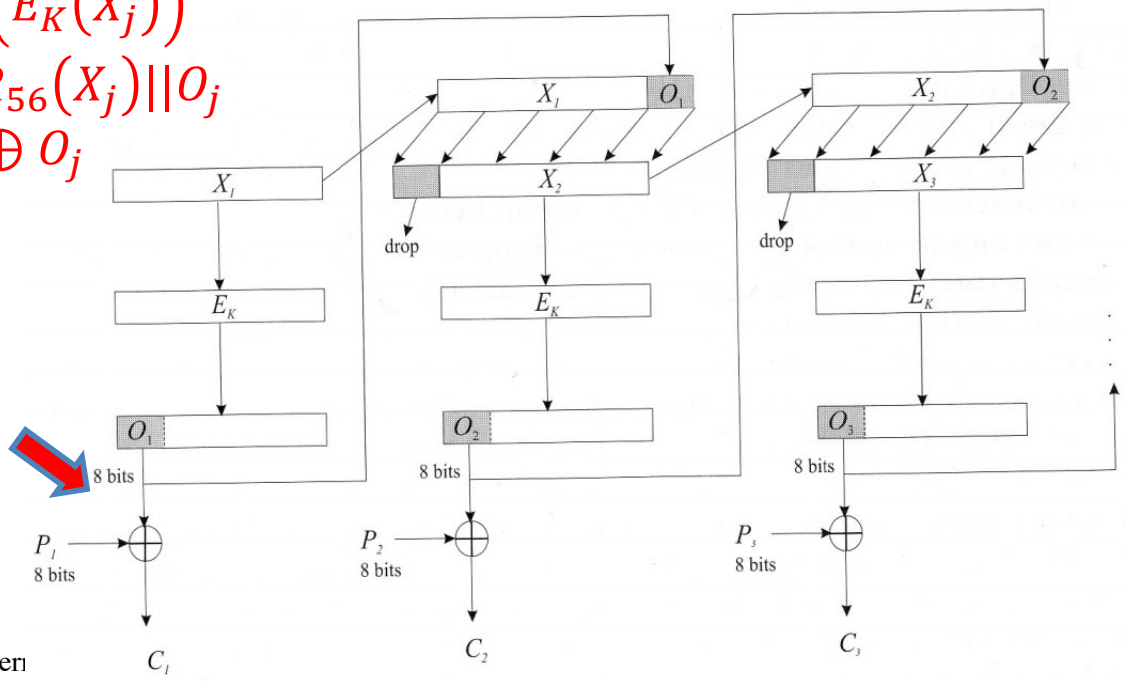
- **Cipher FeedBack (CFB)**
  - The problem with both CBC and ECB methods is that encryption cannot begin for messages below 64 bits.
  - The CFB is in stream mode and works in k-bits
    - A k-bit message can be encrypted without waiting to form the necessary block size
  - Example the case of 8-bits: $P = [P_1, P_2, \dots]$ $with\ the\ size\ of\ P_i\ is\ 8\ bits$
    elect $X_1 of\ 64\ bits$;
    For $j = 1, 2, 3, \dots$
    $O_j = L_8(E_K(X_j))$ // with L_8 represents the 8 bits on the left
    $C_j = P_j \oplus O_j$
    $X_{j+1} = R_{56}(X_j)||C_j$



DES: Data Encryption Standard

- **Output FeedBack (OFB)**
  - Stream mode
  - To overcome the disadvantages of both CBC and CFB modes
  - Avoids error propagation during the iteration process (the dependency between C_i and the next iteration i+1)
  - The formation of the X_i vector does not depend on C_{i-1}
  - $For j = 1, 2, 3, \ldots$

$$O_j = L_8\left(E_K(X_j)\right)$$
$$X_{j+1} = R_{56}(X_j)||O_j$$
$$C_j = P_j \oplus O_j$$



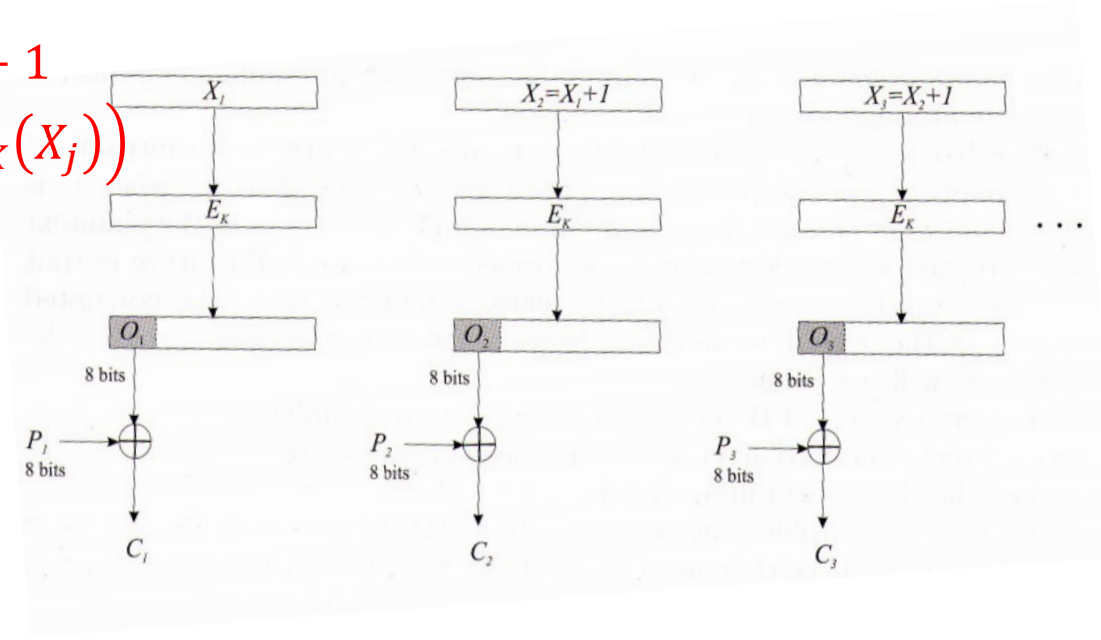
DES: Data Encryption Standard

- **CounTeR (CTR)**
  - The idea of CTR is based on the same principle as the OFB
  - The main difference between CTR and OFB is not to link O_j with the next vector X_{j+1}

$$X_j = X_{j-1} + 1$$

$$O_j = L_8\left(E_K(X_j)\right)$$

$$C_j = P_j \oplus O_j$$



  - The advantage is to be able to parallelize the operations
  - It gives better performance

DES: Data Encryption Standard

- **Limites of DES**
  - Problem of the key size: Initially defined with a 112-bit key, the DES was finally provided by the American authorities with a 56-bit key.
  - The DES with 56bits key size is very likely to be attacked by computer means more or less heavy at the reach of the states
- **Conclusion**
  - Fairly old standard that finally held up well
  - Excellent performance in encryption speed
    - 1 Gigabit/s with low layer (circuit/hardware encryption)
    - 1 Mégabit/s with software encryption
  - Security level for a correct private key algorithm for applications that do not require a high level of confidentiality

DES: Data Encryption Standard

- One of the possibilities to increase the size of the key is the double encryption.
  - Choose the keys K_1 et K_2 nd encrypt the message M:: $\mathrm{E}_{\mathrm{K}_2}(\mathrm{E}_{\mathrm{K}_1}(\mathrm{M}))$

- Triple DES (3DES)
  - Has a security level equivalent to a 112-bit key size
  - At least two methods to implement 3DES :
    - 1st method:
      Choose 3 keys: K_1, K_2, K_3 , the cryptogram $C = E_{K_1}(E_{K_2}(E_{k_3}(M))$
    - 2nd method (also called DESX):
      Choose 3 keys: K_1, K_2, K_3 , the cryptogram $C = K_3 \oplus E_{K_2}(K_1 \oplus M)$

- The AES (Advanced Encryption Standard) algorithm is the replacement of the DES algorithm.

Using DES

- Unix password
  - Using DES with 25 iteration
  - password (Pwd) = DES key to cipher the initialisation vector IV

$$H = DES_{Pwd}(IV)$$

  - The H is recorded in the file /etc/passwd
  - To check the password given to the login, just calculate

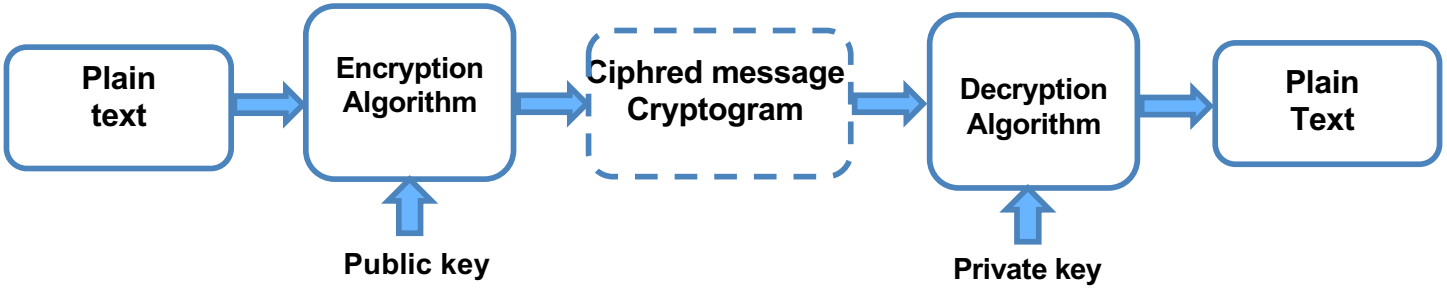
$$DES_{Pwd'}(IV) ?= H$$

AES: Advanced Encryption Standard

- In 1997, a call by NIST (National Institute of Standards and Technology) to replace the DES
  - Possibility to have keys of 128bits, 192bits, 256bits
  - Works with 128-bit blocks
  - Ability to run on different hardware platforms (8-bit processor)
- In 1998, 5 finalists were selected:
  - **MARS** (from IBM)
  - **RC6** (from RSA lab.)
  - **Rijndael** (from Joan Daemen and Vincent Rijmen)
  - **Serpent** (from Ross Anderson, Eli Biham, and Lars Knudsen)
  - **Twofish** (from Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall and Niels Ferguson)
- Rijndael has been selected as AES
  - It works with the 5 modes: ECB, CBC, CFB et CTR

Asymetric encryption

- ***The principle:***
  - Each person has a pair of keys: private and public.
    - The private key: he is the only one to have
    - The public key (generated according to the private key): known by its correspondents,
    - Usually the public key is used to encrypt the message and the private key to decrypt it.



- The idea is to find two functions Ek and Dk' which depend on keys k and k'
  - E_k is the encryption method
  - $D_{k'}$ is the decryption method
  - $\mathbf{D_{k'}(E_k (M)) = M}$ avec $\mathbf{k \neq k'}$
  - It is very difficult to deduce $D_{k'}$ from the knowledge of E_k's encrypted messages

Asymetric encryption

- Asymmetric encryption algorithms are used for encryption and also for digital signatures.
- Among these algorithms: RSA, ELGAMAL, …

- **Advantage**
  - No need to share a secret key
- **Disadvantages**
  - Requires an important computing capacity
  - The algorithms are slow. Encryption and decryption speeds are significantly lower than for secret key algorithms.

- Hybrid schemes allow to increase the speed:
  - A secret key algorithm is used to encrypt the message.
  - A public key algorithm is used to encrypt the key

RSA (Rivest Shamir Adleman)

- RSA proposed by Ron **R**ivest, Adi **S**hamir et Leonard **A**dleman
- The level of security depends on the difficulty of factoring large numbers
- Finding plain text from a key and ciphertext is equivalent to factoring the product of the two prime numbers.

- **Key generation process:**
  - Choose two large prime numbers p and q (of 100 digits) with $n=pq$
  - Choose a random encryption key such that e and $(p\text{-}1)(q\text{-}1)$ are prime between them.
  - Use Euclid's algorithm to compute the decryption key **d** such that
    $$ed = 1\ (mod\ (p-1)(q-1)) \Rightarrow d = e^{-1}\ mod((p-1)(q-1))$$
  - The numbers e and n form the public key and the number d and n the private key.
  - The security of this process depends on the numbers p and q

RSA (Rivest Shamir Adleman)

- Encryption process
  - Split the message ***M*** to be encrypted into blocks $\boldsymbol{M = \{m_1, m_2, \dots, m_l\}}$
  - Each block $\boldsymbol{m_j}$ must have about 200 digits (size of n)
  - The block encryption formula m_j : $\boldsymbol{c_j = m_j^e \ mod\, n}$

- Decryption process
  - Just calculate: $\boldsymbol{m_j = c_j^d \ mod\, n}$
  - Since all operations performed in ***modulo n*** :

$$\boldsymbol{c_j^d = \left(m_j^e\right)^d = m_j^{ed} = m_j^{k(p-1)(q-1)+1} = m_j \times m_j^{k(p-1)(q-1)} = m_j \times (1)^k = m_j}$$

- **Why does it work?**

Euler's function:

If n= p x q and p and q are prime, then $\boldsymbol{\varphi(n) = (p-1)(q-1)}$

if n is prime, then $\boldsymbol{\varphi(n) = n - 1}$

Euler's theorem: if $\gcd(a, n) = 1, then\ a^{\varphi(n)} = 1\ mod\ n$

Fermat's theorem: if *n* is *prime* and *m* is not a multiple of *n* then

$$\boldsymbol{m^{n-1} = 1\ (mod\ n)}$$

RSA (Rivest Shamir Adleman)

Key generation	$n = P * Q$ $d * e = 1 \bmod \Phi(n)$
Encryption	$c = m^e \bmod n$ Public Key(n,e)
Decryption	$m = c^d \bmod n$ private key (d)

- ***The security of the RSA algorithm is based on :***
  - The safety of the system is based on **the difficulty of factoring a large integer n** into two first integers ***p*** and ***q*** (the size of n: 320 bits, 512 bits, 1024 bits also conditions the speed of the algorithms).
  - The non-disclosure of *p* and ***q***
  - Lack of a mathematical method to calculate ***d*** from ***(n,e)***

RSA (Rivest Shamir Adleman)

- **Example: (you can use RSA caculator here)**
  1) Two prime integer **p** = 47, **q** = 71, **n = p . q** = 3337
  2) $\varphi(n)$ **= (p-1).(q-1)** = 46 . 70 = 3220
  3) Choice of **e** where e is prime with **(p-1).(q-1).** Example e = 79
  4) Compute the key ***d*** with ***d*** is the inverse of e

$$d = e^{-1}\ mod\big((p-1)(q-1)\big)$$

One possible solution: **Euler's theorem**

$\mathrm{d} = e^{-1} \Rightarrow ed = e.e^{-1} = 1$, d'après le théorème ($e^{\varphi(n)} = 1\ mod\ n$)

$\mathrm{ed} = 1 = e^{\varphi(n)} = e^{\varphi(n)-1}\ e \Rightarrow d = e^{\varphi(n)-1}\ (mod\ z)$

N.A. : $(79)^{3219}(\mathrm{mod}\ 3220) = 1019$

Another possible solution: using Euclide algorithm

5) **Encrypt the message M :** M = 688232687966683

Decomposition into blocks smaller than n= 3337 => Blocks of **3 digits**

M= 688 232 687 966 668 3

Encrypt 688: $(688)^{79}\ (\mathrm{mod}\ 3337) = 1570$

E(M) = 1570 2756 2091 2276 2423 158

6) **Decipher starting with the first block either** 1570:

$(1570)^{1019}\ (\mathrm{mod}\ 3337) = 688$

RSA (Rivest Shamir Adleman)

- Exercises:
  - We have the cryptogram c=5859 obtained using the RSA algorithm with those parameters: n=11413 et e=7467.
    -Find the plain text with this factorization of n: n=101. 113

We have $\varphi(n) = (p-1)(q-1) = 100\times112 = 11200.$
To find d $(d = e^{-1}\ (mod\ \boldsymbol{\varphi(n)}) \Rightarrow d = (7467)^{-1}\ mod\ 11200)$
We check: ed $= 1\ (mod\ 11200) \Rightarrow ed = k\ 11200 + 1 \Rightarrow k = 2,\ d = 3$

$$m = (c)^d\ mod\ n = (5859)^3\ mod\ 11413 = \mathbf{1415}$$

  - RSA algorithm has the folowing parameters: $\mathrm{n} = 55 = 5\times11$ et $\mathrm{e} = 3$
    - Find the decryption key parameter d.
    - Assuming that $gcd(m, 55) = 1$. Show if $\mathrm{c} = m^3\ (mod\ 55)$ is the cryptogram, then $\mathrm{m} = c^d\ \ (mod\ 55)$ is the associated plain text.

a) We have $\varphi(n) = (p-1)(q-1) = 4\times10 = 40$.
   we look for d knowing $3\mathrm{d} = 1\ (mod\ 40) \Rightarrow d = 27\ car\ 3\ .\ 27 = 81 = 1\ mod\ 40$
b) Here we use Euler's theorem. $3\mathrm{d} = 1 + \mathrm{k}\,\varphi(n)$, then

$$\mathbf{c^d = m^{3d} = m^{1+k\,\varphi(n)} = m\ (mod\ n)}$$

One way hash function

- The one-way hash function is noted ***H(M)=y***
- It operates on a *variable-length M message*, and it provides a *fixed-length* hash value.
- The properties of the hash function are:
  - It is easy to compute y from M
  - It is difficult to calculate *M* from *y* : Non-invertible (Low collision function)
  - It is difficult to find another $M' \neq M$ message such as $\boldsymbol{H(M)=H(M')}$: Collision resistant (High collision function)

- The hash function is used to:
  - Generate **digital signatures**
  - Ensure and to check **data integrity**

One way hash function

Security of a hash function

- The size (length) of « digital fingerprint »
  - It is important to resist the "**Birthday attacks**".
  - This attack can be successful with a fingerprint of length n if at least $\mathbf{2^{n/2}}$ random fingerprints are computed
  - The size of the *64-bit* fingerprint is too small to resist against this type of attack.
    - $\mathbf{2^{32}}$ **possibilities**
  - 128 bits is considered correct because it forces the attacker to compute $\mathbf{2^{64}}$ random fingerprints to find two that have the same fingerprint
  - A 160-bits fingerprint requires the calculation of 2^{80} random fingerprints (NIST recommendation).

One way hash function

- Birthday Attacks
  - Can be used to find **hash function collisions** if the length of the digital fingerprinting is *not long enough*
  - Suppose that **H** is a function with **n-bits** as output ⇒ $N = 2^n$ possibilities
  - Build a list of ***H(x)*** with $\sqrt{N} = 2^{\frac{n}{2}}$ random possibility of x.
    - It has a great chance to have $x_1 \ and \ x_2$ with $H(x_1) = H(x_2)$
  - The more the list of **H(x)** increases the more the probability of having a collision increases.
  - Example:
    - In a class of 23 students, the probability that 2 students have the same birthday is a little more than 50%.
    - A class of 30 students the probability is 70% => This phenomenon is "Birthday paradox".
    - The probability that 2 people have different birth dates: $(1 - \frac{1}{365})$.

      For 3 people: $(1 - \frac{1}{365})(1 - \frac{2}{365})$ and for 23 people: $\left(1 - \frac{1}{365}\right)\left(1 - \frac{2}{365}\right) \dots \left(1 - \frac{22}{365}\right) = 0.493$

      ⇒ $\boldsymbol{The\ probability\ that\ 2\ people\ have\ the\ same\ birthday\ is} \ = 1 - 0{,}493 = 0{,}507$

One way hash function

- Exercices
  - In a family of 4 persons, what is the possibility that there are not 2 persons who have *the same month of birth* (suppose that all the months are equiprobable)?

$$1\times\left(1-\frac{1}{12}\right)\times\left(1-\frac{2}{12}\right)\times\left(1-\frac{3}{12}\right)=\frac{165}{288}=0,573$$

  - Let the hash function $H(x)=\alpha^x \bmod p$. with α is a prime number and alpha does not divide *p*
    Explain why ***H(x)*** is not a good hash function (refer to Fermat's theorem)

According to the Fermat's theorem:
If *p* is prime and α is not a multiple of p then $\alpha^{p-1}=1 \bmod p$

$H(x+p-1)=H(x)$ *because*

$H(x+p-1)=\alpha^{x+p-1} \bmod p=\alpha^x\times\alpha^{p-1} \bmod p=\alpha^x \bmod p$

so the H function does not resist to strong collusion

One way hash function

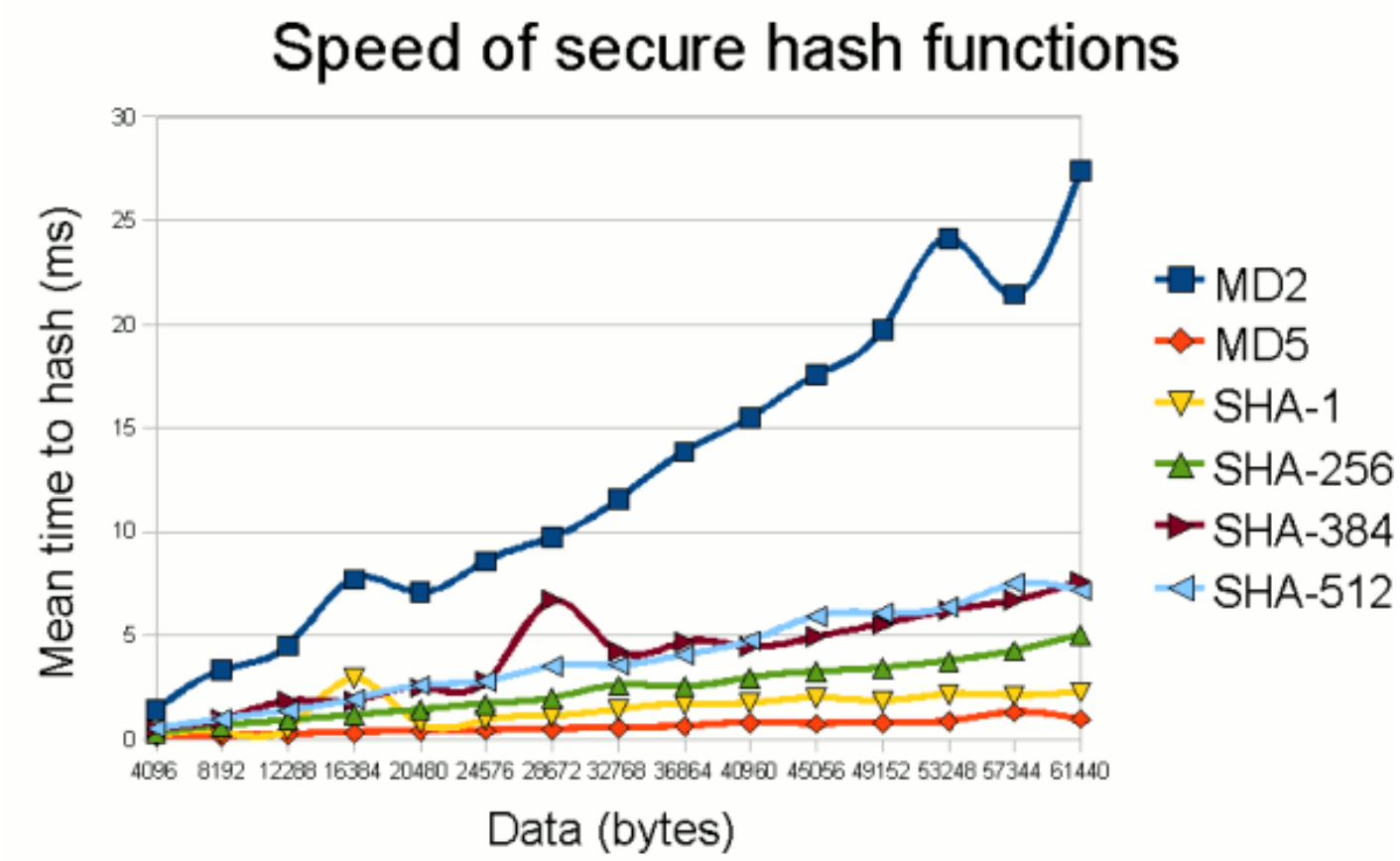
- Exercises (next)
  - Suppose that $\mathrm{H(m)} = M_1 \oplus M_2 \oplus \cdots \oplus M_l$ is a hash function with M_i is 64bits block of message $m = M_1||M_2|| \ldots ||M_l$

    Check the three properties of the hash function

- The first property is the speed to generate the digital footprint. The XOR is a fast operation
- On the other hand the low collision resistance is not checked because

$$H(M_1||0||0||0) = M_1 = m$$

One way hash function - performance



SHA-Secure Hash Algorithm

- SHA developed by *NSA (National Security Agency)* and then donated to *NIST (National Institute of Standards and Technology)*
- Its first version SHA-0 was published in *1993*.
- SHA-1 is the version recommended by NIST after the correction of SHA-0.
- SHA-1 produces a *160-bit digital fingerprint*
- SHA-1 uses the same principle as *MD4* and *MD5* (MD: Message Digest).

- SHA-1 defines some functions and constants:

$$f_t(B,C,D) = \begin{cases} (B \wedge C) \vee ((\neg B) \vee D) & if\ 0 \le t \le 19 \\ B \oplus C \oplus D & if\ 20 \le t \le 39 \\ (B \wedge C) \vee (B \wedge D) \vee (C \wedge D) & if\ 40 \le t \le 59 \\ B \oplus C \oplus D & if\ 60 \le t \le 79 \end{cases}$$

  - $K_t = \begin{cases} 5A827999 & if\ 0 \le t \le 19 \\ 6ED9EBA1 & if\ 20 \le t \le 39 \\ 8F1BBCDC & if\ 40 \le t \le 59 \\ CA62C1D6 & if\ 60 \le t \le 79 \end{cases}$

SHA-Secure Hash Algorithm

SHA-1 Algorithm

- The message ***M*** of variable size ***L*** is divided into several blocks of size ***512***
- *M=[$m_1, m_2, \dots, m_l$] with* $\boldsymbol{l = \left[\frac{L}{512}\right]}$ *, (is completed so that its length is a multiple of 512)*
  - The M padding technique: a bit 1 is added followed by as many 0's as necessary so that 64 bits are missing compared to a multiple of 512 , the remaining 64 bits reserved for the message length before padding.
- Five variables are initialized as follows:
  $\boldsymbol{H_0 = 0x67452301, H_1 = 0xEFCDAB89, H_2 = 0x98BADCFE,}$
  $\boldsymbol{H_3 = 0x\ 10325476, H_4 = 0xC3D2E1F0}$
- The main loop (processes each block m_j) has 4 rounds of 20 operations each
- Transform m_j as follows: $\boldsymbol{m_j = W_0\ ||W_1|| \dots ||W_{15}}$ with $\boldsymbol{W_j}$ has 32 bits
- For t=16 to 79 : $\boldsymbol{W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \triangleleft 1}$
- $\boldsymbol{A = H_0; B = H_1; C = H_2; D = H_3; E = H_4;}$
- For t =0 to 79 : $\boldsymbol{T = (A \triangleleft 5) + f_t(B, C, D) + E + W_t + K_t \quad ; E = D; D = C;}$
  $\boldsymbol{C = (B \triangleleft 30); B = A; A = T;}$
- $H_0 = H_0 + A; H_1 = H_1 + B\ ; H_2 = H_2 + C; H_3 = H_3 + D; H_4 = H_4 + E;$
- Output: $\boldsymbol{H_0||H_1||H_2||H_3||H_4}$ is 160-bits

Message Authentication Codes (MAC)

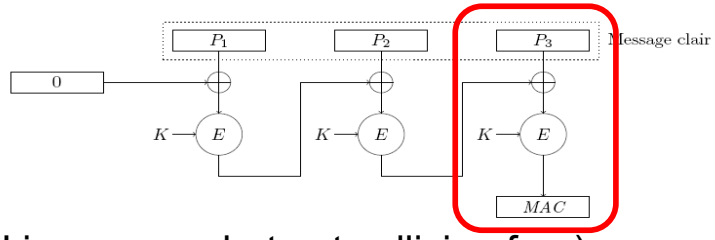
- MAC is a one-way hash function dependent on a key
- Whoever has the key can check the fingerprint (digest)
- This is a very useful function to prove integrity and authentication without providing confidentiality.

- A CAM is a family of functions $\boldsymbol{H_k}$ parameterized by a secret key k and which have the following properties :

  - **Compression**: H_k akes an input of *any length* and produces an output of fixed size
  - **Easy to compute**: for a known function H_k, given k and an input x, it is easy to calculate $H_k(x)$
  - **Collusion resistance:** given any number (including zero) of pairs $(X_i, H_k(X_i))$, it is impossible in practice to calculate, without knowledge of the k-key, $(X, H_k(X))$ for any new entry $X \neq X_i$

Message Authentication Codes (MAC)

- Three ways to create MAC
  - ***using secret encryption algorithm***
    - CBC-MAC (Cipher Block Chaining Message Authentication Code) :
      - Encrypt the message with a block cipher algorithm in CBC (or CFB) mode
      - The CAM is the last encrypted block



  - ***Using one-way hash function H_k***
    - $\boldsymbol{H(M,K)}$ this scheme has some problems (if H is one-way but not collision free)
    - The best solution is: : $\boldsymbol{H(K,M,K)}$ ou $\boldsymbol{H(K_1,M,K_2)}$ $where\ K_1 and\ K_2\ are\ not\ the\ same\ key$
    - The secure and safest schemes are :
      - $H(K_1, H(K_2, M))$
      - $H(K\ \ , H(K\ \ , M))$
      - $H(K\ \ , p, M, K)$ $where$ p is used to fill k until a message block is obtained
    - The HMAC (Key Hashed Message Authentication Code) standard uses the following scheme: $\boldsymbol{H((K \oplus opad)||H((K \oplus ipad)||M))}$

Message Authentication Codes (MAC)

- ***Using a combination of hash function and encryption algorithm***
  - Calculate the digest of the message M ($H(M)$) then encrypt only this digest with a secret key algorithm $\mathrm{E_k(H(M))}$

- **Different possible combination schemes for authentication and encryption:**
  - **Encrypt-and-MAC** (encrypt the plain text, compute a MAC on the plain text and concatenate the encrypted and the MAC),
    => $\mathbf{E_k(M)||MAC(M)}$
  - **MAC-then-encrypt** (we calculate the MAC of the plain text and encrypt the concatenation of the plain text and the MAC)
    => $\mathbf{E(M||MAC(M))}$
  - **Encrypt-then-MAC** (we encrypt the plain text and calculate the MAC on the encrypted text).
    => $\mathbf{E(M)||MAC(E(M))}$

The electronic signature

- Allows the receiver of a message to verify the identity of the sender
- Therefore the sender cannot then deny the content of the message,
  - the signature ensures: identification, non-repudiation and integrity
- Properties:
  - The signature cannot be imitated
  - It proves that the signatory deliberately signed the document
  - The signature authenticates the signatory
  - Only the signatory may have signed
  - The signature belongs to a single document (it is not reusable)
  - The signed document cannot be partially or totally modified
  - The signature cannot be denied

DSA (Digital Signature Algorithm)

- In 1991, NIST introduced the Digital Signature Algorithm (DSA) compliant with the Digital Signature Standard (DSS).
- DSA is a variant of signature algorithms such as ELGAMAL

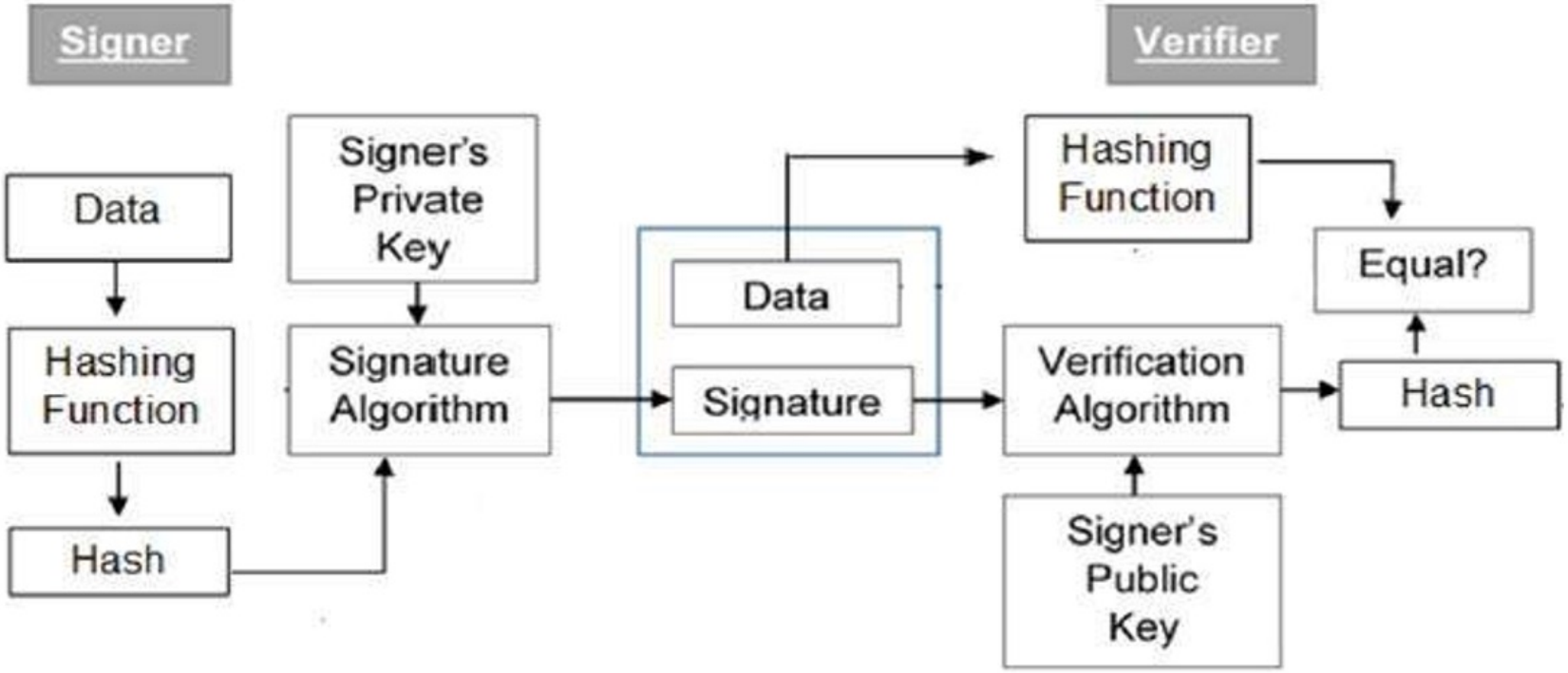
- The DSA algorithm uses the following parameters:
  - p: a prime number of L bits long ($\mathrm{L} \in [512 - 1024]$) and it is a multiple of 64
  - q: a prime factor of $(\mathrm{p} - 1)$, 160 bits long
  - $g = h^{(p-1)/q} \ (mod\ p)$ where h is any number less than $\mathrm{p} - 1$ and greater than 1
  - x: a number less than q
  - $y = g^x mod\ p$
  - The first three parameters p, q and g are public, and the private key is $\mathbf{x}$ and the public key is $\mathbf{y}$

DSA (Digital Signature Algorithm)

- **Signing process: -** A message m is signed as follow:
  - Choose an integer k randomly from [1..q-1]
  - Compute: $r = (g^k mod\ p) mod\ q$ where $r \neq 0$
  - Compute: $s = (k^{-1}(H(m) + x \times r))\ \ mod\ q$
  - The signature is: **(r,s)**

- **Signature verification :** - verify if the signature (r, s) is valid for the message m
  - **Verify that** $0 < r < q$ **and** $0 < s < q$
  - **Compute:** $w = s^{-1}\ mod\ q$
  - **Compute:** $u_1 = (H(m) \times w)\ \ mod\ q$
  - **Compute:** $u_2 = (r \times w)\ \ mod\ q$
  - **Compute:** $v = ((g^{u_1} \times y^{u_2})\ \ mod\ q) mod\ p$
  - **If v=r then the signature is valid**

DSA (Digital Signature Algorithm)



The electronic signature

Using RSA algorithm

- Suppose Alice agrees to sign Bob's document (*m*).
- Signature generation
  - Alice generates two prime numbers p and q. Then she computes $\mathbf{n} = \boldsymbol{p}\times\boldsymbol{q}$
  - She selects $\mathbf{e_A}$ where $1 < \mathbf{e_A} < \boldsymbol{\varphi}(\boldsymbol{n})$ and $\gcd(\mathrm{e_A}, \varphi(n)) = 1$
  - She computes $\boldsymbol{d_A}$ where $\mathbf{e_A}\ \boldsymbol{d_A} \equiv \mathbf{1}\ \boldsymbol{mod}\ \boldsymbol{\varphi}(\boldsymbol{n})$
  - Alice publishes $(\mathbf{e_A}, \mathbf{n})$ and she keeps the parameters secret $(\boldsymbol{d_A}, \boldsymbol{p}, \boldsymbol{q})$
  - Alice signs the document ***m*** as follows $\mathbf{sign}(\mathbf{m}) = \mathrm{y} = \boldsymbol{m}^{\boldsymbol{d_A}}\ \boldsymbol{mod}\ \boldsymbol{n}$
  - Then ***m*** and ***sign(m)*** are published
- Verification procedure
  - For verification, the parameters $(\mathbf{e_A}, \mathbf{n})$ are needed
  - Computes $\mathrm{Z} = \boldsymbol{y}^{\boldsymbol{e_A}}\ \boldsymbol{mod}\ \boldsymbol{n}$, if *z=m* then the signature is valide

- If an attacker wants to replay Alice's signature then he cannot associate her with the modified document (associate y to m') .

The blind electronic signature

- Blind signature (important concept for electronic vote)
  - Alice signs the message without being able to read it
  - Blind signature procedure
    - Alice generates two prime numbers p and q.
      Then she computes $\mathbf{n} = \boldsymbol{p}\times\boldsymbol{q}$
    - She selects $\mathbf{e_A}$ where $1 < \mathbf{e_A} < \boldsymbol{\varphi}(\boldsymbol{n})$ and $\gcd(\mathrm{e_A}, \varphi(n)) = 1$
    - Parameters $(\mathbf{e_A}, \mathbf{n})$ are public and the parameters $(\boldsymbol{d_A}, \boldsymbol{p}, \boldsymbol{q})$ are secret
    - Bob choose a random integer $\boldsymbol{k}$ where $\gcd(k \ \ , n) = 1$
    - Bob computes $\mathrm{t} = (\boldsymbol{k^{e_A}}\times\boldsymbol{m})\boldsymbol{mod\ n}$ and he send t to Alice
    - Alice signs t as follow: $\mathbf{s} = \boldsymbol{t^{d_A} mod\ n}$ and he send s to Bob
    - Bob computes : $\frac{\boldsymbol{S}}{\boldsymbol{k}}\ \boldsymbol{mod\ n}$ then the message is : $\boldsymbol{m^{d_A} mod\ n}$
    - $\frac{\boldsymbol{S}}{\boldsymbol{k}} = \frac{\boldsymbol{t^{d_A}}}{\boldsymbol{k}} = \frac{\boldsymbol{k^{e_A x\, d_A}}\times\boldsymbol{m^{d_A}}}{\boldsymbol{k}} = \frac{\boldsymbol{k^1}\times\boldsymbol{m^{d_A}}}{\boldsymbol{k}} = \mathrm{m}$ because $\mathrm{e_A}\times d_A = 1\ mod\varphi(n)$
    - $\boldsymbol{k^{e_A}}\times\boldsymbol{m\ mod\ n}$: this does not give any information about the message m

Threshold cryptography

- Threshold *scheme-(m, n) where m<= n*
- Proposed by Adi Shamir and George Blakley
- The secret (private) key is divided into *n* elements and the association of *m* (where *m<= n*) elements enables to get the message
- Secret key is shared among trustees s.t.
- Trustees can decrypt or sign only if enough cooperate (m)
- Faulty trustees can’t prevent decryption or signature

Multiple private keys for single public key

- Using RSA
  - n is the result of the multiplication of two prime numbers q and p (n=pq)
  - instead of choosing e and d ($ed = 1 \left(mod\ (p-1)(q-1)\right)$), it is necessary to choose t keys such as: $\left(k_1 \times k_2 \times \cdots \times k_t \equiv 1\ (p-1)(q-1)\right)$
  - $M^{k_1 \times k_2 \times \cdots \times k_t} = M$

- **Example**
  - We have 5 keys: $k_1, k_2, k_3 \times k_4 \times k_5$, a message encrypted with k_3 and k_5, it can be decrypted using k_1, k_2, and k_4
  - *Encryption process*: $C \equiv M^{k_3 \times k_5} mod\ n$
  - *Decryption process*: $M \equiv C^{k_1 \times k_2 \times k_4} mod\ n$

Multiple private keys for single public key

Example

- Suppose Alice and Bob must both sign a document to make it valid.
- We have three keys: k_1 for Alice, k_2 for Bob and k_3 a public key.
- Step 1: Alice signs M and sends to Bob $M' \equiv M^{k_1} mod\ n$
- Step 2: Bob can get M from M': $M \equiv M'^{k_2 \times k_3} mod\ n$
- Step 3: Bob can add his signature $M'' \equiv M'^{k_2} mod\ n$
- Step 4: Anyone can verify the signature with the public key k_3.

$$M \equiv M''^{k_3} mod\ n$$

The constraint: it is necessary to have a key distribution system with a trusted third party.

Multiple private keys for single public key

- For a missile launch program and for security reasons, you need the agreement of ***three officers out of five***.
- The case of a mechanical launching system :
  - A key must be given to each of the five officers...
  - At least three officers must have their key in the right lock.
- Exercice
  - To build a more sophisticated system. A general and two colonels can launch the missile, but if the general is busy then it takes five colonels to authorize the launch of the missile.
    Give the parameters of the launch controller so that it requires five keys.

Give three keys to the general and one key to each colonel.
The general and 3 colonels: can launch the missile.
Five colonels: can launch the missile

Digital certificates and trust model

- The digital certificates answer the following question:

  "***How can I be sure of the public key?"***

- The idea is to use a *trusted third party* to certify the public key
- A certificate => electronic document, which contains (standard X509):
  - the public key
  - the identity of its owner
  - the validity date
  - A serial number
  - the identity of the trusted third party

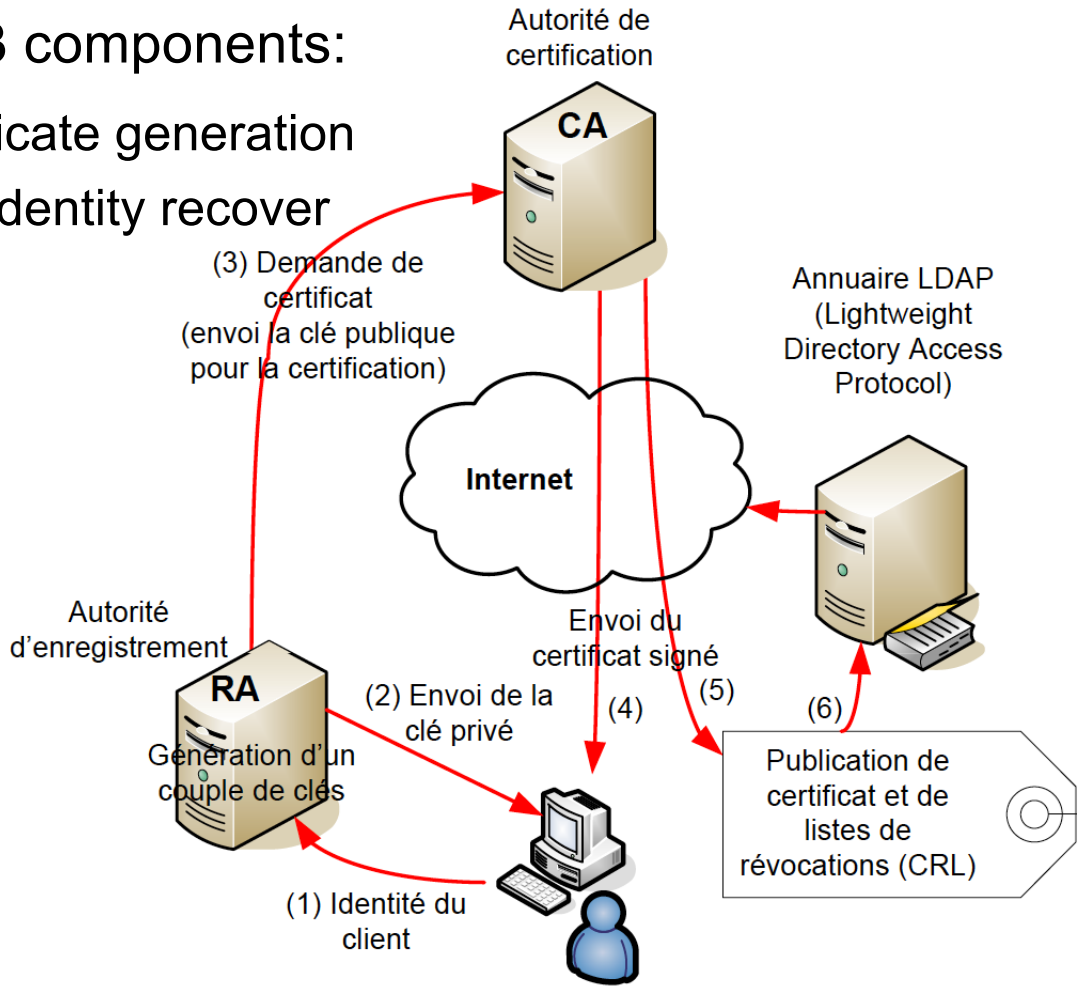
- Example of a trusted third party:
  - VeriSign, GTE, AT&T
  - CERTplus, Certinomis (in France)

PKI (Public Key Infrastructure)

- It is an infrastructure that allows to ensure security with the use of public key cryptography.

- A PKI architecture, consists of 3 components:
  - **Certification Authority:** certificate generation
  - **Registration authority:** user identity recover
  - **The key distribution system:** publication of certificates

- How it works



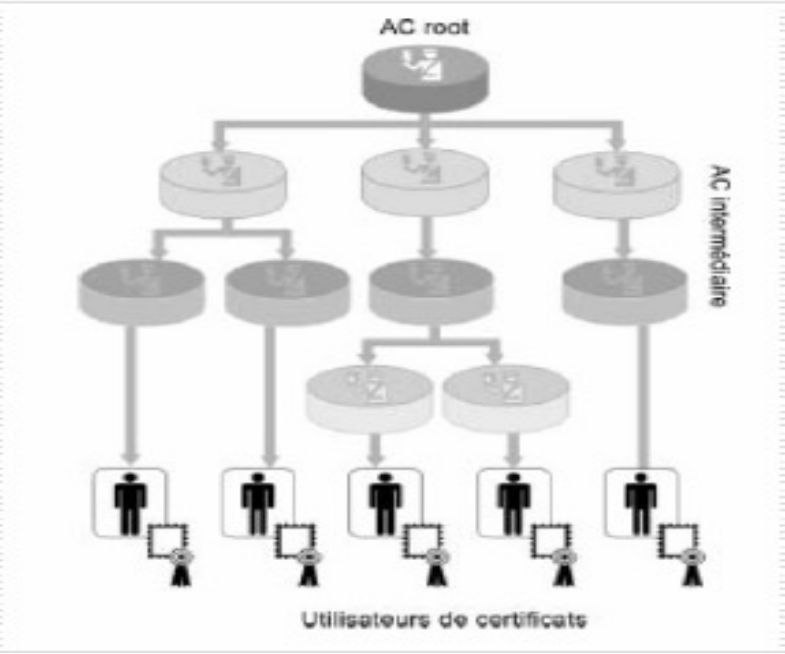
PKI (Public Key Infrastructure)

- Key management:
  - Need for trusted publication of the public key
  - The publication must ensure the validity and ownership of the key.
  - The publication of certificates is done using LDAP (RFC2251) directory structures.
  - Revoked certificates are grouped in revocation lists (CRLs).

- Available solutions
  - Commercial:
    - Unicert of Baltimore Technologies
    - from Entrust Technologies
    - TrustyKey from TrustyCom
  - Free: OpenPKI based on OpenSSL (http://www.openssl.org)

Trust models

- **Hierarchical (centralized) model:**
  - There is a higher authority that everyone trusts



Trust models

- **Peer to peer model:**
  - Lack of the certification authority (CA), this is the model used by the PGP.
  - The principle: each one generates its own certificate and publishes it in a directory and distributes it to its correspondents,
  - The idea is: "**friends of my friends are my friends**".
  - Example of use