

ch 2 : Chiffrement symétrique

I] Modélisation

① Confidentialité parfaite (one-time pad)

② Chiffrement par blocs

x (bloc de n bits, n fixé)



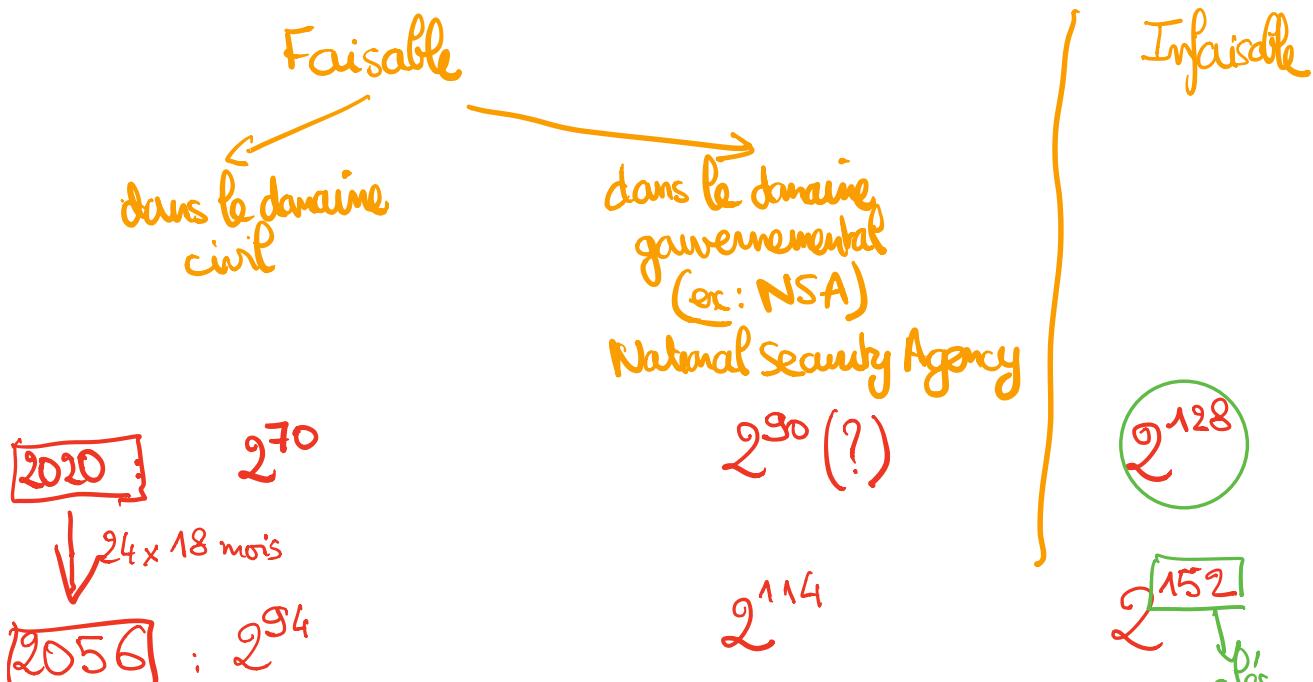
$$y = E_K(x)$$

inconnue

Possibilité pour l'attaquant : résoudre cette équation par une méthode de recherche exhaustive" $\rightarrow (2^k)$ possibilités à essayer

\rightarrow Sécurité calculatoire (\neq sécurité parfaite du one-time pad)

le nombre d'opérations à effectuer dépasse en pratique les capacités de calcul de l'attaquant.



Loi de Moore : La puissance de calcul disponible double tous les 18 mois
 (1956)

③ Types d'attaques

Pour définir la sécurité d'un système, il faut préciser

- 1) Ce que l'attaquant cherche à faire (= son objectif)
- 2) Où où l'attaquant a accès

Dans le cas du chiffrement symétrique

- 1) Objectif
 - retrouver la clé
 - déchiffrer un message chiffré

- 2) Plusieurs types d'attaques

- attaques à clés connues: l'attaquant connaît un ou plusieurs couples (M, C) (où $C = E_K(M)$)
- attaques à clés choisies: l'attaquant peut choisir lui-même un ou plusieurs messages clairs (M_1, M_2, \dots) et obtenir leurs chiffres (C_1, C_2, \dots) avec $\forall i, C_i = E_K(M_i)$
- attaques à chiffrés choisisis: l'attaquant peut choisir lui-même un ou plusieurs messages chiffrés (C_1, C_2, \dots) et obtenir les messages clairs correspondants (M_1, M_2, \dots) avec $\forall i, C_i = E_K(M_i)$

Ex: Un système de chiffrement symétrique est en général considéré comme "sûr" si étant donné un attaquant qui peut

- (scénario clé choisie) \rightarrow - soit choisi des messages clairs M_1, M_2, \dots et ne obtient C_1, C_2, \dots avec $C_i = E_K(M_i)$
- (scénario chiffre choisi) \rightarrow - soit choisi des messages chiffrés C'_1, C'_2, \dots et obtenir M'_1, M'_2, \dots avec $C'_i = E_K(M'_i)$

Si on lui donne un message chiffré C''
 $(\notin \{C_1, C_2, \dots\} \text{ et } \notin \{C'_1, C'_2, \dots\})$
 il est calculatoirement difficile de trouver
 le message clair M'' tel que $C'' = E_K(M'')$

↓
 ↗ impossible

III] Data Encryption Standard (DES)

① Histoire
 années 1970 : besoin de chiffrement symétrique
 dans le domaine civil

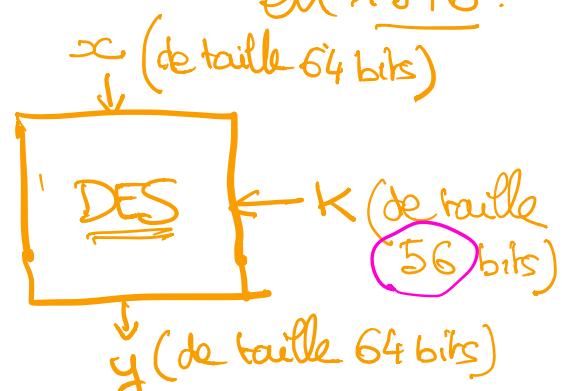
1975 : NBS (National Bureau of Standards)
 a lancé un appel d'offres

→ IBM a proposé l'algorithme LUCIFER

Après modifications (notamment par la NSA)
 LUCIFER est devenu DES en 1976

DES a été publié sous la forme d'un standard
 en 1976.

② Description



2020 : $\begin{cases} \text{faisable dans le ciel} \approx 2^{70} \\ \text{infaisable} \approx 2^{128} \text{ calculs élémentaires} \end{cases}$

$\downarrow \approx 30 \times 18 \text{ mois}$

1976 : $\begin{cases} \text{faisable dans le ciel} \approx 2^{40} \\ \text{infaisable} \approx 2^{98} \end{cases}$

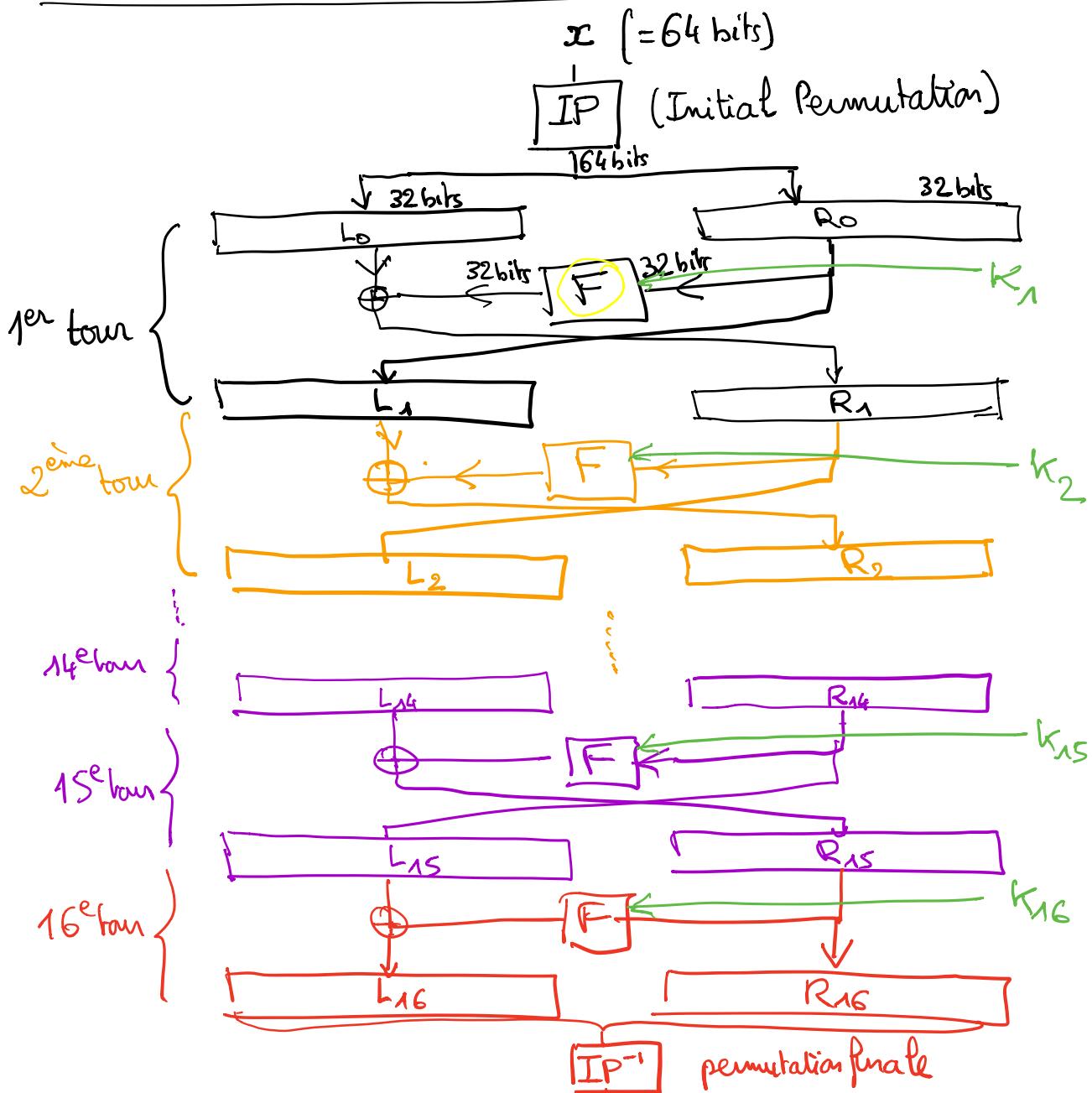


Schéma de Feistel

Sous-système d'équations

$$1^{\text{er}} \text{ tour : } \begin{cases} L_1 = R_0 \\ R_1 = L_0 \oplus F(R_0, K_1) \end{cases}$$

\downarrow (= inverse de la permutation initiale)
 y (= 64 bits)

1^{re} sous-clé (48 bits)
 (obtenue à partir de la clé K)

$$2^{\text{e}} \text{ tour : } \begin{cases} L_2 = R_1 \\ R_2 = L_1 \oplus F(R_1, K_2) \end{cases}$$

2^{re} sous-clé (48 bits)
 (obtenue à partir de K)

$$15^{\text{e}} \text{ tour : } \begin{cases} L_{15} = R_{14} \\ R_{15} = L_{14} \oplus F(R_{14}, K_{15}) \end{cases}$$

$$16^{\text{e}} \text{ tour : } \begin{cases} L_{16} = L_{15} \oplus F(R_{15}, K_{16}) \\ R_{16} = R_{15} \end{cases}$$

Pour qu'un algorithme E de chiffrement par blocs soit utilisable, il faut vérifier 3 conditions

a) E_K doit être une bijection, c'est-à-dire :

$$\begin{cases} E_K \text{ injective : } E_K(x) = E_K(x') \Rightarrow x = x' \\ E_K \text{ surjective : } \forall y, \exists x / E_K(x) = y \end{cases}$$

b) Étant donné x et K , le calcul de $E_K(x)$ doit se faire en temps raisonnable

c) Étant donné y et K , le calcul de $D_K(y) = E_K^{-1}(y)$ doit se faire en temps raisonnable

Pour le DES

(schéma de Feistel)

il suffit de vérifier que c'est vrai pour chaque
tun (car le DES est la composée
de 16 tuns)

ex: 1er tun

$$\begin{cases} L_1 = R_0 \\ R_1 = L_0 \oplus F(R_0, K_1) \end{cases}$$
$$\Leftrightarrow \begin{cases} L_0 = R_1 \oplus F(R_0, K_1) = R_1 \oplus F(L_1, K_1) \\ R_0 = L_1 \end{cases}$$

Cinéquences: • le 1er tun est une bijection
(quelle que soit la fonction F)

- Si F peut se calculer en temps raisonnable, alors le 1er tun
(dans le sens chiffrage ou dans le
sens déchiffrement) peut se
calculer en temps raisonnable