

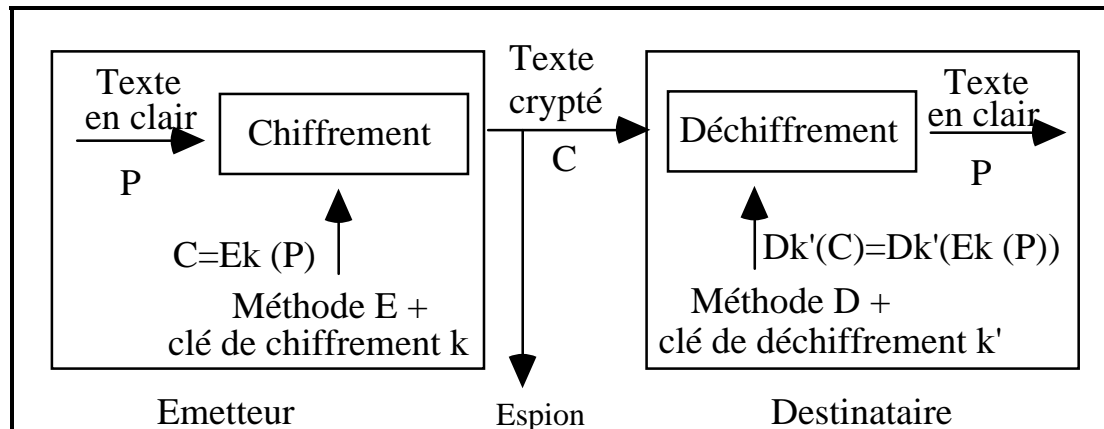
# **LES TECHNIQUES DE CRYPTOGRAPHIE**

**G Florin, S Natkin**

**Unité de valeur  
Systèmes et applications répartis**

# Introduction

## La cryptographie: l'outil de base indispensable de la sécurité



Pour réaliser des propriétés de sécurité:

- **confidentialité**
- **intégrité**
- **authentification**

On utilise des **cryptosystèmes** basés sur des fonctions  $E$ ,  $D$ , et des clés  $k$ ,  $k'$  telles qu'une partie soit ignorée du reste du monde.

## Le chiffrement

Transformation d'un message pour en  
cacher le sens

- L'émetteur doit transmettre un message  $M$   
(en clair)  $M \in \text{Messages\_à\_envoyer}$
- Il construit un texte chiffré  $C$  au moyen  
d'une fonction  $C = E_k (M)$  qui dépend clé  $k$   
 $C \in \text{Messages\_chiffrés}$

## Le déchiffrement

Opération inverse du chiffrement  
Récupération d'un message en clair

- La fonction de déchiffrement  $D_{k'} (C)$   
 $\text{Messages\_chiffrés} \rightarrow \text{Messages\_à\_envoyer}$
- Possède la propriété d'être l'inverse à  
gauche de  $E_k$  :  
$$D_{k'} (C) = D_{k'} (E_k (M) ) = M$$

## Cryptosystèmes

- **Décrypter ou casser un code** c'est parvenir au texte en clair sans posséder au départ les règles ou documents nécessaires au chiffrement.
- L'art de définir des codes est la **cryptographie** (un spécialiste de cryptographie est un **cryptographe**).
- L'art de **casser** des chiffres est la **cryptologie ou cryptanalyse** (un spécialiste de cryptanalyse est un cryptanalyste, cryptologue ou casseur de codes)
- **Un cryptosystème** est l'ensemble des deux méthodes de chiffrement et de déchiffrement utilisable en sécurité.

## Propriétés générales des cryptosystèmes

L'usage ancien du chiffre et l'usage actuel en informatique ont conduit aux contraintes suivantes:

a) Réalisation simple et rapide du chiffrement et du déchiffrement (pour atteindre des débits élevés).

b) Éviter un encombrement important des clés.

c) Une méthode de cryptographie (fonctions E et D) doit être stable. On ne peut la changer que très rarement.

d) Elle est le plus souvent publiée (largement connue).

f) Un cryptosystème dépend de paramètres (clés) qui doivent pouvoir être modifiés aisément et fréquemment.

g) On estime que la sécurité ne doit pas dépendre du secret des algorithmes E et D mais uniquement du secret des clés  $k$  et  $k'$  (exception pour le domaine militaire).

## **Différentes difficultés d'attaque d'une méthode de cryptage**

Crypter ne se justifie que relativement à l'existence d'attaquants ou cryptanalystes dont le travail est plus ou moins difficile.

### **a) - L'attaque à textes chiffrés**

On dispose seulement de textes chiffrés

### **b) - L'attaque à textes en clair connus**

On dispose de quelques morceaux de texte en clair et de leur cryptage

### **c) - L'attaque à textes en clair choisis**

On peut faire crypter ce que l'on veut par la méthode de cryptage et voir ce qu'elle produit

### **Remarque.**

Une bonne méthode doit résister aux attaques de type c.

## Les deux grandes catégories de cryptosystèmes

### *Les chiffres symétriques (à clé privée)*

- Les méthodes de chiffrement et de déchiffrement  $D_{k'}$  et  $E_k$  sont très voisines (voire identiques) et sont liées du point de vue du secret.

- On peut déduire très facilement l'une des clés de la connaissance de l'autre : pratiquement  $k = k'$ .

Exemple: Décalage des lettres de l'alphabet de  $n$  positions (chiffre de César).

- Le secret des clés doit être très bien gardé.

## *Les chiffres asymétriques (à clé publique)*

- On choisit  $D_k'$  et  $E_k$  telles qu'il est **très difficile de déduire  $D_k'$  de la connaissance de  $E_k$ .**

Il faut trouver une fonction dont la fonction inverse est difficile à déterminer.

- On peut donc rendre  $E_k$  publique (**notion de clé publique**) connue de tous dans un annuaire car c'est très pratique.

Tout le monde peut chiffrer:  $C = E_k (M)$

- Par contre la clé  $D_k'$  (**la clé privée**) reste secrète et particularise chaque utilisateur.

Seul le destinataire peut déchiffrer

$$M = D_{k'} (C)$$

- Propriété complémentaire (très utile) d'un système à clé publique: **la commutativité.**

$$D_{k'} ( E_k ( M ) ) = E_k ( D_{k'} ( M ) ) = M$$



## **Problème de la cryptographie** **Retarder le travail des cryptanalystes**

La sécurité d'un cryptosystème repose en fait sur l'analyse de la **complexité** des algorithmes définis et sur les puissances de calcul disponibles pour une attaque.

=> domaine évoluant en permanence avec la recherche

### **Exemples :**

- L'algorithme du sac à dos  
Proposé comme une solution à clé publique => Rejeté en quelques années

- Le DES 56 bits  
=> Déclassifié en 1988

**Le problème est de faire en sorte que la durée nécessaire pour casser un code soit supérieure à la durée de validité des données.**

# Plan de l'exposé

## Chapitre I

- Les chiffres à clés privées
  - . Systèmes classiques de cryptographie
  - . Chiffres symétriques

## Chapitre II

- Les chiffres à clés publiques
  - . Systèmes modernes de cryptographie
  - . Chiffres asymétriques

## Chapitre III

- Les signatures numériques  
(fonctions de hachage à sens unique).

**Chapitre I**

**LA CRYPTOGRAPHIE  
CLASSIQUE  
(à clés privées)**

## Principe général de la cryptographie à clé privée

- La connaissance de la méthode et de la clé de chiffrement et celle de la méthode et de la clé de déchiffrement **se déduisent facilement l'une de l'autre.**

- Les deux méthodes et les clés sont connues de l'émetteur et du destinataire  
=> **L'émetteur et le destinataire doivent se mettre préalablement d'accord sur un secret (la clé) pour utiliser le chiffre.**

### Deux problèmes

- L'échange préalable à toute communication sécurisée d'un secret

**("la distribution de clés)**

- Dans un réseau de  $N$  entités susceptibles de communiquer secrètement il faut distribuer  **$N*(N-1)/2$  clés.**

# Les méthodes de chiffrement par substitution

## Principe général

A chaque lettre ou groupe de lettres on substitue une autre lettre ou un autre groupe de lettres.

### La substitution simple

(substitution mono alphabétique)

Pour chaque lettre de l'alphabet de base on se donne une autre lettre utilisée dans le texte chiffré.

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

Exemple historique: Le chiffre de César

**On décale les lettres de 3 positions**

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

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

## Les techniques d'attaque statistique

- Analyse statistique des textes cryptés.
- Détermination des fréquences d'apparition des symboles
- Comparaison avec les fréquences types caractéristiques des langues

Fréquences d'apparition (en anglais)

Lettres	Digrammes	Trigrammes
E 13,05	TH 3,16	THE 4,72
T9,02	IN 1,54	ING 1,42

Une analyse statistique d'un texte suffisamment long permet de casser un code mono ou même poly-alphabétique

Le problème est de disposer:

- de puissance de calcul
- de suffisamment de texte en regard de la longueur des clés utilisées.

## La substitution polyalphabétique

- Une attaque est facile avec un seul alphabet.
- Pour rendre la méthode plus résistante on utilise une suite de chiffres mono alphabétiques réutilisée périodiquement.

### Exemple : le chiffre de Vigenere

On prend les 26 chiffres de César.

Les chiffres associés aux 26 décalages possibles sont représentés par une lettre.

Le chiffre avec décalage de  $k$  est associé à la  $k$  ième lettre de l'alphabet.

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

- On choisit une clé de répétition comme une suite de lettres: un mot ou une phrase ou un livre
- Cette clé répétée indéfiniment vis à vis de chaque lettre d'un texte à chiffrer sert à déterminer le chiffre à utiliser.

## **Autres substitutions**

### **Les substitutions homophoniques**

Au lieu d'associer un seul caractère crypté à un caractère en clair on dispose d'un ensemble de possibilités de substitution de caractères dans laquelle on choisit aléatoirement.

### **Les substitutions de polygrammes**

Au lieu de substituer des caractères on substitue par exemple des digrammes (groupes de deux caractères)

- Au moyen d'une table  
(système de Playfair)
- Au moyen d'une transformation mathématique (système de Hill).



## Les chiffres de substitution à masque jetable

Pour éviter les attaques statistiques il faut utiliser une substitution qui rend le texte crypté non analysable statistiquement.

### Solution des clés jetables

- Générer une clé qui est **une suite binaire uniformément distribuée**  
Ex: utilisation d'un phénomène physique

- Chaque clé  $K_M$  ne sert **qu'une fois (pour un seul message)**.

- Pour chiffrer un message **faire le ou exclusif du message et de la clé.**

$$C = M \oplus K_M$$

- Pour déchiffrer un message l'opération est la même:

$$M = C \oplus K_M = M \oplus K_M \oplus K_M$$

**Le chiffre à masque jetable est  
incassable.**

- On connaît un message chiffré C  
On essaye toutes les clés possibles  $K_x$   
On obtient tous les messages possibles  
(syntaxiquement corrects):

$$\forall \text{Mess} \exists K_x : \text{Mess} = C \oplus K_x$$

On a aucun moyen de décider qu'un message correct est le message émis plutôt qu'un autre message.

- Un cryptosystème est incassable si

$$\text{Prob}(M/C) = \text{Prob}(M)$$

c'est à dire que M et C sont indépendants.

$$\text{Prob}(M/C) = \text{Prob}(M \text{ et } C) / \text{Prob}(C)$$

$$\text{Prob}(M \text{ et } C) = \text{Prob}(M) \text{Prob}(C).$$

Ici toutes les clés sont équiprobables donc C a pu être construit à partir de n'importe quel message M':

=> M et C sont indépendants.

**Remarque:** L'information apportée par la réception de C est en apparence nulle:

$$I = -\log_2(\text{Prob}(M/C) / \text{Prob}(M)) = 0$$

En fait on utilise la connaissance commune de la clé.

<p style="text-align: center;"><b>Conclusion:</b> <b>méthode des masques jetables</b></p>
---

**Avantage**

- Perfection de la méthode en sécurité.

**Inconvénients**

- **Volume des clés = Volume des données**  
    Connu des deux extrémités.
- **Problème de synchronisation**  
    Si l'on perd une seule donnée on ne sait plus décrypter.

# Les méthodes de chiffrement par transposition

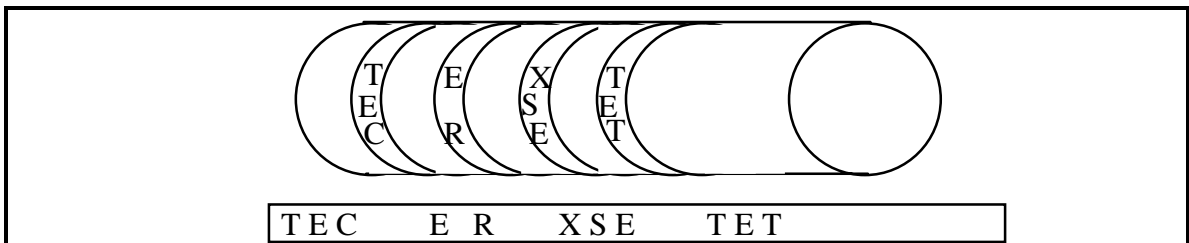
## Principe général

On procède à un réarrangement de l'ensemble des caractères (une transposition) qui cache le sens initial.

La technique est très peu résistante aux attaques statistiques.

## Exemple

Le plus souvent on utilise deux visions géométriquement différentes du texte.



- On enroule une fine langue de papyrus ou de peau sur un tambour d'un diamètre donné (technique assyrienne 400 av JC).
- On écrit horizontalement un texte sur la lamelle enroulée.
- Quand la lamelle est déroulée les lettres sont incompréhensibles.
- Pour décrypter le message il faut un cylindre du bon diamètre.

## Exemple de transposition à base matricielle

- Le message en clair est écrit dans une matrice.
- La clé est la matrice.
- La technique de transposition de base consiste à lire la matrice en colonne.

Exemple (6,5):

M	E	S	S	A	G
E		S	E	C	R
E	T		A		T
R	A	N	S	P	O
S	E	R			

Le message crypté est donc:

**MEERSE TAESS NRSEAS AC P GRTO**

## Chiffre à transposition avec chiffre à substitution simple.

- On combine la transposition avec une substitution et on réarrange l'ordre des colonnes selon une permutation qui est ajoutée à la matrice pour former la clé.

Exemple d'ordre d'exploration des colonnes 1 6 4 3 2 5, le texte crypté est:

"MEERSGRTO SEAS SN NRE TAEAC P "

- On peut générer et mémoriser simplement des permutations en prenant une clé sous forme d'un mot qui ne comporte pas deux fois la même lettre

On numérote les colonnes dans l'ordre ou apparaissent les lettres du mot dans l'alphabet.

Exemple ESPOIR correspond à la permutation 1 6 4 3 2 5.

## Le DES "Data Encryption Standard"

-Dès le début des années 1960 la technologie des circuits intégrés permet de travailler à des circuits combinatoires complexes permettant d'automatiser:

la méthode de substitution.

la méthode de transposition.

=> Idée d'appliquer ces techniques en cascade dans un produit de chiffres.

- Mise au point à partir de 1968 d'une méthode de cryptage basée sur 16 étages de substitutions et transpositions basés sur des clés (IBM)

- Appel d'offre NBS (1973) pour la mise au point d'un système de cryptographie

- Proposition IBM (1975)

- Adoption définitive et normalisation du DES d'IBM (1978) par le NBS ("National Bureau of Standards").

-Normalisation ANSI X3.92 connue sous le nom de DEA ("Data Encryption Algorithm").

## Principes Généraux du DES

Choix possibles pour la sécurité

- Méthodes simples de chiffrement et des clés très longues .

Le DES

- Produit de transpositions et substitutions nombreuses et compliquées pour une clé relativement courte

=> facilité de transport.

- Les chiffres à substitution et à transposition sont faciles à réaliser en matériel.

**Les boîtes de transposition**

**"P-Box"**

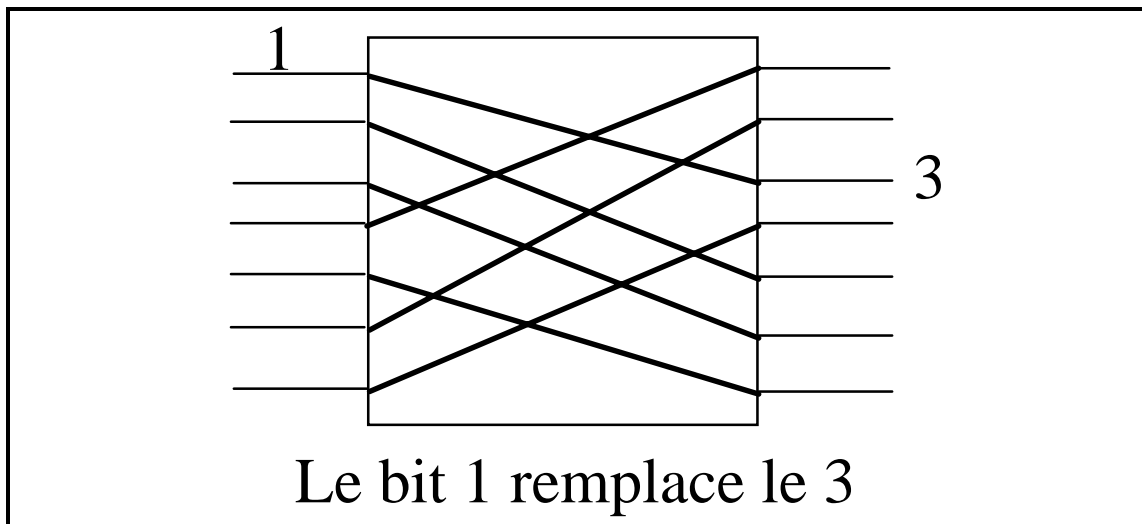
**Les boîtes de substitution**

**"S-Box"**



## Boîte de transposition (P - box "Permutation box")

Exemple pour 8 bits (solution matérielle)



Facile à réaliser par simple câblage  
Autre solution (logicielle) par des tables

Exemple de transposition sur 64 bits

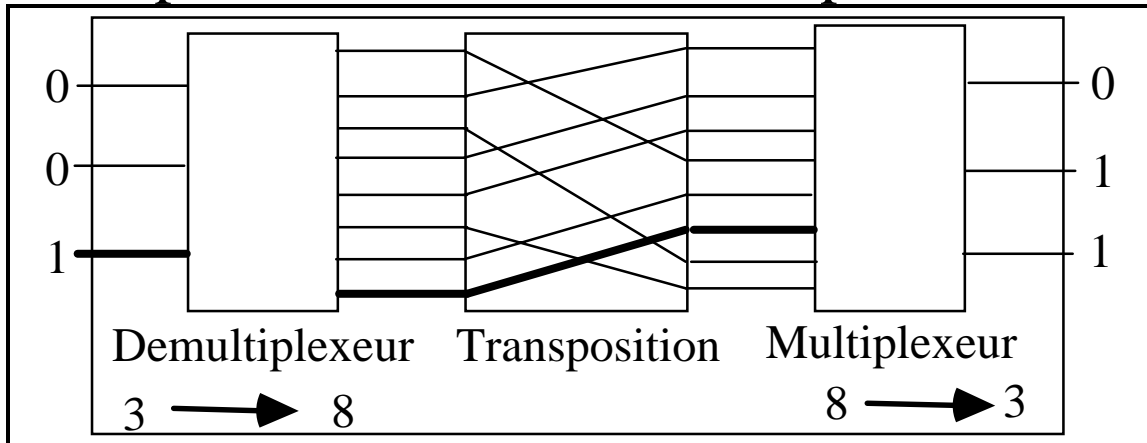
La permutation initiale du DES

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

Le bit 1 remplace le 58

## Boîte de substitution (S - box)

Exemple de solution matérielle pour 3 bits



- Trois bits sélectionnent un fil en sortie
- L'ensemble subit une transposition.
- Le résultat est remultiplexé sur 3 bits

### Solution par consultation de table

Pour une configuration d'entrée on sélectionne directement au moyen d'une table la configuration de sortie.

Exemple: Table S-1 du DES

Approche particulière on substitue à une valeur sur 6 bits une valeur sur 4 bits.

Les deux bits faible et fort sélectionnent la ligne, les 4 bits intermédiaires la colonne.

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

# DES - Caractéristiques

## Deux modes

- Mode cryptage par bloc de 64 bits
- Mode cryptage à la volée ("stream")  
(octets par octets avec des registres à décalage)

## Utilisation d'une clé sur 56 bits

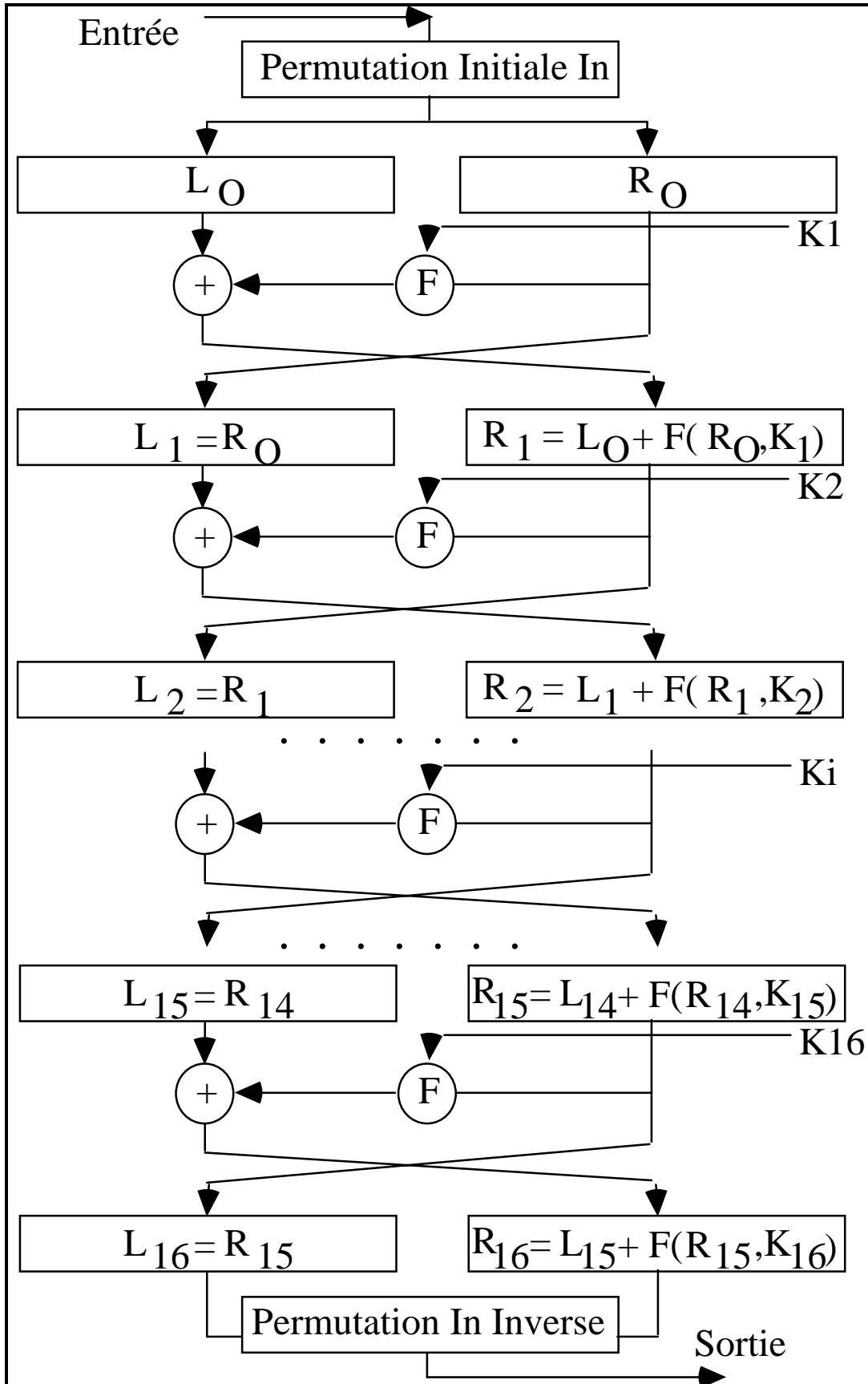
En fait 8 fois 7 bits avec une parité  
(initialement 128 bits)

## 19 étages de logique combinatoire

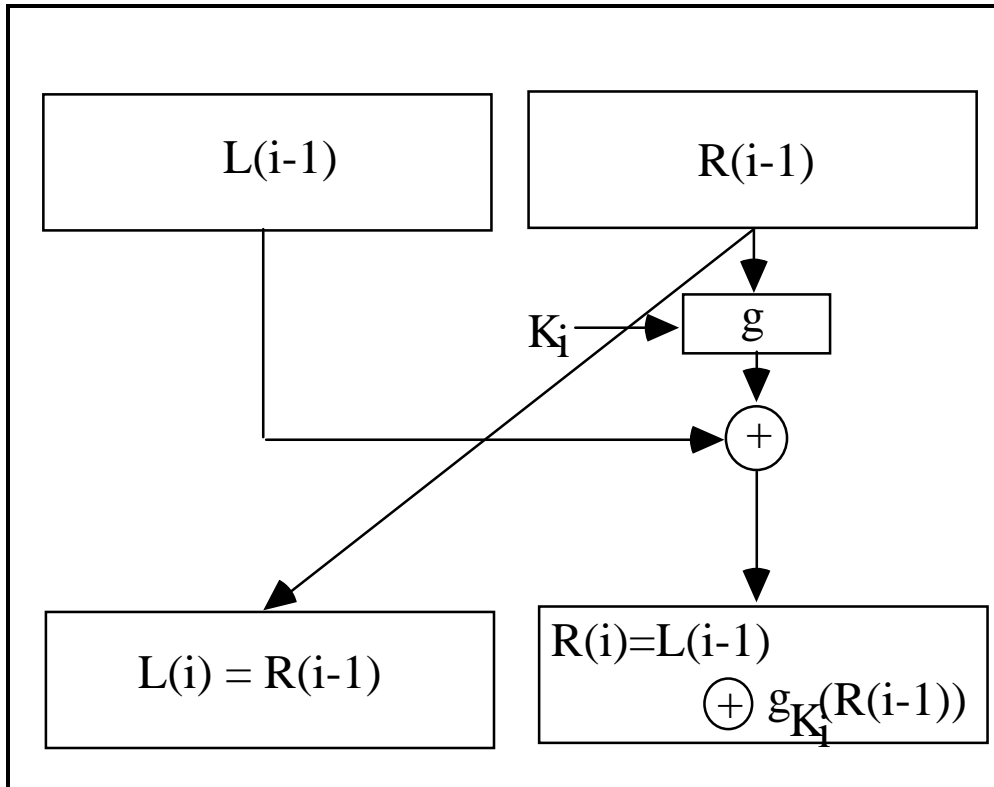
Appliquent des transpositions  
substitutions sur des blocs de 2 x 32 bits

- 1 étage amont, 2 en aval sont des transpositions simples fixes
- 16 étages intermédiaires dépendent de la clé de façon complexe.

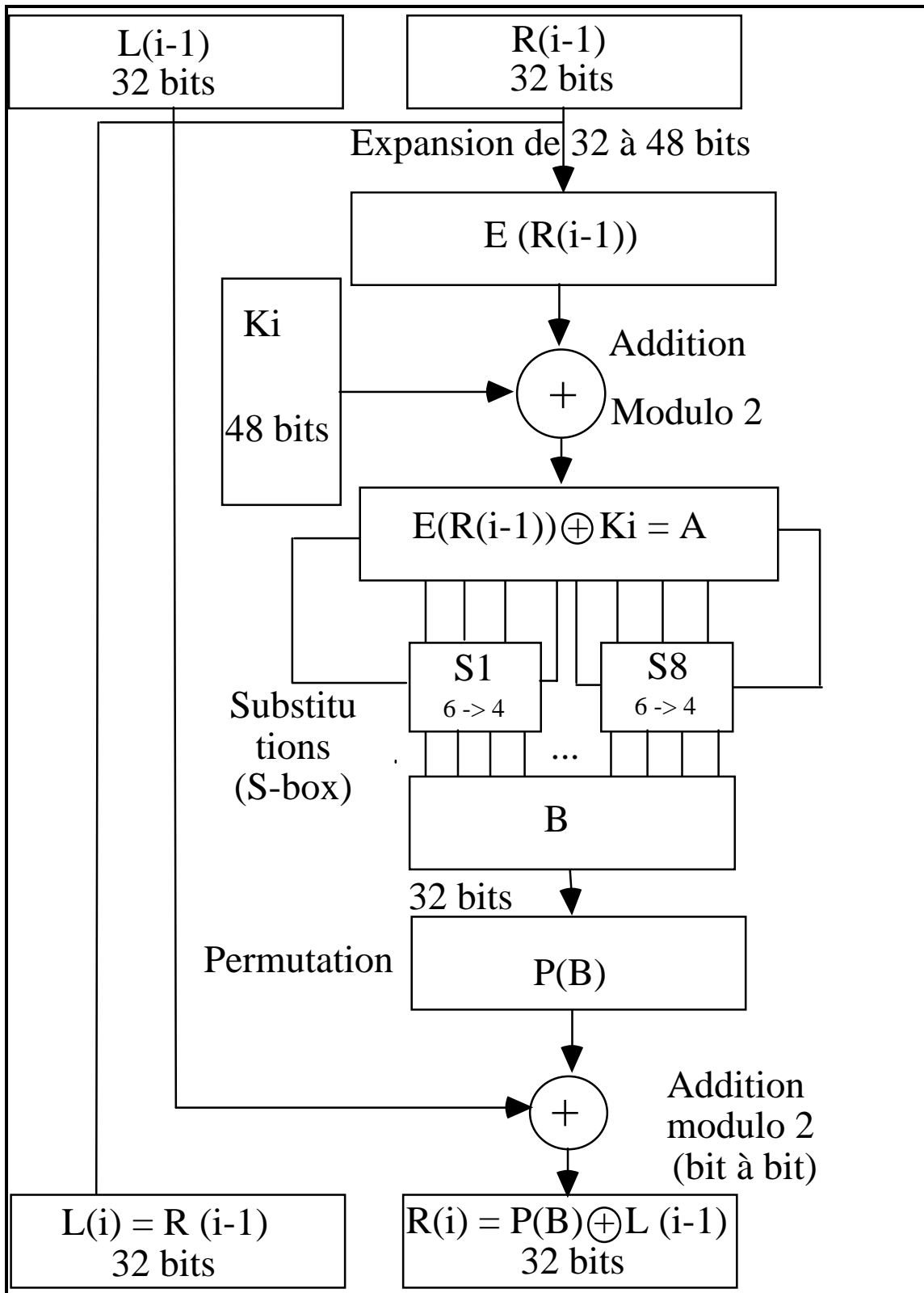
# Architecture générale du DES



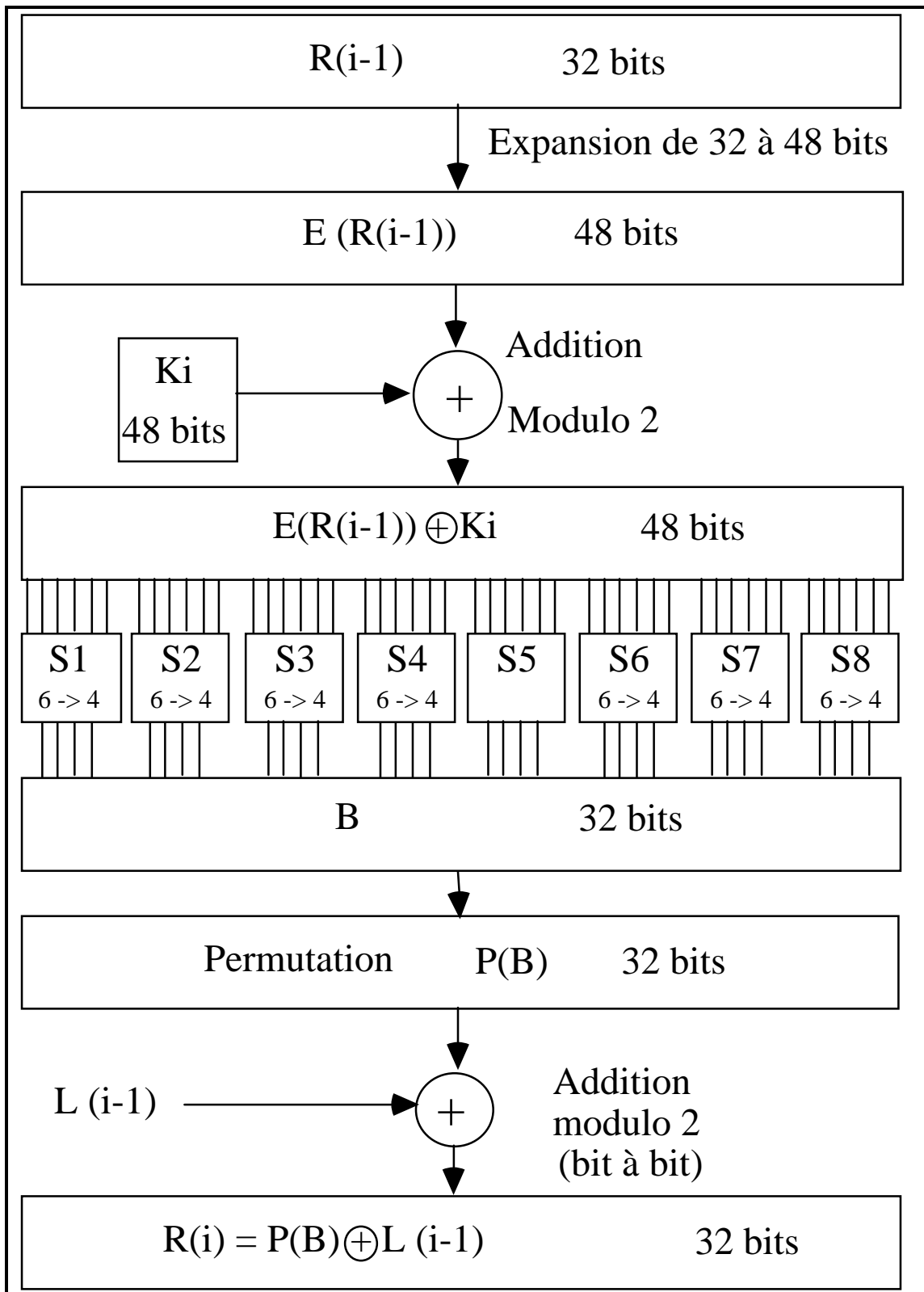
## Principe de réalisation d'un étage



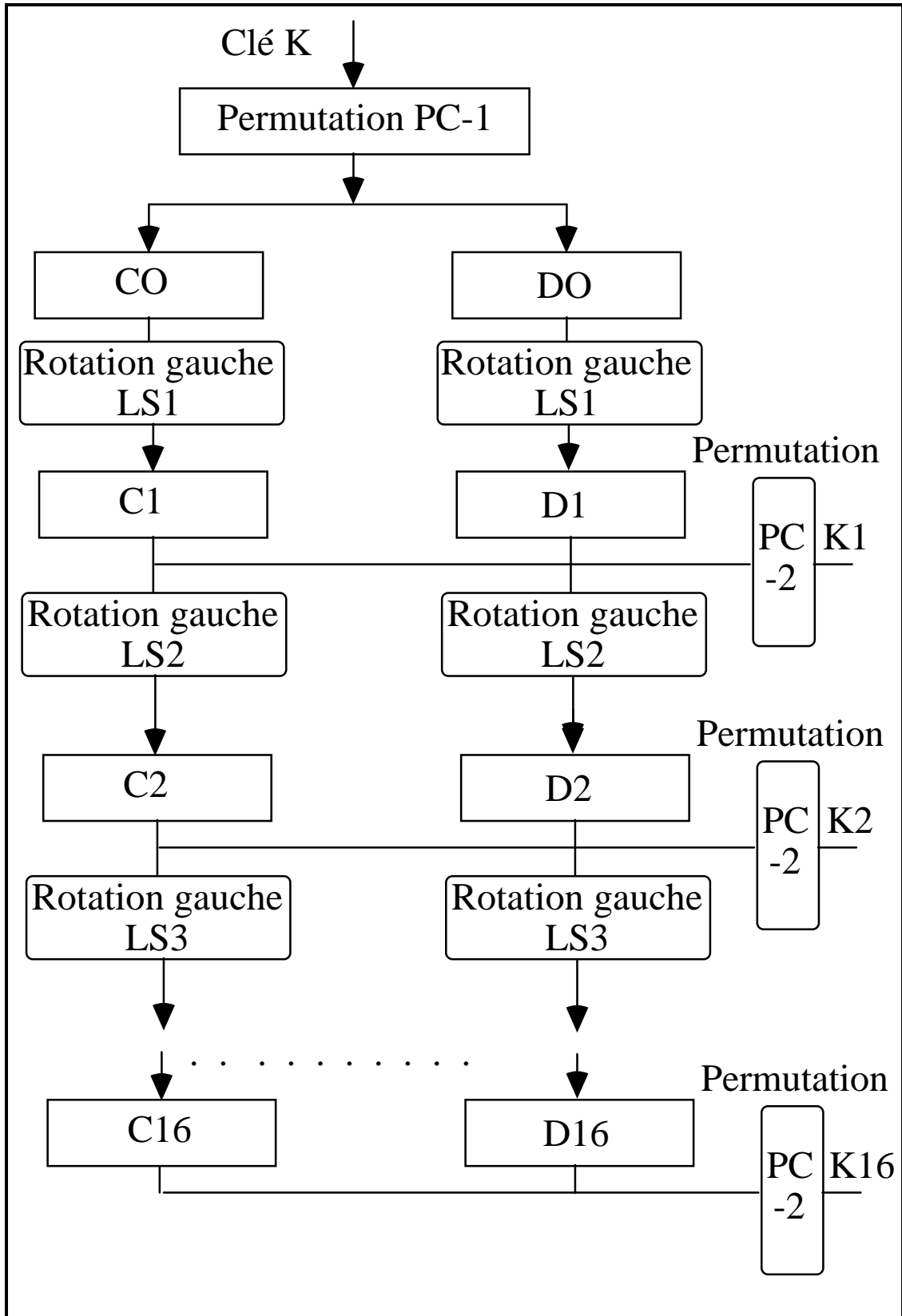
## Détails de la fonction principale d'un étage



## Détail des boîtes de substitution



# Méthode de calcul des clés



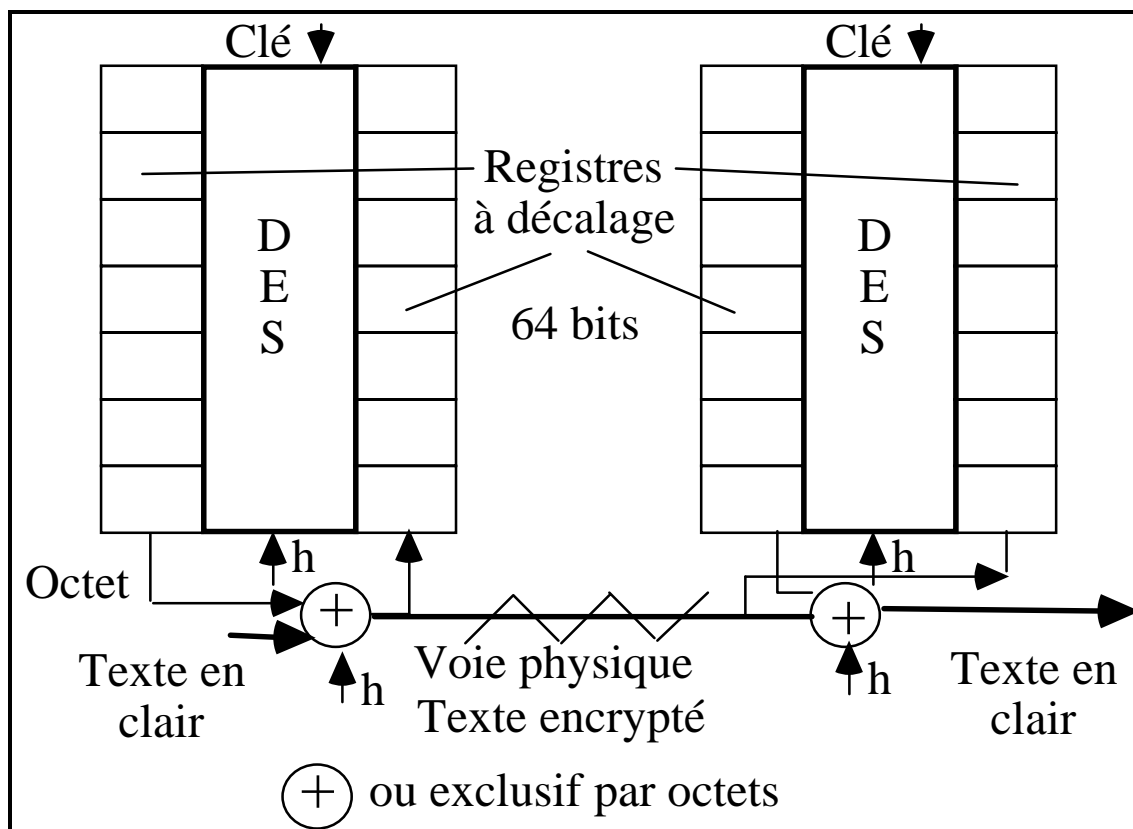


## Complément sur le calcul des clés intermédiaires

- La clé initiale  $K$  est sur 64 bits.
- La permutation PC-1 enlève les bits de parité et opère sur les 56 bits restants.
- On divise le résultat en deux moitiés  $C_0$  et  $D_0$  de 28 bits.
- On génère une suite  $C_i, D_i$  en opérant des décalages à gauche successifs:
  - $C_i = L_{S_i}(C_{i-1})$
  - $D_i = L_{S_i}(D_{i-1})$
- Pour obtenir la clé  $K_i$  on regroupe  $C_i$  et  $D_i$  et l'on opère sur les 56 bits une permutation PC-2

$$K_i = PC-2(C_i D_i)$$

## DES Utilisation A la Volée



- Un circuit DES de cryptage par blocs de 64 bits est utilisé octets par octets au moyen de registre à décalage (octets) d'entrée et de sortie.

- Performances Excellentes - cryptage à la volée à débits potentiellement très élevés (dizaine/ centaine de Mégabits/seconde).

- Utilisation multiples

Transmission de données informatiques

Cryptage de chaînes de télévision à péage.

## **Controverse sur la sécurité du DES**

### **Problème de longueur des clés**

- Initialement défini avec une clé de 112 bits le DES a été finalement doté par les autorités américaines d'une clé de 56 bits.

=> Le DES 56 est très probablement attaquable par des moyens informatiques plus ou moins lourds à la portée des états.

Des puces spéciales permettant l'essai de  $10^6$  clés par seconde ont été construites. Elles peuvent être organisées en processeurs spéciaux massivement parallèles.

### **Problème du choix des substitutions**

- Les principes de choix des S-box n'ont jamais été rendu public.

Officiellement elles sont conçues pour résister à une attaque particulière (la cryptanalyse différentielle).

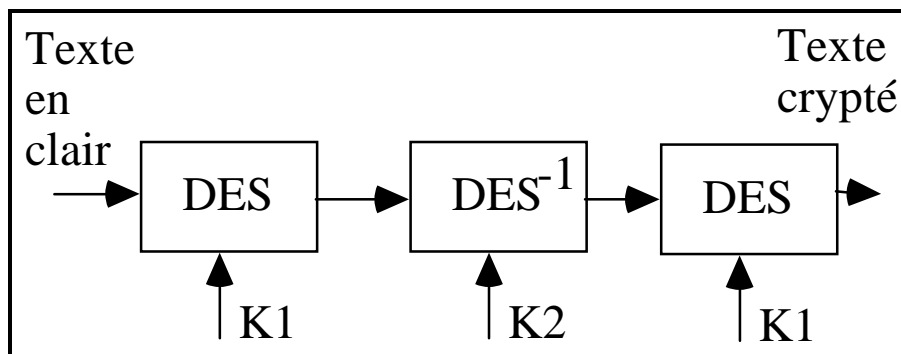
=> Personne n'a jamais rien trouvé concernant d'éventuelles propriétés cachées des boîtes de substitution.

# Amélioration de la sécurité du DES

## Utilisation de DES en cascade

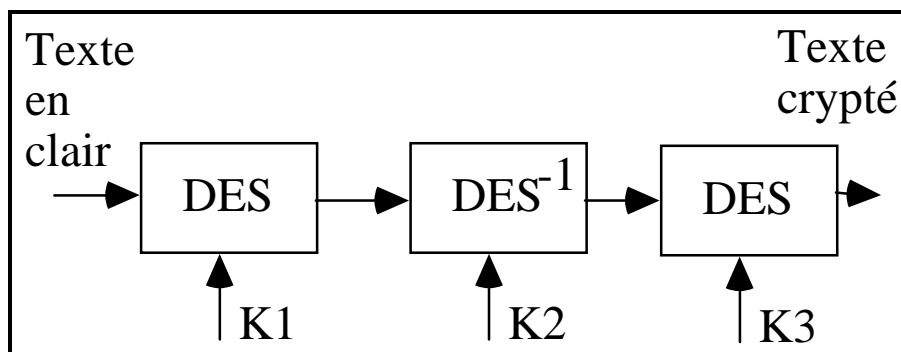
### Première proposition

Avec deux clés  $K_1$ ,  $K_2$  (128 bits).  
Moins bon qu'un DES 128 bits



### Seconde proposition

Avec trois clés  $K_1$ ,  $K_2$ ,  $K_3$ .



## Conclusion

### - DES -

- Standard maintenant assez ancien ayant finalement bien tenu.
- Excellentes performances en vitesse de cryptage.
  - Un circuit dédié crypte à 1 Gigabit/s
  - En logiciel on crypte à 1 Mégabit/s
- Niveau de sécurité pour une solution à clés privées très correct pour des applications ne nécessitant pas une confidentialité de haut niveau (militaire).

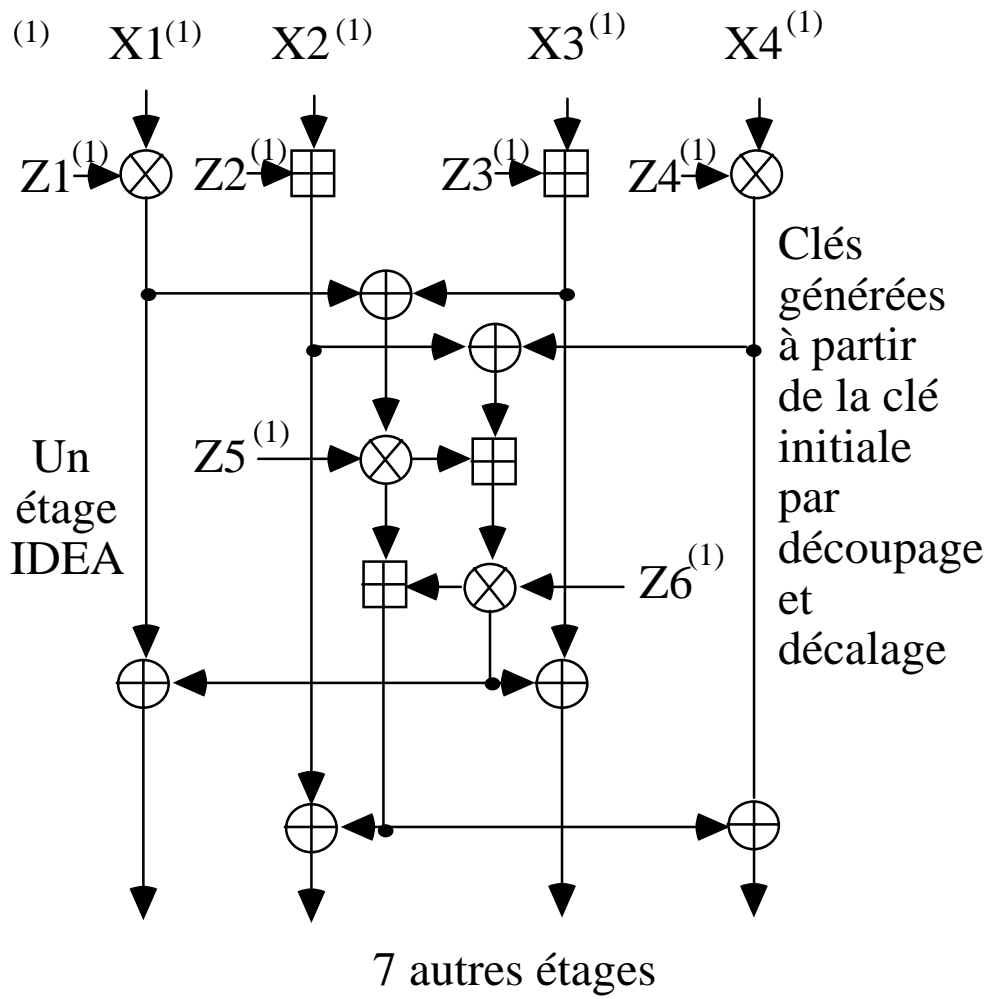
**Le DES 56 est probablement peu sûr pour un attaquant ayant de gros moyens mais performant et trop coûteux à casser pour des applications habituelles.**

## IDEA: International Data Encryption Algorithm

Autre solution de chiffrement par blocs de 64 bits basé sur huit étages facilement réalisable en matériel ou en logiciel.

Les opérations utilisées sont des opérations arithmétiques:

- ou exclusif  $\oplus$
- addition modulo 216  $\boxplus$
- multiplication modulo  $216 + 1$   $\otimes$



## Conclusion IDEA

- IDEA est considéré par les spécialistes comme l'un des meilleurs cryptosystème à clé privée.
- La longueur de clé est élevée (128 bits).
- La vitesse de chiffrement et de déchiffrement peut-être élevée au moyen de circuits spéciaux.
  - Circuits à 55 Mb/s et 177 Mb/s
  - En logiciel sur 386 33Mhz: 880 Kb/s
- Les attaques semblent difficile mais le système est assez récent (1990)



**Chapitre II**

**LA CRYPTOGRAPHIE  
A CLÉS PUBLIQUES**

**(Chiffres Asymétriques)**

## Introduction

Problèmes des systèmes à clés privées:

- **L'échange de clés** entre des sites qui n'ont jamais été en relation  
=> Il faut un moyen différent du réseau pour échanger des clés.

- Pour communiquer dans un groupe de  $n$  participants il faut  **$n(n-1)/2$  clés.**

### **La cryptographie à clés publique.**

1976 - Diffie et Hellman définissent les principes d'une nouvelle approche en cryptographie sans proposer de solution concrète au problème qu'ils posent.

### **La méthode RSA.**

1978 - Une solution pratique R. Rivest  
A. Shamir L. Adelman (la plus utilisée) :

## Cryptographie à clés publiques

On suppose que l'on sait trouver deux fonctions  $E_k$  et  $D_{k'}$  qui dépendent de clés  $k$  et  $k'$ .

$E_k$  est la méthode de chiffrement.

$D_{k'}$  est la méthode de déchiffrement.

1- **Définition de base:** le déchiffrement est l'inverse du chiffrement.

$$D_{k'} ( E_k (M) ) = M$$

2- Il est très très difficile **de déduire  $D_{k'}$  de la connaissance la méthode de chiffrement  $E_k$**  car cette fonction est diffusée à tous.

=> Des milliers d'années de calcul seraient nécessaires dans l'état des connaissances.

3- Idéalement  $E_k(M)$  et  $D_{k'}(M)$  devraient être **faciles à calculer**.

## Les clés publiques: une révolution dans l'approche cryptographique

Un utilisateur a un couple ( $E_k, D_{k'}$ )

- L'idée essentielle est que  $E_k$  (en fait  $k$ ) peut-être **rendue publique** par exemple dans un **annuaire** (le nom vient de là).

-  $D_{k'}$  est **privée** (en fait  $k'$  est privée et nécessairement différente de  $k$ ).

- **Tout le monde peut connaître  $E_k$**  et envoyer des messages secrets qu'**un seul destinataire** (celui qui connaît  $D_{k'}$ ) **peut comprendre**.

- D'où l'hypothèse fondamentale d'un tel système.

**On ne doit pas pouvoir trouver  $D_{k'}$  quand on connaît  $E_k$  et des messages chiffrés par  $E_k$**  (casser  $D_k$  en essayant des messages connus).

# L'Algorithme RSA

## Fonction E Encodage (publique)

- La clé publique comporte deux entiers:

$$\mathbf{k} = (\mathbf{e}, \mathbf{n})$$

- Le chiffrement se fait au moyen de l'élevation à la puissance e modulo n:

$$\mathbf{E}_k (\mathbf{M}) = \mathbf{M}^{\mathbf{e}} \pmod{\mathbf{n}}$$

## Fonction D Décodage (secrète)

- La clé secrète est un couple d'entiers:

$$\mathbf{k}' = (\mathbf{d}, \mathbf{n})$$

- Le décodage se fait au moyen de l'élevation à la puissance d modulo n:

$$\mathbf{D}_{k'} (\mathbf{M}) = \mathbf{M}^{\mathbf{d}} \pmod{\mathbf{n}}$$

**Remarque:** Les entiers n, e, d doivent être choisis selon des règles précise pour que ça marche.

# Détermination des clés

## 1. Détermination de n

Trouver **deux entiers premiers** p et q très grands.

La sécurité du système repose sur la difficulté de factoriser un grand entier n en deux entiers premiers p et q (la taille de n : 320 bits, 512 bits, 1024 bits conditionne également la vitesse des algorithmes).

$$\text{Calculer } n = p \cdot q$$

## 2. Détermination de e

Calculer  $z = (p-1)(q-1)$

Choisir un entier **e premier avec z**.

**La clé publique est ( e , n )**

## 3. Détermination de d

Choisir un entier d tel que :

$$e d \equiv 1 \pmod{z}$$

(d inverse de e dans l'arithmétique mod z)

**La clé privée est ( d , n )**

# Inversibilité de RSA

## Fonction d'Euler

Pour  $n$  entier  $\phi(n)$  (fonction d'Euler) est le nombre d'entiers premiers avec  $n$ .

Exemples connus:

- Si  $n$  est premier  $\phi(n) = n-1$
- Si  $n = p \cdot q$  avec  $p$  et  $q$  premiers  
 $\phi(n) = (p-1) \cdot (q-1)$

## Théorème d'Euler

Si  $a$  et  $n$  sont premiers entre eux

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

## Pourquoi RSA marche

$$\begin{aligned} D(E(M)) &= ((M)^e \pmod{n})^d \pmod{n} \\ &= (M^e)^d \pmod{n} = M^{e \cdot d} \pmod{n} \end{aligned}$$

On a choisi  $e \cdot d \equiv 1 \pmod{\phi(n)}$

Supposons  $e \cdot d = j \phi(n) + 1$

$$M^{e \cdot d} = M^{j \phi(n) + 1} \equiv M \pmod{n}$$

Parceque théorème d'Euler:

$$M^{j \phi(n)} \equiv (M^{\phi(n)})^j \equiv (1)^j \equiv 1 \pmod{n}$$

## Remarques

1. Le RSA doit toujours être appliqué à des blocs de chiffres d'amplitude inférieure à  $n$  pour faire des calculs modulo  $n$ .

=> Décomposition des messages en blocs

2. On voit ici que l'on a aussi la propriété de commutativité des opérations de chiffrement et de déchiffrement:

$$D ( E ( M ) ) = E ( D ( M ) ) = M$$

$$(M^e)^d = (M^d)^e = M^{ed} \pmod{n}$$



## Exemple: B. Schneier

1 Deux entiers premiers  $p = 47$ ,  $q = 71$

$$n = p \cdot q = 3337$$

2  $z = (p-1) \cdot (q-1) = 46 \cdot 70 = 3220$

3 Choix de  $e$

Exemple  $e = 79$  (premier avec  $n$ )

4 Calcul de  $d$  (inverse de  $e$  modulo  $z$ )

Une solution possible: le théorème d'Euler

$$e \cdot d = e \cdot e^{-1} = 1 = e^{\phi(n)} = e \cdot e^{\phi(n)-1} \pmod{z}$$

$$\text{Donc } d = e^{-1} = e^{\phi(n)-1} \pmod{z}$$

Numériquement  $79^{3219} \pmod{3220} = 1019$

Une autre solution plus simple:

L'algorithme d'Euclide

5 Chiffrer le message  $M$

$$M = 6882326879666683$$

Décomposition en blocs de taille inférieure

à  $n = 3337 \Rightarrow$  Des blocs de 3 chiffres

$$M = 688 \ 232 \ 687 \ 966 \ 668 \ 3$$

Chiffrer 688:  $688^{79} \pmod{3337} = 1570$

$$E(M) = 1570 \ 2756 \ 2091 \ 2276 \ 2423 \ 158$$

6 Déchiffrer en commençant par le premier bloc soit 1570:

$$1570^{1019} \pmod{3337} = 688$$

# Algorithmique de RSA

## Calcul des puissances modulo n

**Chiffrer M**  
 **$C = M^e \pmod{n}$**

Soit la décomposition de e sur k bits

$$e = \sum e(i) 2^i$$

début

  C := 1

  pour i := k-1 à 0 faire

    C := C<sup>2</sup> (mod n) ;

    si ( e(i) = 1 ) alors

      C := C . M (mod n) ;

    finsi;

  finpour;

fin;

Même algorithme pour déchiffrer.

## Algorithmique RSA: Détermination de grands nombres premiers Méthode de Soloway-Strassen

Choisir aléatoirement un grand entier  $p$ .  
Tester si ce nombre est premier.

**début**

$i := 0;$

test := VRAI ;

**tant que** (  $i \leq m$  et test = VRAI ) **faire**

$i := i + 1 ;$

choisir aléatoirement  $a$  (  $1 < a < p$  ) ;

calculer  $j = a^{(p-1)/2} \bmod p$  ;

calculer  $J(a,p)$  symbole de Jacobi;

**si** (  $j \neq J(a,p)$  ) **alors**

test := FAUX;

**finsi** ;

**fin tant que;**

**fin**

Si test = VRAI la probabilité que  $p$  soit décomposable est inférieure à  $2^{-m}$  .

En fait à chaque étape si l'on passe le test on a 50% de chances de décider que le nombre est décomposable et 50% de chances de ne pas pouvoir décider.

## Intuitions relatives au RSA

Chiffrer = bousculer les informations pour en rendre le sens inaccessible.

RSA = l'utilisation de l'élévation à la puissance puis d'une congruence.

- L'élévation à une puissance permet de changer le registre des entiers choisis

Exemple très simple  $e = 3$  et  $n = 41$ :

Pour  $M = 27$ ,  $M' = 28$  peu différents.

$$M^3 = 27^3 = 19683$$

$$M'^3 = 28^3 = 21952$$

- Les congruences introduisent des discontinuités  $\Rightarrow$  il est très difficile de trouver le logarithme d'un nombre dans un ensemble d'entiers modulo  $n$ .

$$E(M) = 27^3 \bmod (41) = 19683 \bmod (41)$$

$$E(M) = 480 * 41 + 3 \pmod{41}$$

$$E(M) = 3$$

$$E(M') = 28^3 \bmod (41) = 21952 \bmod (41)$$

$$E(M') = 535 * 41 + 17 \pmod{41}$$

$$E(M') = 17$$

# Attaque du RSA

## Solution de base

-  $n$  étant public le cryptanalyste cherche à trouver  $p$  et  $q$  pour calculer  $z$ .

=> Il doit factoriser un grand nombre en deux facteurs premiers.

Ce problème est **complexe**

Meilleurs algorithmes connus

- En 1989 avec 400 Vax pendant 3 semaines factorisation d'un nombre de 106 chiffres (352 bits)

- Actuellement factorisation possible de nombres de 110 à 120 chiffres (350 à 400 bits)

- Si on a trouvé  $p$ ,  $q$  et on connaît  $e$  alors utiliser l'algorithme d'Euclide pour trouver  $d$  tel que  $ed = 1 \pmod{(p-1)(q-1)}$

**D'autres attaques sont à découvrir...**

## Sécurité et performances du RSA

### Utiliser des longueurs de clés de plus en plus importantes

Valeurs utilisées: 512 bits, 640 bits  
1024 bits est considéré comme assez sûr pour plusieurs années.  
2048 bits pour une grande sécurité.

### Utiliser des circuits intégrés de cryptage de plus en plus performants

Actuellement une dizaine de circuits disponibles.

Vitesse de cryptage de base pour 512 bits: de 10 à 30 Kb/s

Évolution en cours de l'ordre de 64 Kb/s

A venir de l'ordre de 1 Mb/s

**Remarque:** Compte tenu de la complexité des traitements le DES doit être environ toujours 100 fois plus rapide que le RSA.

## Problèmes du RSA

- Trouver de grands nombres premiers (on prend en fait des nombres premiers en probabilité).
- Choisir des clés secrètes et publiques assez longues.
- Réaliser les opérations modulo  $n$  rapidement.

## RSA carte bancaire

Limitation des calculs du fait de la puissance de calcul disponible.

$n$  sur 320 bits (de l'ordre de 95 chiffres)

Même clé publique pour tout le monde  
Exemple de valeur 3

## Conclusion RSA

### Problème principal

Complexité algorithmique de la méthode.

### Solution pour l'utilisation de RSA.

Utiliser le RSA brièvement au début d'un échange pour échanger des clés secrètes de session d'un algorithme efficace à clés privées.

### Efficacité en sécurité de RSA

La méthode est officiellement sûre si l'on respecte certaines contraintes de longueur de clés et d'usage.

**Personne depuis 2500 ans n'a trouvé une solution rapide au problème de factorisation ...**



## **Chapitre III**

### **Les fonctions à sens unique et les fonctions de hachage**

## **Notion de fonction à sens unique ("One way function")**

**Fonction  $f(M)$  facile à calculer mais telle qu'il est extrêmement difficile de déduire  $M$  de  $f(M)$  (de calculer la fonction inverse).**

### **Exemple**

Calcul modulo  $n$  (dans un corps fini)

$M^2$  est facile à calculer

(plus généralement  $M^e$ ).

$\sqrt{M}$  est difficile à calculer

(plus généralement  $\log M$ ).

### **Utilisation**

Les fonctions à sens unique sont utiles pour garder sous forme inaccessible des mots de passe.

Par contre pour la confidentialité elles sont peu utiles car une fois  $M$  chiffré on ne sait pas déchiffrer  $M$ .

## Notion de fonction à sens unique à brèche secrète

**Fonction  $f(M)$  facile à calculer telle qu'il est extrêmement difficile de déduire  $M$  sauf si l'on connaît un secret  $K$ .**

Exemple pratique:

- Utilisation de RSA

$$n = p q$$

$$f(M) = M^e \bmod n$$

Le secret est constitué par la connaissance de  $p$ ,  $q$  et  $e$ .

## Fonctions de hachage

Une fonction de hachage est une fonction mathématique qui à partir d'un message (d'une donnée) génère une autre chaîne (généralement plus courte).

Terminologie: fonction de contraction, digest, empreinte digitale, "hash code"...

Exemples: Calcul de parité verticale.

On fait le ou exclusif de tous les octets d'une chaîne de caractères.

Calcul de code polynomial.

## Fonctions de hachage à sens unique sécuritaires

### **F à collision faible**

Il est difficile de créer un message  $M$  significatif tel que  $f(M) = K$

### **F à collision forte**

Il est difficile de trouver  $M$  et  $M'$  significatifs tels que  $f(M) = f(M')$ .

## **Notion de fonction de hachage à sens unique sans clé**

C'est une fonction de hachage à sens unique qui peut être calculée sans connaissance d'un secret (par n'importe qui).

Exemple type: MD4, MD5, SHA etc...

## **Notion de fonction de hachage à sens unique avec clé**

C'est une fonction de hachage à sens unique qui ne peut être calculée que par une entité détentrice de la clé.

Nombreux exemples de fonctions de hachage à sens unique avec clé déduites de méthodes de cryptographie.

## Signatures numériques

Une signature manuscrite idéale est réputée posséder les propriétés suivantes:

- La signature **ne peut-être imitée**.  
Elle prouve que le signataire a délibérément signé le document.
- La signature **authentifie** le signataire.  
Seul le signataire peut avoir signé.
- La signature appartient à un seul document (elle **n'est pas réutilisable**).
- Le document signé ne peut être partiellement ou totalement **modifié**.
- La signature ne peut-être **reniée**.

### Base de la signature numérique

L'existence d'une fonction de hachage à sens unique avec clé.

Une solution possible: une fonctions de hachage à sens unique et une technique classique de cryptographie (exemple le RSA)

**Exemple de fonction de hachage  
à sens unique  
MD5 "Message Digest" version 5**

- Un message est décomposé en blocs de 512 bits soient 16 sous-blocs  $M_j$  de 32 bits (le message est complété à un nombre entier de blocs).

- MD5 génère un hachage sur 128 bits.

- On calcule le résultat sous la forme de quatre mots de 32 bits initialisés au départ:

a = 01234567

b = 89ABCDEF

c = FEDCBA98

d = 76543210

- Pour chaque bloc de 512 bits on réalise un nouveau calcul sur le bloc à partir de la dernière valeur a b c d.

## L'algorithme MD5

- Quatre fonctions F, G, H, I non linéaires sont données par:

$$F(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$

$$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$

$$H(X, Y, Z) = (X \oplus Y \oplus Z)$$

$$I(X, Y, Z) = Y \oplus (X \vee \neg Z)$$

- Le calcul comprend 4 rondes de 16 applications successives des fonctions FF, GG, HH, II qui dépendent des fonctions F, G, H, I, des sous-blocs  $M_j$ , des variables a, b, c, d, et de constantes  $t_i$ :

$$FF(a, b, c, d, M_j, s, t_i) \longrightarrow a = b + ((a = F(b, c, d) + M_j + t_i) \triangleleft s)$$

$$GG(a, b, c, d, M_j, s, t_i) \longrightarrow a = b + ((a = G(b, c, d) + M_j + t_i) \triangleleft s)$$

$$HH(a, b, c, d, M_j, s, t_i) \longrightarrow a = b + ((a = H(b, c, d) + M_j + t_i) \triangleleft s)$$

$$II(a, b, c, d, M_j, s, t_i) \longrightarrow a = b + ((a = I(b, c, d) + M_j + t_i) \triangleleft s)$$

$\triangleleft s$  est le décalage à gauche de s positions.



## L'algorithme MD5 (suite)

- La première ronde:

FF (a,b,c,d, M(0),7, D76AA478)

FF (d,a,b,c, M(1),12, E8C7B756)

FF (c,d,a,b, M(2),17, 242070DB)

FF (b,c,d,a, M(3),22, C1BDCEEE)

FF (a,b,c,d, M(4),7, F757C0FAF)

FF (d,a,b,c, M(5),12, 4787C62A)

FF (c,d,a,b, M(6),17, A8304613)

FF (b,c,d,a, M(7),22, FD469501)

FF (a,b,c,d, M(8),7, 698098D8)

FF (d,a,b,c, M(9),12, 8B44F7AF)

FF (c,d,a,b, M(10),17, FFFF5BB1)

FF (b,c,d,a, M(11),22, 895CD7BE)

FF (a,b,c,d, M(12),7, 6B901122)

FF (d,a,b,c, M(13),12, FD987193)

FF (c,d,a,b, M(14),17, A679438E)

FF (b,c,d,a, M(15),22, 49B40821)

Quatre rondes analogues:

FF, GG, HH, II.

## **Autres exemples de fonctions de hachage à sens unique**

MD5 présenterait quelques problèmes  
relativement aux collisions fortes.

### **SHA : "Secure Hash Algorithm"**

- Autre amélioration du MD4.
- Proposé comme standard fédéral  
aux USA.
- Empreintes sur 160 bits plus  
longues que celles de MD5 128 bits.

Considéré comme l'une des meilleures  
solutions actuelles

## **Bibliographie**

A.S. Tannenbaum - Computer Networks.  
Prentice Hall

B. Schneier - Cryptographie appliquée.  
Thomson Publishing International France

D.E. Denning - Cryptography and data  
security. Addison Wesley 1982

# **La sécurité informatique**

**G Florin, S Natkin**

# Introduction

## Définition de base de la sécurité

L'ensemble des moyens mis en oeuvre pour minimiser la vulnérabilité d'un système contre des menaces intentionnelles.

*Domaine d'élection :*

les systèmes informatiques réalisant des traitements sensibles ou comprenant des données sensibles

## Minimisation des risques

Différences entre une approche financière et une approche d'intolérance au risque => pour tout risque évaluation du coût du risque et du coût de sa protection.

## Évolution de la problématique

Passage d'un problème "militaire" à un problème de relations contractuelles "économiques" (signature, notariation) relevant de la méfiance et de la preuve juridique autant que de la protection.

# Politique de sécurité

## Définition

Une politique de sécurité est un ensemble de règles qui fixent les actions autorisées et interdites dans le domaine de la sécurité.

## Étapes types dans l'établissement d'une politique de sécurité

### *- Identification des vulnérabilités*

En mode fonctionnement normal

En cas de défaillances

Un système fragilisé est en général plus vulnérable : c'est dans un de ces moments intermédiaires qu'une intrusion peut réussir.

*- Évaluation des probabilités associées à chacune des menaces.*

*- Évaluation du coût d'une intrusion réussie.*

*- Choix des contre mesures.*

*- Évaluation des coûts des contre mesure.*

*- Décision.*

# Obligations légales en France

## Validité juridique d'opérations informatiques

Certaines transactions informatiques entraînent des obligations légales de responsabilité => Elles sont considérées comme valides juridiquement par la loi ou la jurisprudence.

Exemples:

- Ordre de virement (deux fois le même ordre: le paiement doit-être honoré).
- Commandes télexées

# Loi informatique et liberté

## Loi 78\_17 du 6/1/1978

Définit la constitution et le rôle de la **CNIL** ("Commission Nationale Informatique et Liberté").

Une entreprise ou une administration qui traite des fichiers administratifs nominatifs est **responsable** relativement à la **non divulgation** des informations qu'elle gère.

. Nécessité de **formalités préalables** à la mise en oeuvre des traitements automatisés pour:

. La **collecte, l'enregistrement, la conservation** des données nominatives

. **L'exercice du droit d'accès.**

. Dispositions **pénales** de non respect.



**Loi sur les copies**  
**Loi no 85-660 du 3/7/1985**

**Décrit les règles relatives aux  
contrefaçons et au droit d'auteur**

Par exemple **la copie** (autre que pour sauvegarde) est punissable de trois mois à deux ans de prison , d'une amende de 6000 à 12000 Francs.

**Loi relative à la fraude informatique**  
**Loi no 88-19 du 5/1/1988**

Sont passibles de sanctions pénales pouvant atteindre 5 ans de prison, une amende de 2 millions les faits suivants:

- . Accès frauduleux aux données.
- . Introduction de données.
- . Entrave au fonctionnement.
- . Falsification de données informatisées.

## Lois relatives à l'usage de la cryptographie

En France l'usage des moyens de chiffrement est **limité** (était très limité jusqu'à une époque récente):

- **Déclaration obligatoire** concernant l'utilisation de méthodes de chiffrement en authentification et en intégrité.
- **Demande d'autorisation** pour la confidentialité.
- **Limitation** des domaines (professions) ou la cryptographie est autorisée.

Service administratif compétent:SCSSI  
Service Central de Sécurité des Systèmes Informatiques .

Assouplissement de la réglementation vers une usage analogue à celui de pays comparables.

- Allongement des clés autorisées.
- Extension des méthodes utilisées.
- Extension des domaines d'application

# Plan du cours

## **Chapitre I** **Problèmes de la sécurité informatique**

### **I . 1 Notions générales**

### **I . 2 Techniques centralisées de la sécurité informatique.**

## **Chapitre II** **Protocoles de sécurité**

### **II . 1 Confidentialité.**

### **II . 2 Intégrité.**

### **II . 3 Authentification.**

### **II . 4 Non répudiation.**

**Chapitre I**

**Problèmes de la sécurité  
informatique**

## **I . 1 Notions générales**

### **Notion de politique de sécurité**

- Consiste à garantir toutes les opérations sur les objets informatiques (ressources),
- Ne sont réalisées que par les entités (physique ou informatique) habilitées.

### **Approches de l'univers centralisé**

L'authentification des usagers

La protection

### **Approches de l'univers réparti**

Les protocoles de sécurité.

## **Principales propriétés de sécurité**

### **Authentification**

Seules les entités autorisées peuvent agir dans un système (**protège de l'usurpation d'identité**).

#### **Entités à authentifier:**

- un usager - un processus en exécution
- une machine dans un réseau

### **Confidentialité des données**

Seuls les utilisateurs habilités dans des conditions prédéfinies, **ont accès en lecture** aux informations.

### **Intégrité des données**

Une information n'est modifiée (**accès en écriture**) que dans des conditions pré définies (selon des contraintes précises)

#### **Exemples :**

- Pas de modifications par des usagers non autorisés.
- Une modification intempestive (même très temporaire) est interdite.

## Non répudiation

C'est la propriété qui assure que **l'auteur d'une action ne peut** dénier l'avoir effectué. Deux aspects de la non répudiation dans une transaction électronique

### *a) La preuve d'origine*

Un message (une transaction) ne peut être dénié par son émetteur.

### *b) La preuve de réception*

Un récepteur ne peut ultérieurement dénier avoir reçu un ordre s'il ne lui a pas plu de l'exécuter alors qu'il le devait juridiquement (exécution d'ordre boursier, de commande, ...)

## L'Auditabilité

Capacité de détecter et d'enregistrer de façon infalsifiable les tentatives de violation de la politique de sécurité.

## **Notion de signature** (au sens habituel)

- La signature **ne peut-être imitée.**  
**Le signataire est le seul** à pouvoir signer (caractérisation psychomotrice).

### **Signature = Authentification**

- Le document signé ne peut être partiellement ou totalement **modifié.**

### **Signature = Intégrité**

- La signature ne peut-être **reniée.**

### **Signature = Non répudiation**

- La signature appartient à un seul document (**elle n'est pas réutilisable**).



# Les principes généraux de la réalisation de la sécurité

## Le confinement

L'ensemble des objets est maintenu dans des domaines (de protection) étanches pour empêcher qu'une entité n'interfère avec une autre à la suite d'une erreur involontaire ou volontaire.

Exemple accès à l'espace mémoire d'un autre usager

## Le principe du moindre privilège

Pour qu'un système fonctionne en sécurité il faut donner à ses utilisateurs exactement les droits dont il ont besoin pour s'exécuter : **ni plus ni moins**.

Si l'on donne plus de droits on risque de voir ces droits utilisés anormalement

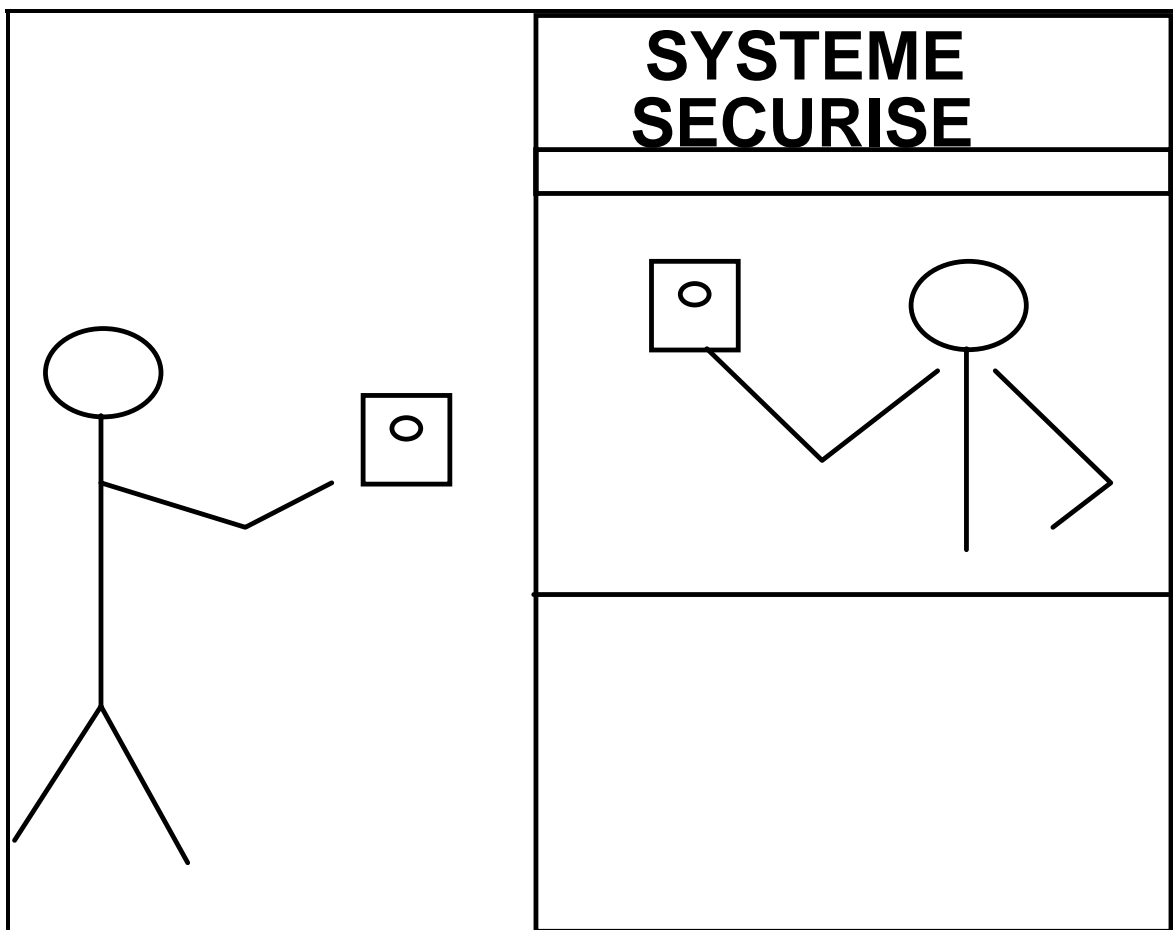
- Soit involontairement
- Soit volontairement

## Notion de guichet

Tout accès à un objet se fait via “un guichet” : point de passage obligé qui permet la vérification des droits.

Une entité se présente au guichet:

- **Authentication** de l'entité
- **Authentication** du guichet  
(risque de déguisement)
- Présentation d'un certificat montrant les **droits** pour réaliser l'opération.



## **Approfondissement: Construction des guichets**

- Pouvoir **protéger** des données secrètes qui constituent la base de l'authentification ou qui doivent être étanches en lecture.
  
- Pouvoir **protéger** en écriture certaines données accessibles en lecture ou en exécution (codes exécutables).
  
- Pouvoir **authentifier** client et guichet.
  
- Pouvoir **garantir** que l'exécution de l'opération ne peut être faite que par le guichetier selon sa spécification.
- Pouvoir **garantir** que les transferts de données entre le client sont protégés en écriture ou lecture et écriture (intégrité ou confidentialité).
- Pouvoir **enregistrer** de façon non falsifiable toutes les opérations (journalisation => non répudiation).
- Pouvoir **noter** toutes les tentatives de fraude (auditabilité).

## **Approfondissement: Administration des guichets**

- Gestion des **entités** (sujets, objets) et des données d'authentification (désignation, création, destruction).

- Gestion des **guichets** incontournables et des données d'authentification de ces guichets pour chaque opération (désignation, création, destruction).

- Gestion des **droits** (désignation, création, destruction, propagation) .

## Modèles du contrôle d'accès

### Sujets et objets

**Sujets:** utilisateurs, processus, groupes d'utilisateurs ou de processus.

**Objets :** segments ou pages mémoires, fichiers, processus, programmes.

Z Dans les applications orientées objet

**Sujets:** utilisateurs ou objets.

**Objets:** objets.

### La matrice de contrôle d'accès

Définit à chaque instant les droits de chaque sujet sur chaque objet.

	<b>Seg M1</b>	<b>Seg M2</b>	<b>Fich F1</b>	<b>Fich F2</b>
<b>P1</b>	<b>R,W,E</b>		<b>R,W</b>	<b>R</b>
<b>P2</b>		<b>R,W,E</b>		<b>R,W</b>
<b>S1</b>	<b>R,W,E, Créer, Détruire</b>	<b>R,W,E, Créer, Détruire</b>	<b>R,W,E, Proprié taire</b>	<b>R,W,E, Proprié taire</b>

	<b>Segment M1</b>				
	<b>R</b>	<b>W</b>	<b>E</b>	<b>Créer</b>	<b>Détruire</b>
<b>P1</b>	<b>1</b>	<b>1</b>	<b>1</b>		
<b>P2</b>					
<b>S1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>

## Évolution de la matrice des droits

La matrice des droits évolue en fonction des événements suivants:

- **Création et destruction des sujets et des objets.**
- **Création et destruction des droits.**
- **Propagation des droits.**

### Exemple

. Dans le systèmes VMS les droits d'un processus fils sont hérités du père et partagés entre les fils.

. Pour le processus racine ce sont les droits de l'utilisateur créateur.

## Modèle de politique de sécurité

### Définition formelle

- Des propriétés de sécurité qui doivent être vérifiées à chaque instant (invariants de sécurité) par la matrice de contrôle des droits.
  
- Des règles qui gèrent les évolutions de cette matrice

### Il est ensuite possible:

- De vérifier que les évolutions préservent les invariants.
  
- De vérifier que l'implantation n'autorise que les transitions valides.
  
- Que le contrôle d'accès est toujours exécuté.

## Deux classes de politiques

### Politiques discrétionnaires

Une politique est **discrétionnaire** si l'entité qui possède un objet ("owner", propriétaire) à tous les droits pour propager les droits sur cet objet.

Exemple: Droits d'accès aux fichiers UNIX.

### Politiques obligatoires

La politique est dite **obligatoire** si ce processus de propagation est défini par un ensemble de règles.



## Exemple de politique obligatoire: systèmes militaires

- A tout sujet est associé **un ensemble de couples (h,d)** qui pour chaque domaine d définit le niveau d'habilitation du sujet.

. **Niveaux d'habilitations** ordonnés

$H = \{\text{non classifié, confidentiel, secret, très secret}\}$

. **Domaines de compétences**

$DOM = \{\text{nucléaire, sécurité ..}\}$

(Définis par un thesaurus ordonné par des relations d'inclusions).

. **Relation d'ordre** notée

$(h,d) \quad (h',d') \text{ si } h < h'$

**ou  $h=h'$  et  $d' \quad d$**

**(d est inclus au sens large dans d')**

- A tout droit sur un objet (exécution d'une opération) est associé **un couple**  $(h_o, d_o)$ .

- Les politiques de sécurité reposent sur des **préconditions** pour réaliser une opération par un sujet jouant sur les deux couples ( exemple:  $(h_o, d_o) \quad (h, d)$  ).

## Politique de Bell Lapadula

Objectif: éviter la **propagation** des données d'un domaine de sécurité vers un domaine inférieur.

.  $D(o,j)$  le couple  $(h,d)$  associé au droit  $j$  (Read, Write, Read+Write... ) sur l'objet  $o$  .  
 $A(s)$  l'ensemble des couples  $h,d$  du sujet  $s$

La matrice de contrôle  $M$  est telle que:

$$\begin{aligned} - M(s,o,Read) = \text{vrai} & \quad (h,d) \in A(s) \\ & \quad D(o,Read) \in (h,d) \end{aligned}$$

Un sujet ne peut lire un document que s'il est habilité

$$- M(s,o,Read) = \text{vrai} \text{ et}$$

$$\begin{aligned} M(s,o',Read+Write) = \text{vrai} \\ D(o') \in D(o) \end{aligned}$$

Pour tout couple de documents  $o$  et  $o'$  tels qu'un sujet ait juste le droit de lire  $o$  et le droit de lire et écrire  $o'$   $o'$  est moins confidentiel que  $o$  (domaine donné)

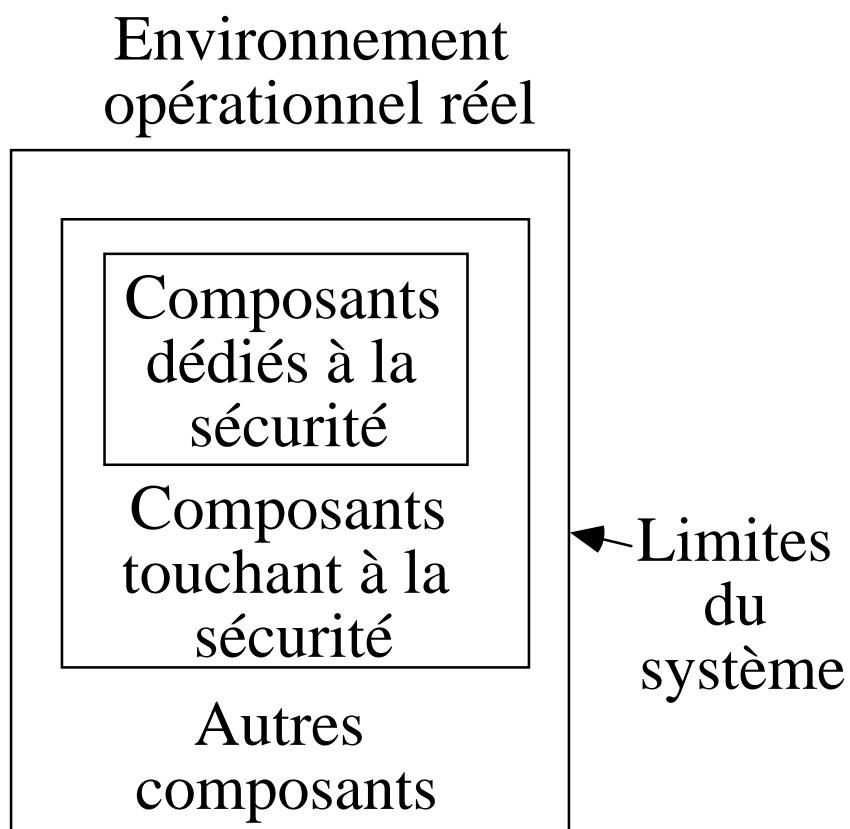
### **Problèmes de cette politique:**

Elle ne détermine pas le mode d'évolution des droits sur les objets.

Elle amène (si la granularité est insuffisante) à surclassifier des objets.

## Noyau de sécurité

**L'implantation des mécanismes de sécurité est basé sur une hiérarchie des fonctions (TCSEC Trusted Computer Security).**



## Le moniteur de référence

Les composants dédiés à la sécurité reposent sur **un moniteur de sécurité** qui assure le contrôle d'accès

Il doit être:

- **Inviolable,**
- **Incontournable,**
- **Correct**

Par rapport à un ensemble de propriétés permettant d'implanter des politiques de sécurité.

Le moniteur de référence est le "**méta guichet**" qui permet de gérer des guichets et les droits d'accès aux guichets

## **Construction du moniteur de référence**

**Un moniteur de référence est construit selon une hiérarchie de mécanismes dont l'étanchéité dépend du type d'attaque:**

Sur chaque machine on utilise **des mécanismes matériels** (adressage, exécution des certaines instructions).

Pour contourner ces mécanismes il faudrait **modifier le matériel.**

- **Gestion de la mémoire.**
- **Mode d'exécution des processus et contrôle d'accès aux instructions privilégiées.**
- **Programme en mémoire morte**
- **Système matériel d'authentification** (exemple: lecteur de carte à puces).

Dans un système réparti, la partie essentielle du moniteur de sécurité est composé **des protocoles de sécurité.**

## **I . 2**

# **Techniques centralisées de la sécurité informatique**

# Les mécanismes centralisés de protection

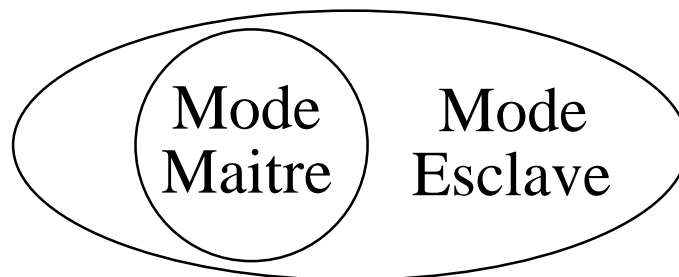
## Les machines à anneaux

- A tout instant un processus s'exécute dans **un niveau de protection** donné.

De même **un niveau** est associé à l'exécution de chaque **instruction** et à chaque **référence mémoire**.

- Les niveaux sont **hiérarchisés**: notion **d'anneaux de protection**.

Exemple: de 2 niveaux (mode maître esclave) à 8 niveaux.



- Toute opération (instruction, référence mémoire) implique **le contrôle du droit d'accès**:

pour qu'une opération soit exécutée le niveau d'appel doit être **supérieur** ou égal à celui de l'objet référencé sinon il y a **déroutement ('trap')**.

## Cas d'un déroutement ('trap')

Exécution d'une instruction ou d'un accès **avec augmentation des droits** délibéré ou accidentel.

### Déroutement

= **Demande de changement d'anneau**

Le déroutement provoque le passage à un guichet => l'exécution d'un code de contrôle du droit de passage dans l'anneau appelé.

### Exemple

Lecture fichier

Usager = 4 -> Gestion des fichiers = 3

Contrôle du droit sur le fichier (3),

Calcul des adresses physiques (3),

Appel à l'exécutif 3 -> 2,

Contrôle du niveau de l'appelant (2),

Exécution d'une écriture disque 2->1

(superviseur d'entrées sorties)

**Objectif réalisé:** certaines instructions (certains accès) ne sont **exécutables** que dans **l'anneau** (privilégié) qui le permet.



## Machines à domaines (machines à capacités)

Dans un tel système chaque objet s'exécute dans un "domaine de protection" défini pour un sujet par un ensemble de droits sur des objets.

Tout appel d'une opération est réalisé moyennant la fourniture d'une **capacité**:

- **référence** de l'appelé (désignation)
- **droit possédé** par l'appelant sur l'appelé (protection).

### **Rôle du moniteur de référence**

- La capacité est **contrôlée** avant d'exécuter la méthode .
  
- La **modification des capacités** (la manipulation, la propagation des droits) ne peut être réalisée que par appel du moniteur (fonctionnant en mode protégé).

## Implantation de la matrice de contrôle d'accès

Matrice des **capacités**.

Une approche de **capacité** stocke la **matrice** (sujet, objet) **par lignes**.

Une approche de **liste de contrôle d'accès** consiste à stocker la **matrice par colonnes**.

### 1) Machines à capacités

Pour chaque sujet on gère au niveau système (dans des segments de mémoire particuliers) **l'ensemble des capacités** de ce sujet (liste de capacités ou **c-liste**).

**Exemple:** Dans le système VAX-VMS les primitives du noyau sont protégées par de très nombreux droits d'accès qui sont spécifiques de chaque utilisateur:

- droit de créer des processus
- droit de faire des opérations réseaux
- droit de modifier ses droits....

A chaque fois qu'un sujet ouvre un objet ou à chaque fois qu'il accède à un objet

Selon la nature de l'objet et le niveau de protection souhaitée on vérifie le droit du sujet.

Exemple avec mécanisme matériel de gestion des segments de capacités : IBM AS 400.

## 2) Les listes de contrôle d'accès "ACL Access Control List"

On définit **pour chaque objet la liste des sujets et leurs droits** sur l'objet.

**Exemple** : Dans le système de fichier d'UNIX (et dans presque tous les systèmes de fichiers on a une approche analogue)

On associe à chaque fichier **une liste de droits** (lire, écrire, exécuter)

- a) Pour un usager particulier.  
(**le propriétaire** du fichier)
- b) Les **membres du groupe** du propriétaire.
- c) **Tous les usagers** du système.

Remarque: Dans de nombreux systèmes de fichiers les droits sont plus fins et l'on peut définir des listes de contrôle d'accès pour d'autres groupes ou pour chaque utilisateur (exemple VAX-VMS).

## Choix entre les deux approches

Repose sur le coté plus ou moins pratique de la manipulation des droits:

**1 Les listes de contrôle d'accès** sont plus pratiques pour la révocation des droits (puisque'ils sont associés à l'objet).

**2 Pour le stockage des droits et la rapidité de leur consultation** cela dépend des objets:

- Les listes de capacités (c-listes) sont **dans les descriptifs des sujets.**

Favorable pour ce qui concerne les droits des objets manipulés en mémoire centrale par le noyau (peu de sujets actifs)

- Les listes de contrôle d'accès **doivent être rangées avec les descriptifs des objets**

Favorable dans le cas des fichiers

**3 Pour la distribution:**

Les systèmes à capacité sont mieux adaptés à la distribution que les listes de contrôle d'accès.

## **Les techniques d'authentification des utilisateurs**

**L'authentification = vérification de l'identité.**

- Il est impossible d'assurer la confidentialité, la non répudiation .... sans s'appuyer sur la garantie indiscutable de l'identité de l'entité émettant une requête.

- L'authentification devrait être assurée en continu. Il ne suffit pas de réaliser l'authentification une fois pour toute à l'ouverture d'un objet (en début de session)

**. Quand l'entité est un usager:**

il peut quitter son poste en le laissant ouvert => procédure de déconnexion automatique, procédure de réautorisation périodique.

**. Quand l'entité est informatique:**

une substitution peut avoir lieu surtout en réseau => nécessité de protocoles de sécurité).

# Les moyens de l'authentification

## Trois méthodes

### 1 Ce que connaît l'utilisateur

La clé, le mot de passe, ....

Technique:     la plus simple  
                  la plus répandue

Applicable aux entités informatiques

Problèmes bien connus:

- le secret peut-être découvert
- le secret peut-être confié à un tiers

Quelques parades:

- **Obliger l'utilisateur à changer** régulièrement de mot de passe.
- **Surveiller les tentatives d'accès** illicite (les enregistrer et les éditer).
- **Prévenir l'utilisateur des connexions** sur son compte en affichant la date et l'heure (ex du dernier accès).

## **2 Ce que détient l'utilisateur**

### **Un secret matérialisé physiquement**

Une clé (traditionnelle), une carte (magnétique, à code barre, à puce)

Technique simple, répandue.

Les problèmes :

- la perte, le vol du support
- la duplication (plus ou moins facile mais toujours possible).

## **3 Ce qu'est l'utilisateur: les méthodes biométriques**

Une solution en rapide développement.

Qui peut-être très efficace (à la limite trop en cas de décès par exemple).

Le plus souvent onéreuse (dans l'état actuel des procédés).

Qui peut-être difficile à accepter dans certains cas par l'utilisateur.

## **Techniques biométriques Approfondissements**

1 - Issues d'un **caractère biologique (morphologique)** a priori caractérisant de manière unique l'utilisateur.

Le prélèvement peut ne pas suivre exactement un profil préenregistré (en raison même de son type).

2 - Issues de la **classification** d'un caractère d'un ensemble d'usagers (analyse statistique, réseaux de neurones).

Nécessité d'études approfondies du caractère utilisé.

- à **l'intérieur du groupe** autorisé.
- **dans une population quelconque** (variabilité du caractère retenu).

### **Incertitudes des techniques biométriques**

- La variabilité intra-individuelle.
- La variabilité inter-individuelle.

conduit à deux types d'erreurs possibles:

- . Le rejet à tort d'un individu autorisé.
- . L'acceptation à tort d'une personne.



## Quelques techniques biométriques utilisées ou à l'étude

### - *L'empreinte digitale*

Le sujet applique son doigt sur un prisme.

La pression déclenche une analyse par balayage d'un faisceau infrarouge.

Le signal reçu dépendant des sillons de l'empreinte (creux et bosses successives) est analysé et classifié).

### - *La vascularisation de la rétine*

Empreinte biométrique analogue à l'empreinte digitale qui est très stable.

L'image de fond de l'oeil est obtenue par un dispositif monoculaire utilisé dans les test de vision médicaux.

La numérisation est effectuée par une caméra infrarouge.

L'image est classifiée.

### - *La voix*

Le sujet prononce quelques mots.

Le système numérise et classifie le signal.

### ***- La géométrie de la main***

Une caméra enregistre l'image de la main.

La caractérisation est obtenue par mesure de la longueur et de la largeur de chaque doigts.

### ***- Dynamique de la signature***

Elle est obtenue par une tablette graphique reliée à un ordinateur

La signature et surtout le mouvement réalisé par la main pour la fabriquer sont analysés en comparaison à plusieurs signatures de référence.

### ***- Dynamique de la frappe clavier***

Un clavier spécial permettant la mesure précise des intervalles dans les séquences de frappe ou la pression des doigts sur le clavier est utilisé.

### ***- Empreinte génétique***

Analyse du code génétique de l'individu.

Demande encore actuellement plusieurs heures.

**Chapitre II**  
**LES PROTOCOLES**  
**DE SÉCURITÉ**

## Introduction

**Problème de la sécurité en réseaux:  
construire des protocoles assurant la  
sécurité**

**en utilisant de la cryptographie**

Deux variantes de solutions.

*Avec un chiffre symétrique (DES, IDEA)  
(à clé privée)*

- On déduit facilement la clé de déchiffrement  $D_k'$  de la clé de chiffrement  $E_k$ .

*Avec un chiffre asymétrique (RSA) (à clé  
publique)*

- On a deux méthodes  $D_k'$  et  $E_k$  ou il est 'impossible' de déduire  $D_k'$  de  $E_k$ .

.  $E_k$  la **clé publique** stockée dans un annuaire.

.  $D_k'$  est la **clé privée**. Elle doit rester secrète.

## Le problème de confidentialité

- Repose sur l'existence d'un chiffre efficace.

### *Cryptographie à clé privée ( $E_k, D_k$ sont des secrets)*

Pour un message  $M$ , on envoie  $E_k ( M )$ .

Personne d'autre que les détenteurs du secret ( $E_k, D_k$ ) ne savent chiffrer ou déchiffrer.

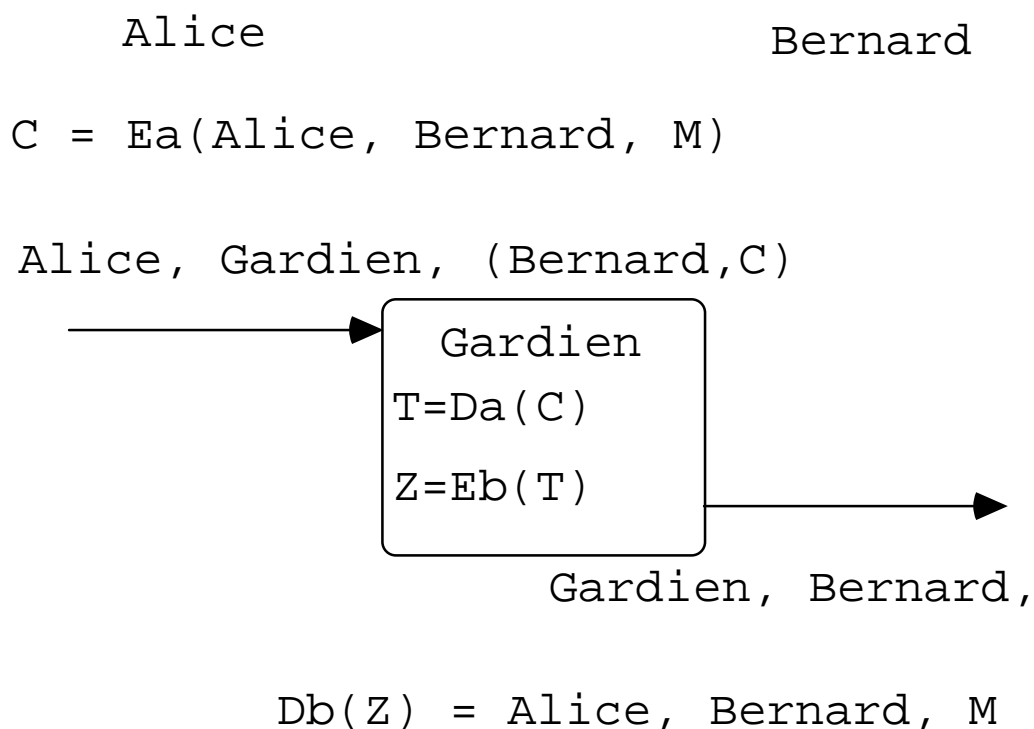
### *Cryptographie à clé publique $E_k$ (clé privée $D_k'$ )*

On envoie  $E_k (M)$ . Tout le monde connaît  $E_k$ , donc tout le monde peut chiffrer.

Seuls les détenteurs du secret  $D_k'$  peuvent retrouver le texte en clair.

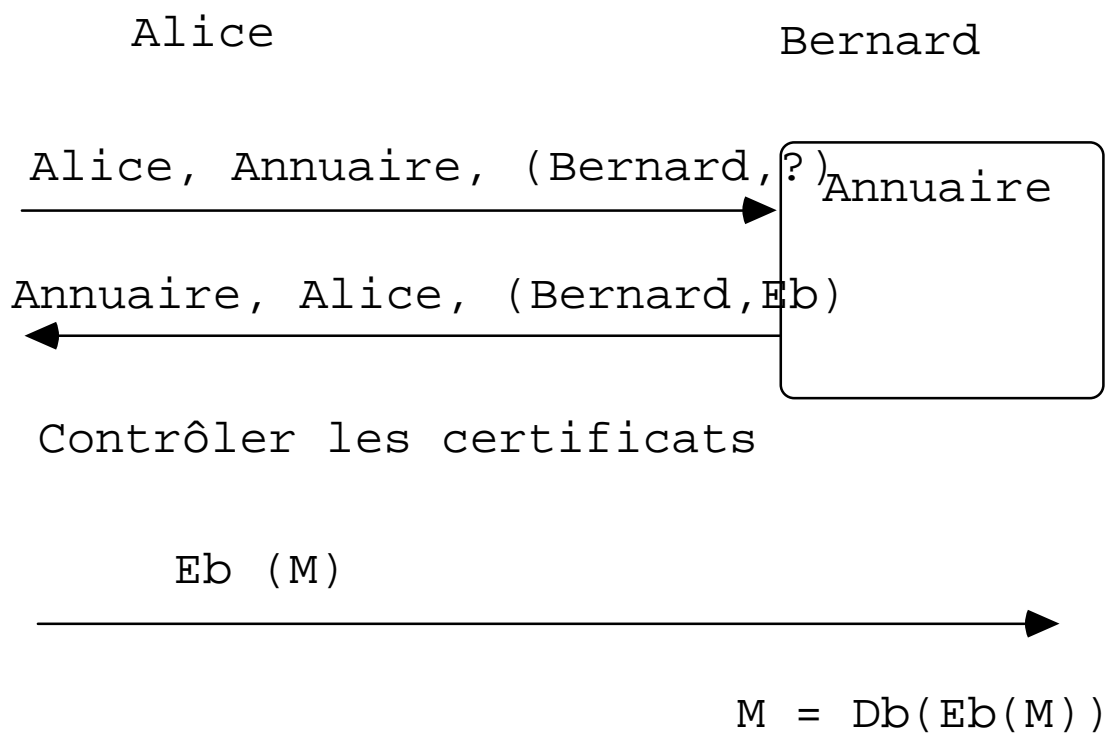
## Protocole de confidentialité avec chiffre symétrique

Utilisation d'un gardien de clés qui connaît la clé de Alice et de Bernard (pour protéger ces clés).



# Protocole de confidentialité avec chiffre asymétrique

Utilisation d'un annuaire pour obtenir la clé publique de Bernard.





## Le problème d'intégrité

On pourrait utiliser le principe de confidentialité basé sur **la possibilité de générer des données correctes par les usagers autorisés** détenteurs du secret.

L'intégrité ne peut être mise en cause que par les détenteurs du secret.

**Problème:** La signature en intégrité est alors **coûteuse** si les données sont **longues**.

**Solution:** Chiffrer une information caractéristique du message **courte**).

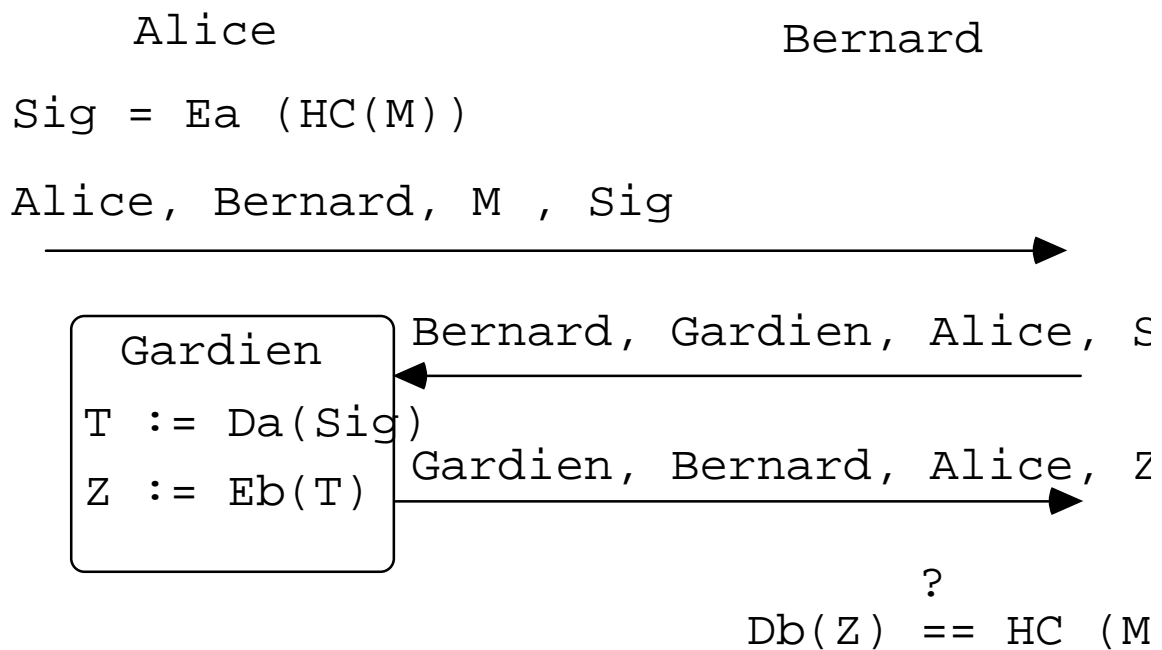
**Fonction de hachage à sens unique** ( "HC" ) caractéristique du message pour s'apercevoir des modifications éventuelles.

## Intégrité en cryptographie à clé privée

- Alice envoie  $M$ ,  $E_a(\text{HC}(M))$
- Tout le monde connaît la méthode de calcul des HC et peut donc calculer  $\text{HC}(M)$ .
- Seul un utilisateur autorisé (connaissant  $E_a$  et  $D_a$ ) peut générer une signature correcte ou la vérifier pour détecter des atteintes à l'intégrité.
  - . soit le destinataire.
  - . soit un tiers de confiance.

## Intégrité en cryptographie à clés privées

Utilisation d'un tiers de confiance gardien des clés qui permet à Bernard de vérifier la signature de Alice.



## Intégrité en cryptographie à clés publiques

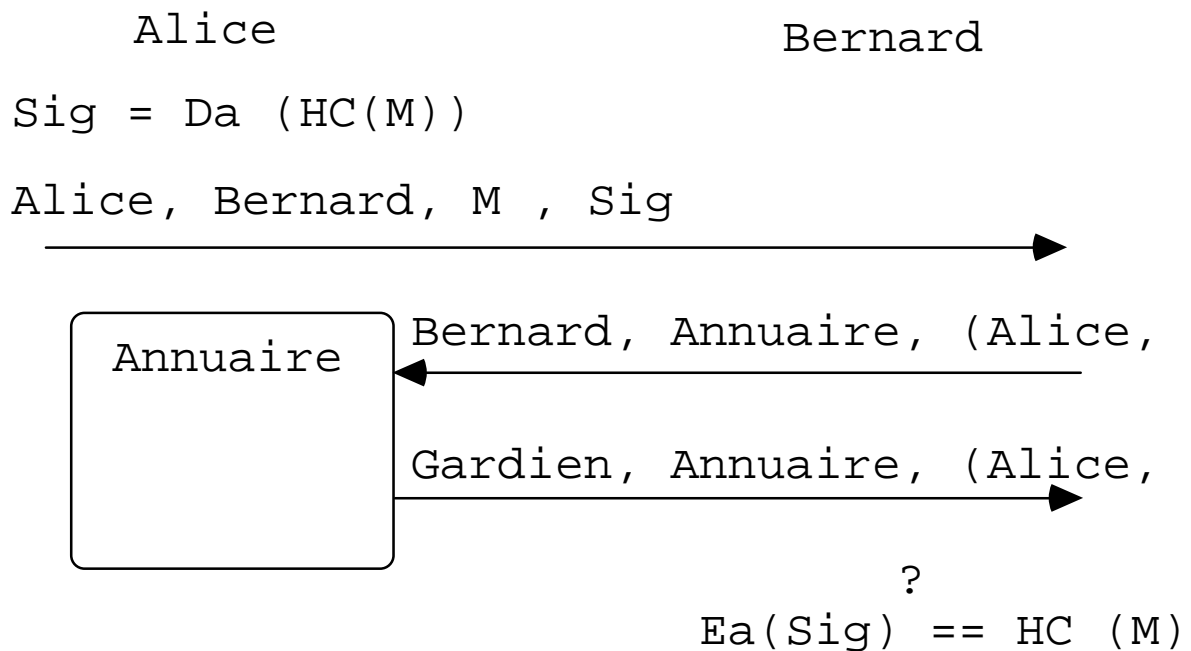
Solution analogue à la précédente

- On envoie  $M$ ,  $D_a(HC(M))$ .
- Il faut **générer le hachage HC crypté au moyen de la clé privée** pour que seuls les utilisateurs autorisés puissent générer des données (des messages) corrects.
- **Tout le monde pourra vérifier ensuite leur correction** au moyen de la clé publique.

**$D(HC(M))$  constitue une signature qui lie un émetteur (le détenteur de la clé privée) à un message  $M$ .**

## Schéma du protocole d'intégrité en chiffre asymétrique

Utilisation d'un annuaire pour obtenir la clé publique de Alice.



## Intégrité d'un message dans un flot de messages

Un flot d'échanges de longue durée doit être caractérisé par une **connexion**.

### Problème de rejeu ("replay")

Un message dupliqué (mais correct du point de vue connexion, séquence et signature) **peut être inséré dans un flot par un usager malveillant** et menacer l'intégrité de l'application.

### Rejeu d'un message ancien

- **D'une connexion ancienne.**
- **De la même connexion ayant un numéro cohérent** dans le flot courant.

## Solutions au problème de rejeu

Utilisation d'un identifiant unique pour chaque message: **nonce** ("use only once")

a) *Utiliser des grands espaces de numérotation* des connexions et des messages (exemple 64 bits)

. Rejeu excessivement difficile (probabilité de succès très faible).

b) *Utiliser un estampillage par l'horloge de l'émetteur*

. La datation nécessite un **protocole de synchronisation d'horloge** (émetteur/récepteur).

. Permet au récepteur de **vérifier la cohérence de la date** et évite ainsi le rejeu.

c) *Utiliser un nombre aléatoire*

. Vérification du nombre utilisé.

## Le problème d'authentification

### Authentification d'un usager par mot de passe

- **Chaque utilisateur a:**

- . un nom d'usager U
- . un mot de passe secret P

- On pourrait stocker le fichier de mots de passe U, P **en clair** mais la protection est trop faible

- **Fonction a sens unique (oneway function) F a clé secrète k => On garde U,  $F_k(P)$ .**

- A l'ouverture de session **l'utilisateur présente U, P: P est immédiatement crypté  $F_k(P)$ .**

- **On compare la valeur obtenue à celle enregistrée (en théorie le fichier peut être accessible en **lecture** puisqu'on ne peut déduire P de  $F_k(P)$ ).**



## Exemple de l'authentification par mot de passe en UNIX

- Le mot de passe est sur **8 caractères**.
- On en fait une **clé de 56 bits** pour chiffrer avec l'algorithme du DES.
- On **chiffre avec cette clé** (le mot de passe) un texte qui est en fait composé de **64 bits à 0** avec la clé précédente.
- On **réitère 25 fois** sur les valeurs obtenues successivement.
- Le résultat est traduit en **11 caractères imprimables** placés dans un fichier (au départ /etc/passwd) accessible en lecture par tous les usagers.

## Problèmes de l'approche UNIX

### Espionnage de la ligne entre la console utilisateur et le calculateur

Le mot de passe est en clair.

Sur ethernet utilisation d'un espion de ligne => interdire l'usage de tels outils.

### Utilisation de techniques de décryptage.

#### 1) Attaque "force brute"

Essayer tous les mots de passe. Facile si l'intrus a pu recopier le fichier (même avec un calculateur peu puissant).

#### 2) Meilleure technique

Les mots de passe sont tirés d'un dictionnaire, d'une liste de prénoms etc => **on peut casser jusqu'à 30% des mots de passe générés à la légère.**

## Solutions pour contrer les tentatives de décryptage

- Interdiction de plus de **n tentatives d'accès en échec.**

- Mise en **protection du fichier des mots de passe** (accès contrôlé par primitive système)

- **Interdiction de l'utilisation** des mots de passe courts ou cassables facilement.

### Clé complémentaire aléatoire

- **Adjonction au mot de passe P de d'une clé complémentaire N** (l'heure de création du mot de passe, une clé aléatoire)

On stocke dans le fichier des mots de passe protégé  $U$  ,  $N$  ,  $E_k(P , N)$ . A chaque ouverture on ajoute au mot de passe  $P$  de l'usager la clé  $N$  avant de crypter.

## Exemple: le "salt" Unix

Pour certains unix une clé complémentaire de 12 bits ("salt") est construite à partir de l'UID (code interne utilisateur) et de l'heure de génération du mot de passe.

Cette clé est ajoutée au mot de passe de l'usager.

On multiplie par 4096 le nombre d'essais à réaliser pour chaque mot de passe par un pirate.

## Protocoles d'authentification dans les réseaux

Assurer la confidentialité des informations  
d'authentification  
=> **utilisation d'un chiffre**

La clé de cryptage est l'analogie du mot  
de passe

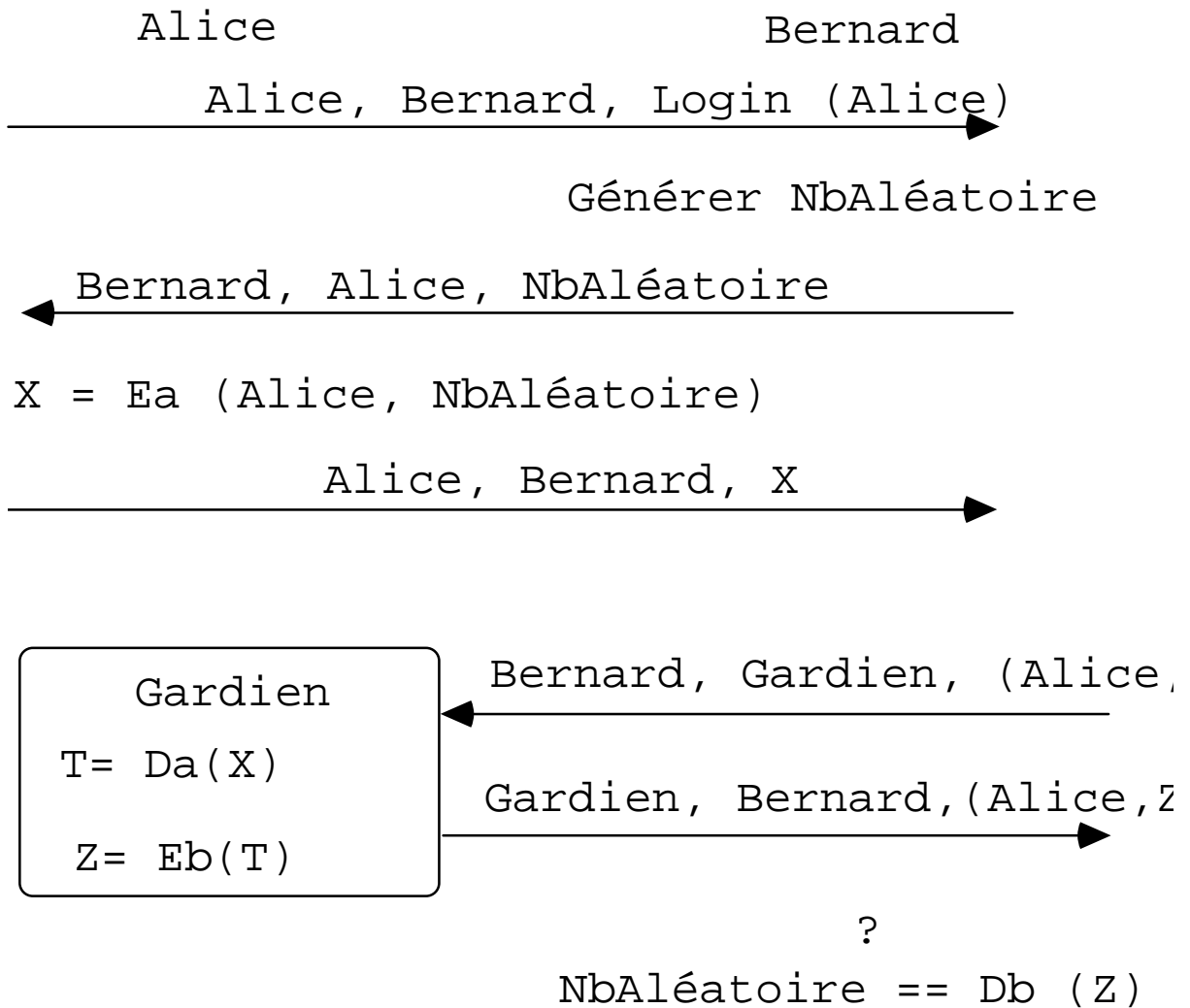
**Problème:** un pirate pourrait écouter  
une authentification pour la réutiliser  
ultérieurement en forme cryptée (**rejeu,**  
**"replay"**)

**Solution:** Utilisation d'un nonce  
(nombre aléatoire)

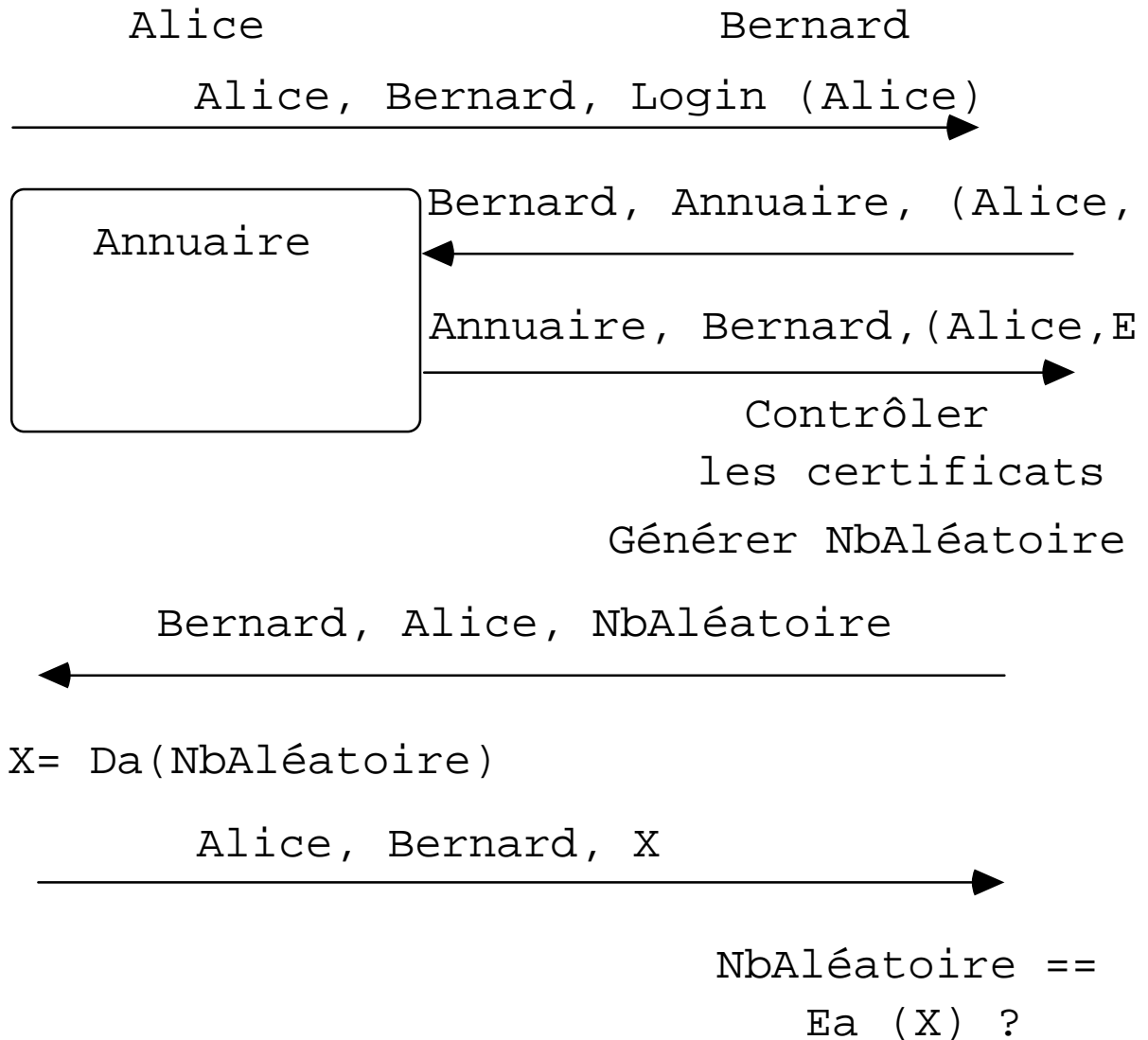
**A chaque ouverture de session on  
échange une valeur différente  
imprévisible.**



## Authentification à clé privée: Un protocole avec utilisation d'un tiers de confiance détenteur des clés.



# Authentification à clé publique





## Remarques : Authentification

- Bernard propose à Alice **un défi** qui est de crypter un nonce.
- **Il n'y a pas de mot de passe** au sens habituel: c'est la connaissance par l'utilisateur U de sa clé privée qui sert de secret.
- Si **Alice retourne le nonce** crypté c'est elle puisque seule Alice connaît Da.
- Bernard **vérifie la cohérence de la valeur du nonce** ce qui empêche la répétition.
- Il faut que **l'annuaire des clefs publiques** (association de Alice et de sa clef publique) **soit protégé**.
- On utilise la commutativité du chiffre asymétrique retenu:  
$$\mathbf{E(D(M)) = D(E(M)) = M}$$

## Un exemple d'authentification à clef privée: Kerberos

- Existence d'un serveur S pour la sécurité. Le serveur à une clef  $K_s$  privée connue de lui seul (méthode de cryptage  $E_{K_s}$ ,  $D_{K_s}$ ).

- Pour chaque entité A il existe une clef privée  $K_a$  connue de A et de S seulement ( $E_{K_a}$ ,  $D_{K_a}$ ).

- A veut s'authentifier à B et réaliser une session chiffrée.

- Pour chaque session le serveur crée une clé de session  $k_{s,e}$  échangée de façon sécurisée.

### Le protocole

A demande au serveur S un ticket (d'authentification) pour B.

A -----> S  
Demande de ticket

- T est la date courante et kses la clef de session créée par le serveur pour la session entre A,B.

A <----- S  
 S,A,Eka{kses},Ekses{s,Eks{A,B,T}}

- A peut déchiffrer la première partie avec ka et obtenir Kses. Il peut alors récupérer Eks{A,B,T} qu'il ne peut déchiffrer. Il envoie cette partie obscure à B.

A----->B  
 A , B , Eks{A,B,T}

- B demande au serveur kses. Il présente Eks{A,B,T}.

B----->S  
 B , S, Eb{Eks{A,B,T}}

- Le serveur déchiffre, retrouve Kses et le renvoie à B.

S----->B  
 S, B, Ekb{A,B,T,Kses}

- A et B dialoguent avec kses.

## Le problème de non répudiation

Éviter un **déni** de responsabilité d'une entité communicante.

- Du **récepteur** qui ne peut dénier avoir reçu une transaction.

- De **l'émetteur** qui ne peut dénier l'avoir émise.

C'est un problème plutôt juridique mais qui doit être réglé dans le cadre de techniques informatiques.

Dans toutes les circonstances l'utilisateur peut prétendre que l'on a usurpé son identité.

Deux catégories de réponses:

- *La responsabilité totale du secret des clés*

- *La notariation*

## **La responsabilité totale du secret des clés**

- Les usagers **ne peuvent en aucun cas arguer de la perte de leur code** s'ils n'en ont pas fait la déclaration immédiate.

- Toute **utilisation du secret hors déclaration** de perte engage la responsabilité du détenteur (Ex: carte bancaire).

- Dans ce cas on peut utiliser un protocole de sécurité (preuve de remise, authentification ... )

- Au procès c'est la bonne foi de l'une des parties contre celle de l'autre (en général en faveur des banques pour la carte bancaire).

- C'est également la qualité des appareils et du protocole de sécurité qui sont en cause.

## Les techniques de notariation

- Les transactions sont effectuées par l'intermédiaire d'une entité juridiquement sûre (un notaire électronique).

- C'est un tiers reconnu contractuellement par les entités qui communiquent.

- Le notaire garantit la sécurité d'une transaction pour réduire l'incertitude en cas de conflit.

**- confidentialité**

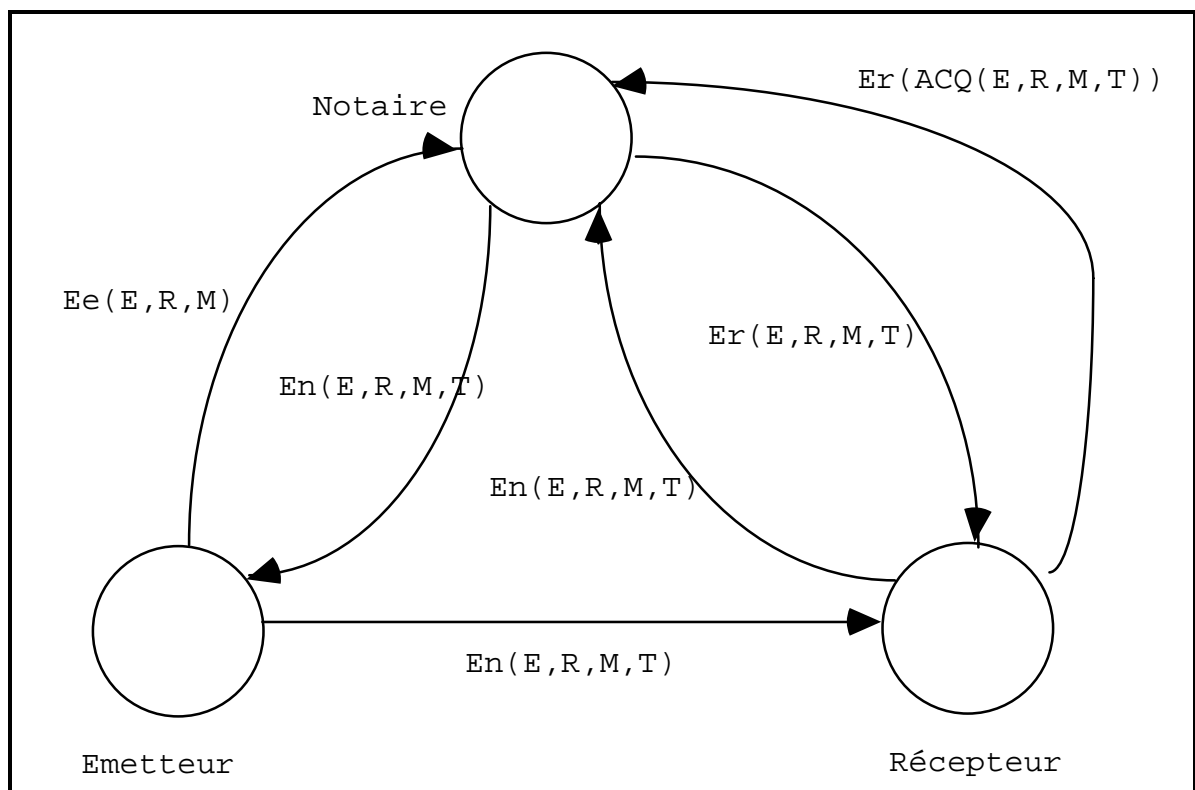
**- authentification**

**- non répudiation**

(datation, journalisation de la transaction etc).

# Un protocole de notarisation utilisant des clés privées.

Le notaire doit disposer des clés  $E_e$ ,  $D_e$  et  $E_r$ ,  $D_r$  (émetteur, récepteur) et d'une clé secrète propre  $E_n$ ,  $D_n$ .



## Commentaires (1)

1 - **E souhaite envoyer le message M** au destinataire R de façon notarisée: il envoie au notaire  $E_e ( E , R , M )$ .

2 - **Le notaire qui reçoit la transaction** peut la décoder puisqu'il connaît  $D_e$ . Il date la transaction T et la journalise:

Enregistrement de E , R , M , T

La transaction ne pourra pas ensuite être reniée par E.

3 - **Le notaire possède une clé secrète personnelle  $E_n$**  qu'il utilise pour signer la transaction

$$S = E_n(E,R,M,T).$$

Il renvoie cette signature en réponse à E qui va la conserver pour preuve de la notariation effectuée.



## Commentaires (2)

4 - **L'émetteur E envoie alors la transaction** à son destinataire sous la forme  $S$  signée par le notaire (il ne peut avoir modifié celle ci entre temps).

R ne peut encore interpréter les informations mais il enregistre  $S$  pour preuve de la requête de E.

5 - **Pour connaître M, R demande au notaire** le déchiffrement de  $E_n$  (  $E$  ,  $R$  ,  $M$  ,  $T$  )

Le notaire envoie à R la transaction chiffrée avec la clé de R soit  $E_r$  (  $E$  ,  $R$  ,  $M$  ,  $T$  ). Seul R peut la lire confidentialité, intégrité et authenticité.

6 - **Pour terminer complètement le protocole** il faut que le notaire dispose d'une preuve de remise à R soit une réponse:

$$E_r ( ACQ ( E , R , M , T ) )$$

que le notaire enregistre.