Algorithmes de chiffrement symétrique par bloc (DES et AES)

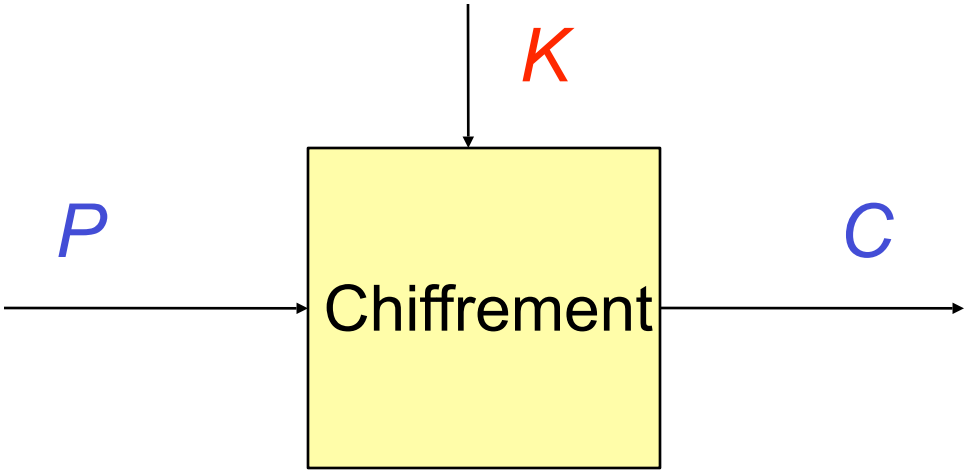
Pierre-Alain Fouque

Equipe de Cryptographie

Ecole normale supérieure

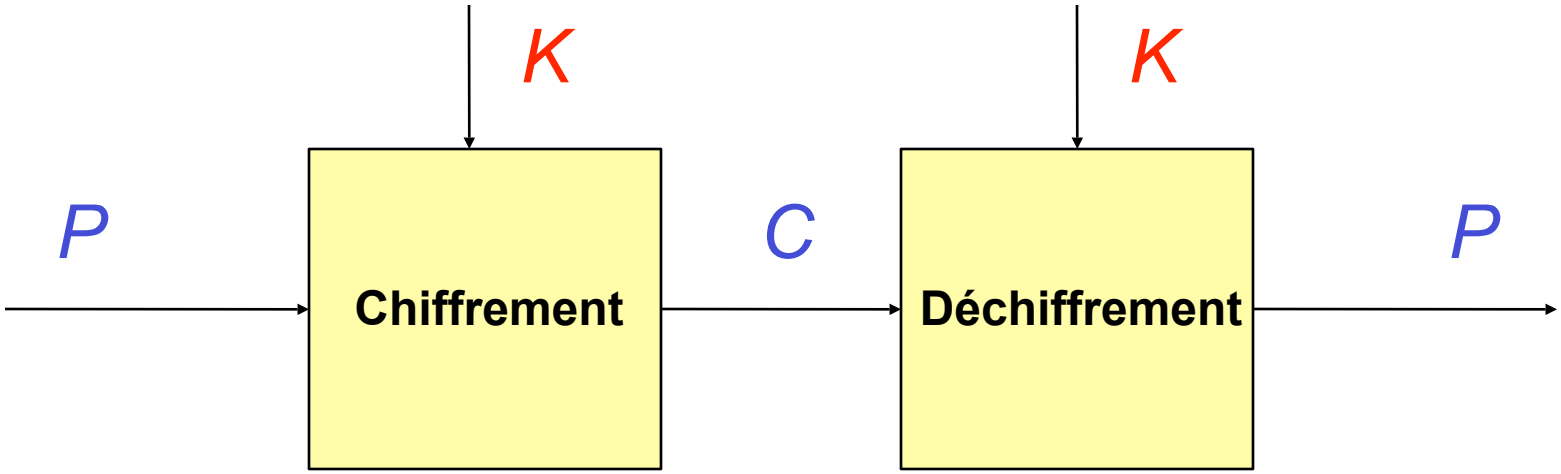
Chiffrement symétrique

Définition : *Un algorithme de chiffrement symétrique transforme un message en clair P avec une clé secrète K. Le résultat est un message chiffré C*



Chiffrement symétrique

La fonction de chiffrement doit être inversible



Deux grandes catégories

Chiffrement par bloc

- P est traité par blocs de données (ex: 64 bits ou 128 bits)
- Algorithmes : DES, AES, IDEA, RC6, BLOWFISH, …

Chiffrement par flot

- P est traité bit par bit
- Algorithmes :RC4, Bluetooth E0/1, GSM A5/1,

Chiffrement par bloc

- Une des primitives (« briques ») les plus largement utilisées en cryptographie
  - Chiffrement symétrique
  - Protection de l'intégrité
  - Construction de fonctions de hachage, de générateur pseudo-aléatoire, etc ….

Historique

- Algorithmes « historiques » (avant 1970)
- 1970-2000 : **DES** (Data Encryption Standard) et autres algorithmes (FEAL, IDEA, RC5, …)
- 2000-2004 : **AES** (Advanced Encryption Standard) et algorithmes récents (RC6, CAMELLIA, …)

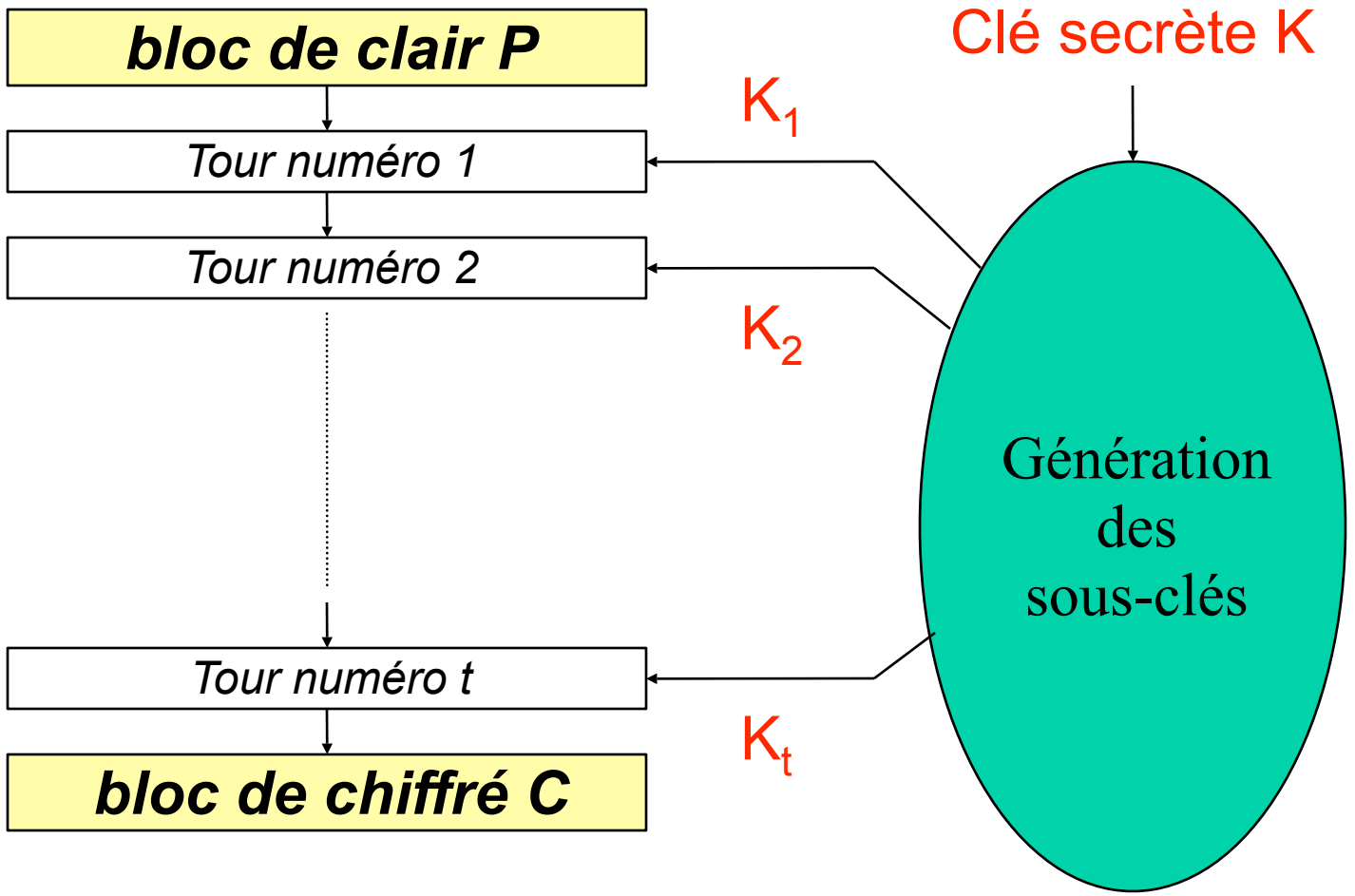
Sécurité

- Deux principaux paramètres de sécurité
  - La taille du bloc (e.g. $n = 64$ ou 128 bits). Les modes opératoires permettent généralement des attaques quand plus de $2^{n/2}$ blocs sont chiffrés avec une même clé
  - La taille de clé (e.g. $k = 128$ bits). Pour un bon algorithme, la meilleure attaque doit coûter 2^k opérations (recherche exhaustive)

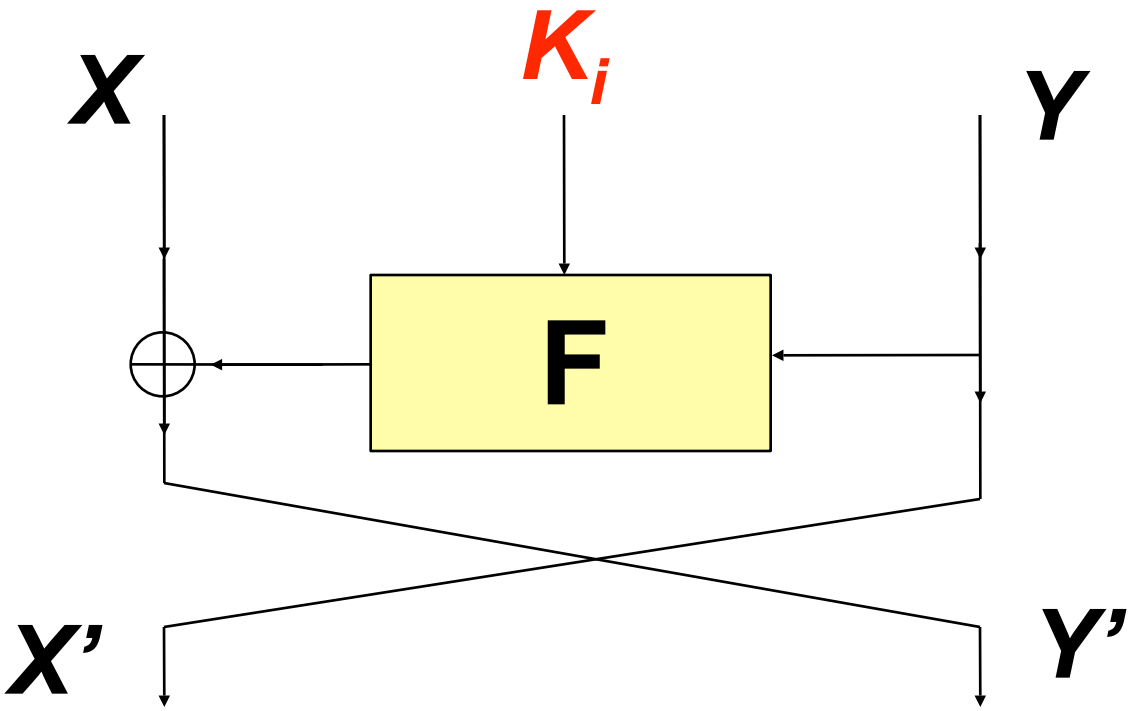
Construction

- Algorithmes itératifs : une fonction de tour est itérée t fois
- Génération de clés de tour (ou sous-clés) à partir de la clé secrète K
- Utilisation d'opérations simples et efficaces (+, XOR, *, tableaux)

Construction

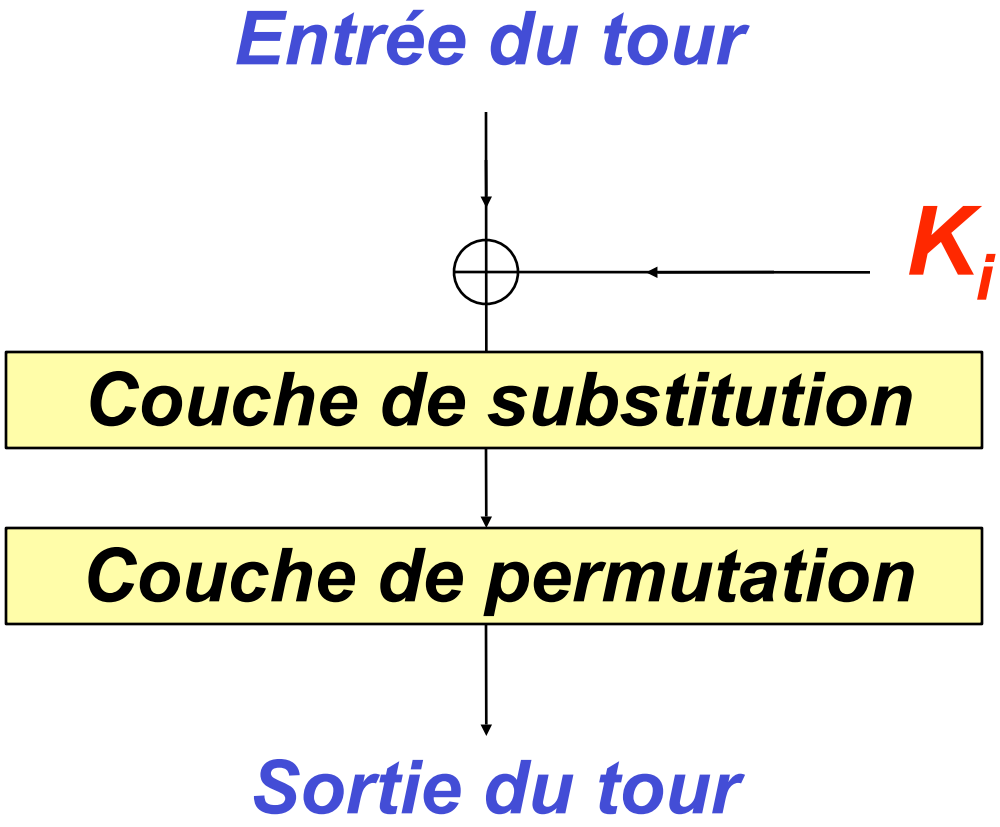


Schémas de Feistel



Fonction de tour inversible même si F ne l'est pas !!

Substitution-Permutation



Le DES

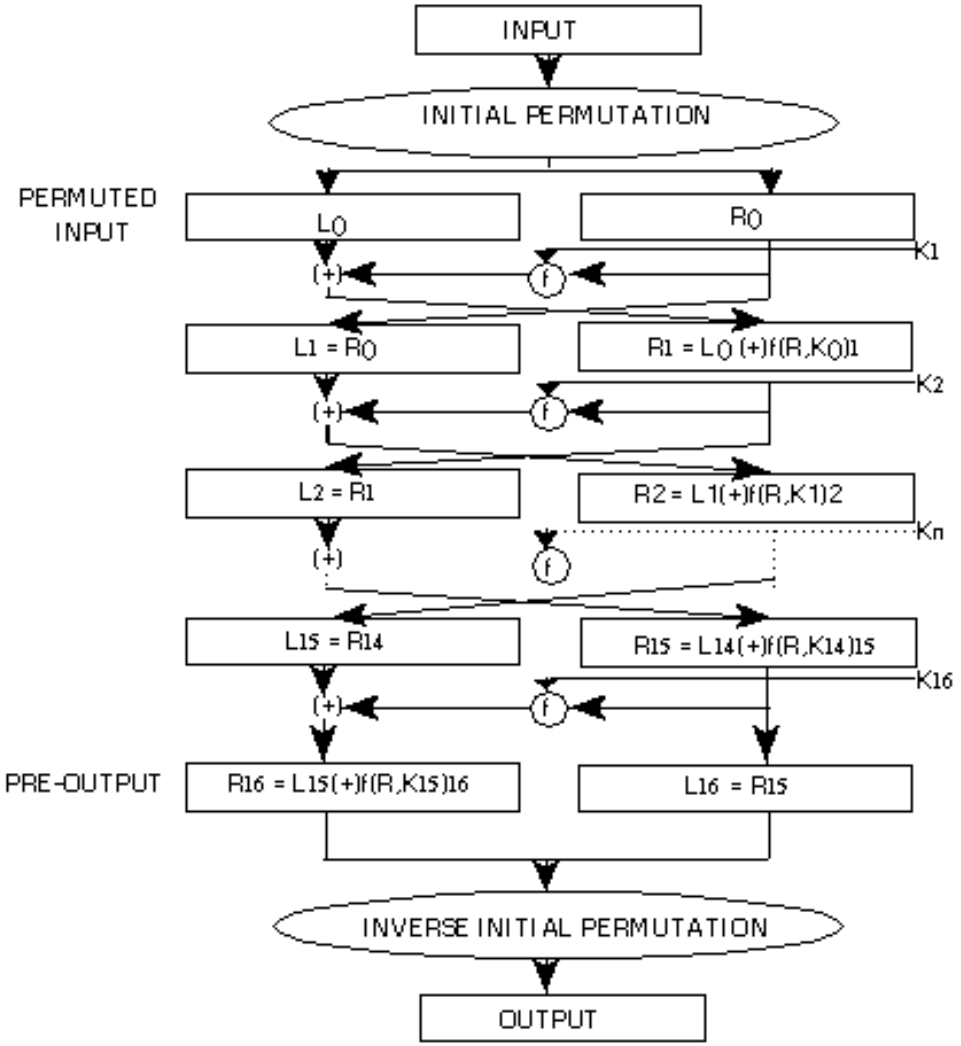
- Algorithme développé par IBM dans les années 1970 (Lucifer)
- Adopté comme standard US par le NBS (FIPS 46-2), en 1977
- Taille de bloc = 64 bits
- Taille de clé = 56 bits
- Schéma de Feistel à 16 tours

Le DES (schéma du NIST)

Tour numéro 1

Tour numéro 2

Tour numéro 16

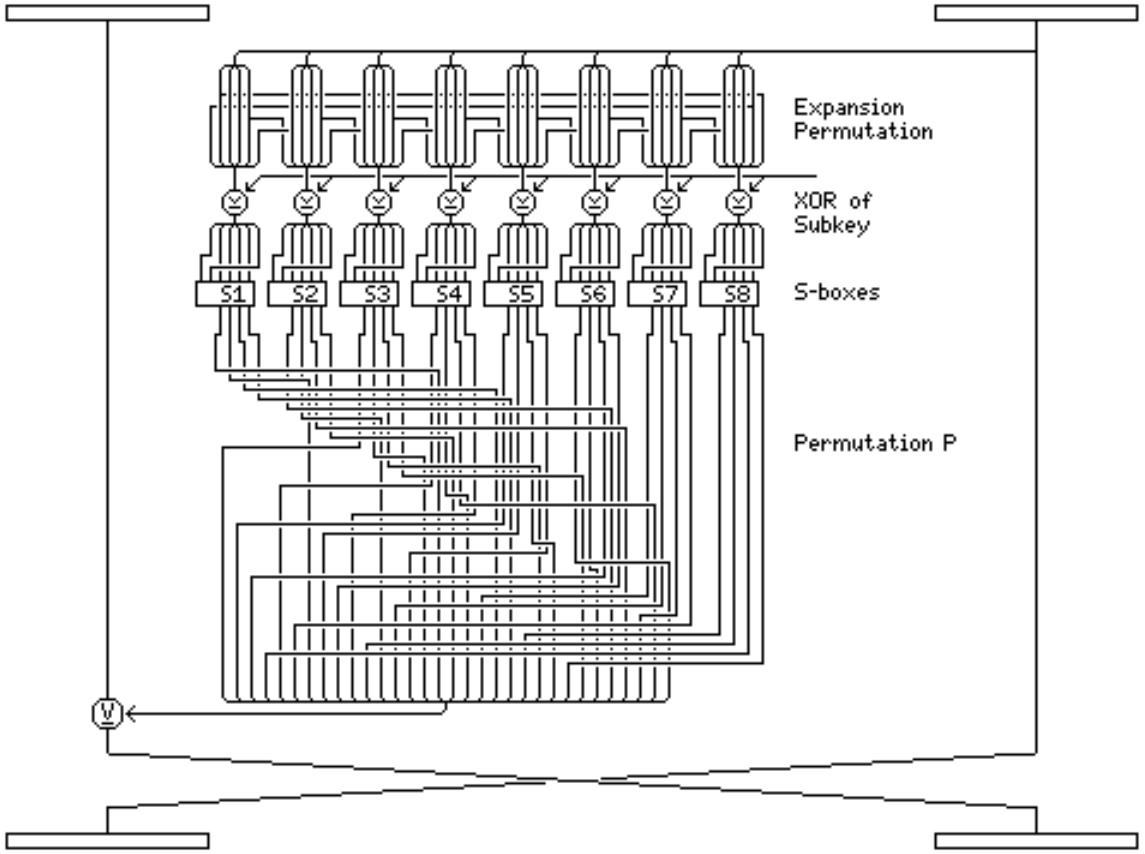


La permutation initiale

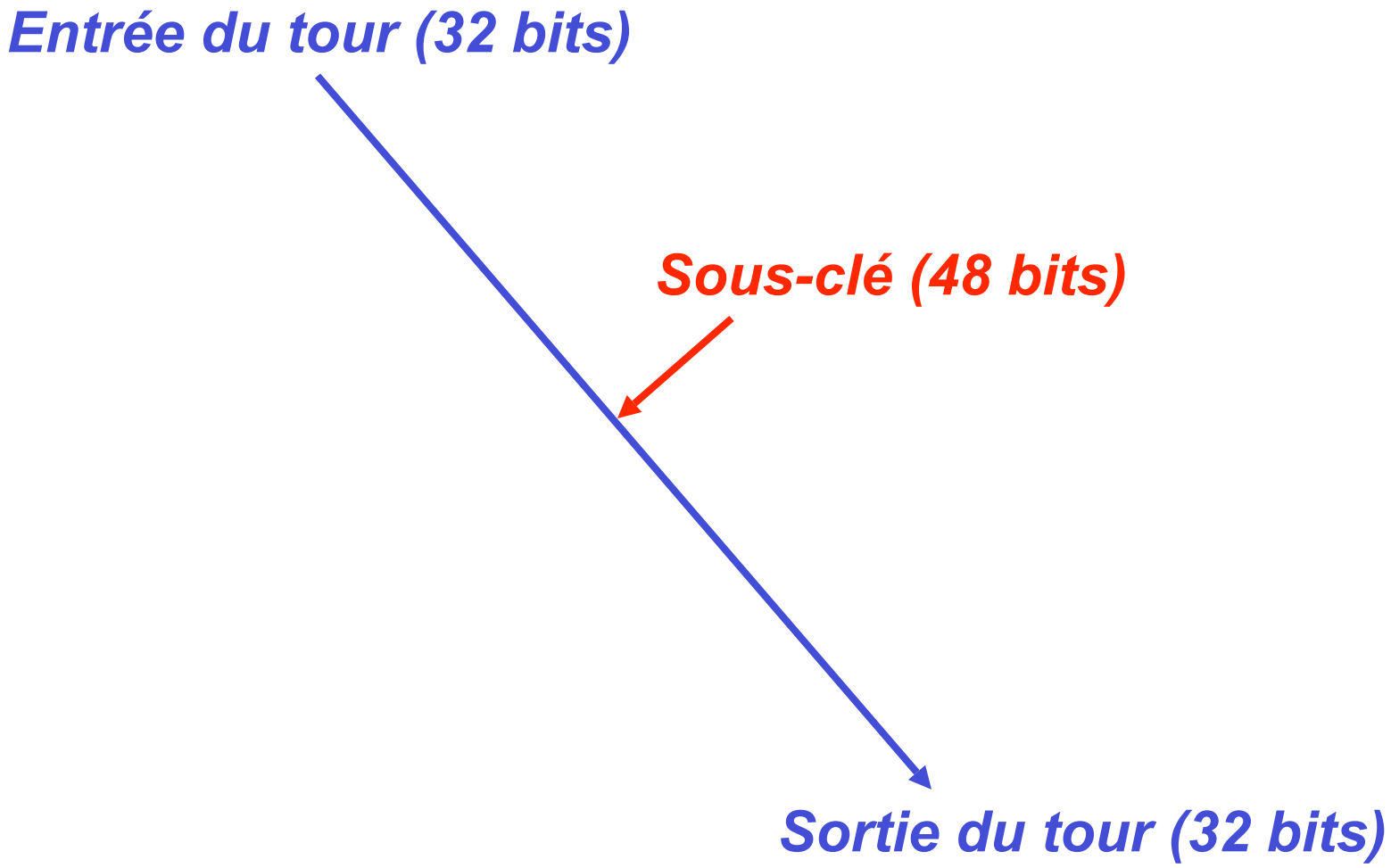
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Le bit numéro 21 de la sortie
…
provient du bit numéro 30 de l'entrée

Fonction F du DES

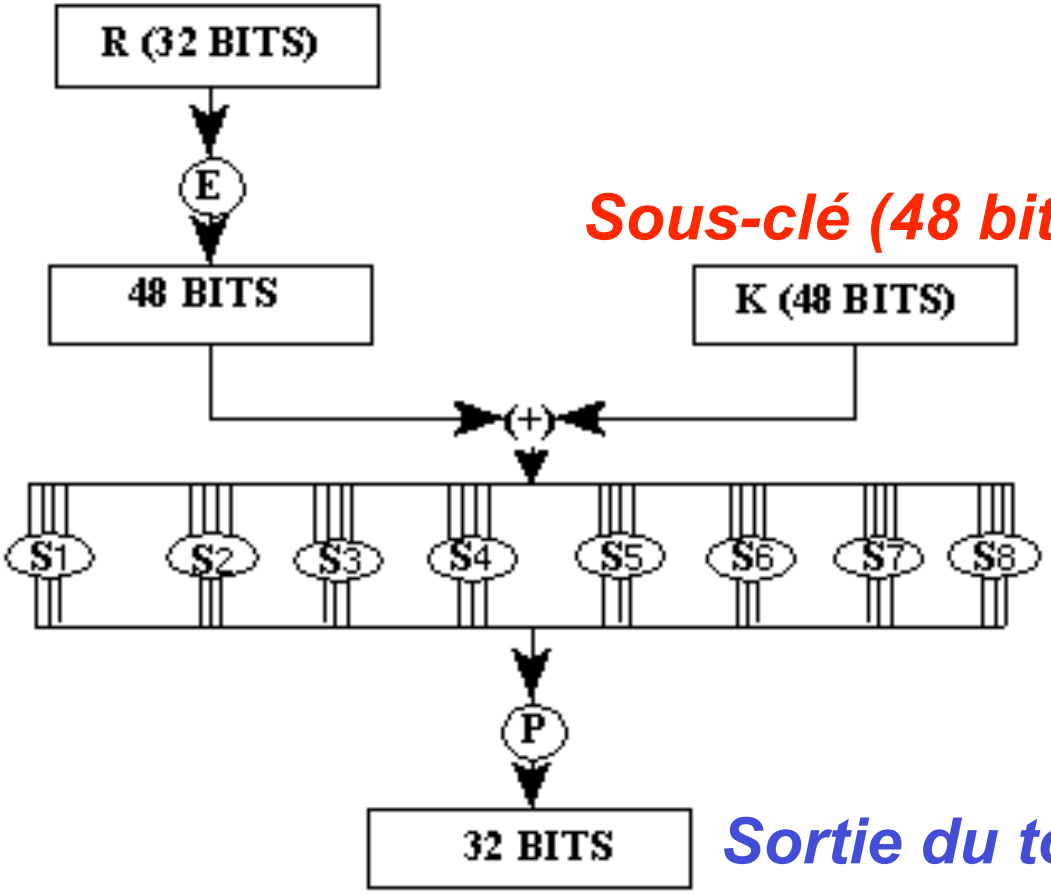


Fonction F du DES



Fonction F du DES

Entrée du tour (32 bits)

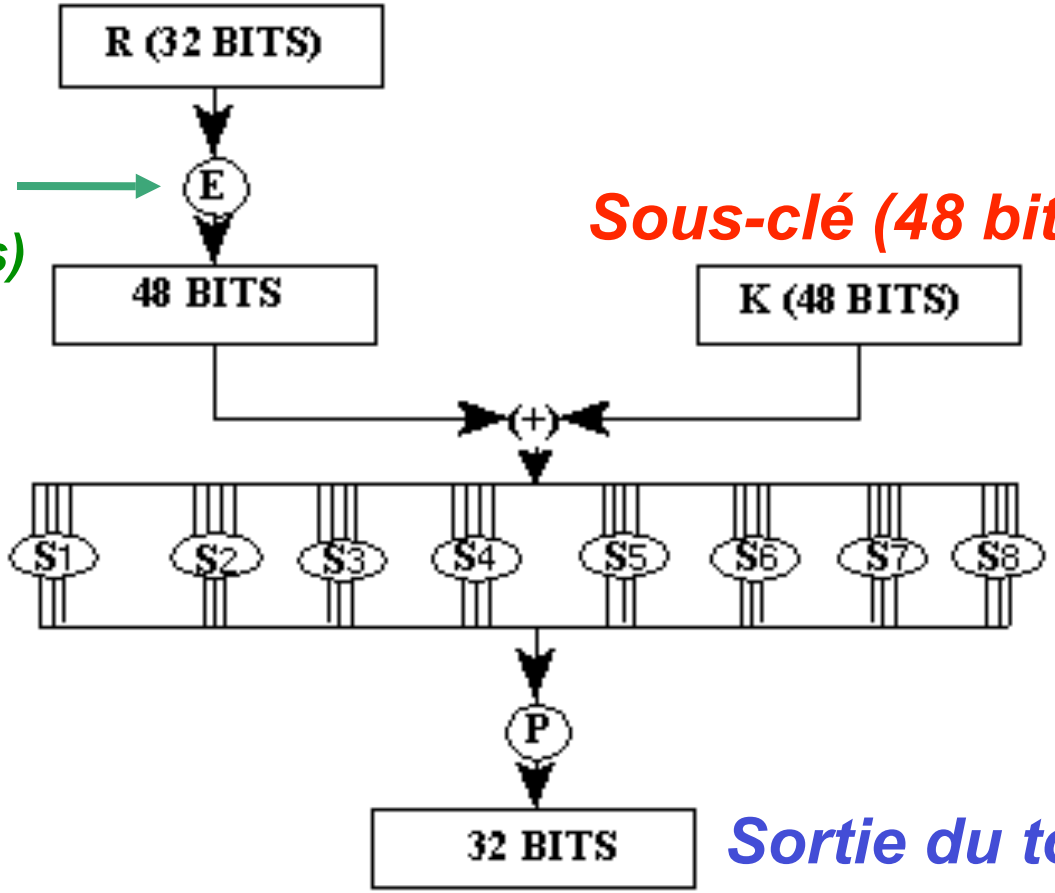


Sous-clé (48 bits)

Sortie du tour (32 bits)

Fonction F du DES

Entrée du tour (32 bits)



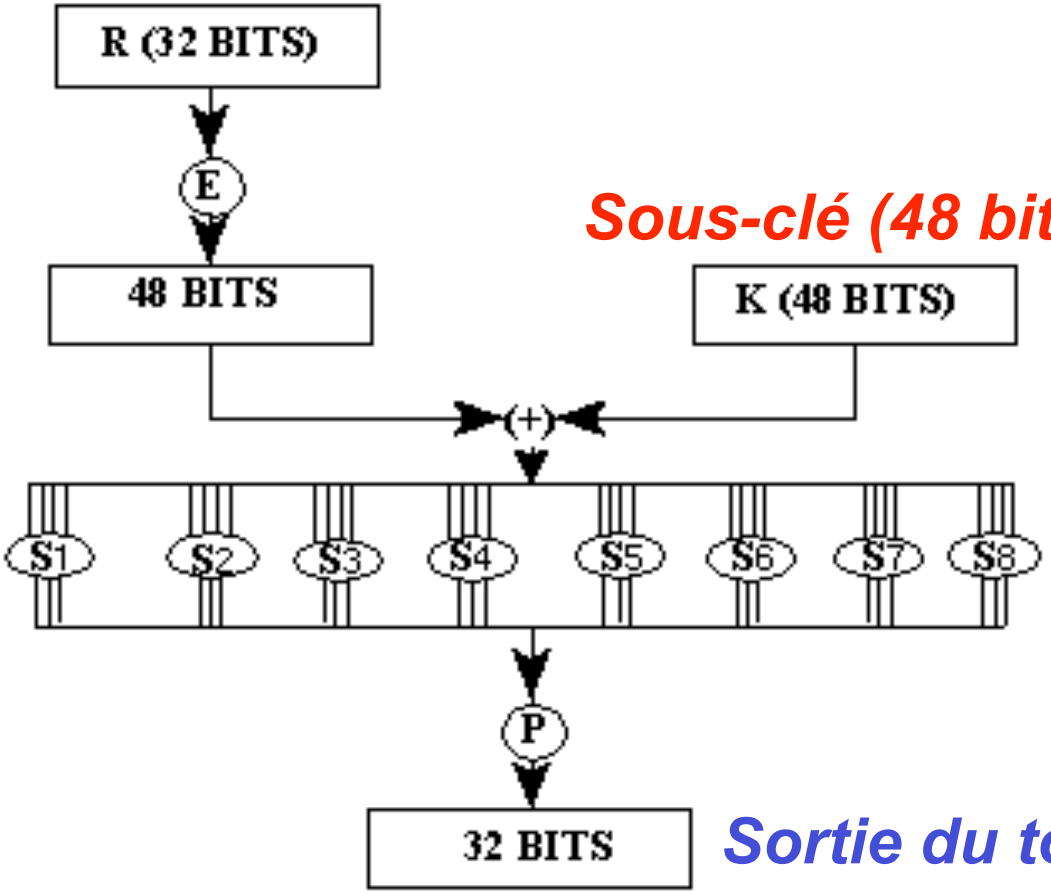
Expansion (fonction de 32 vers 48 bits)

Sous-clé (48 bits)

Sortie du tour (32 bits)

Fonction F du DES

Entrée du tour (32 bits)

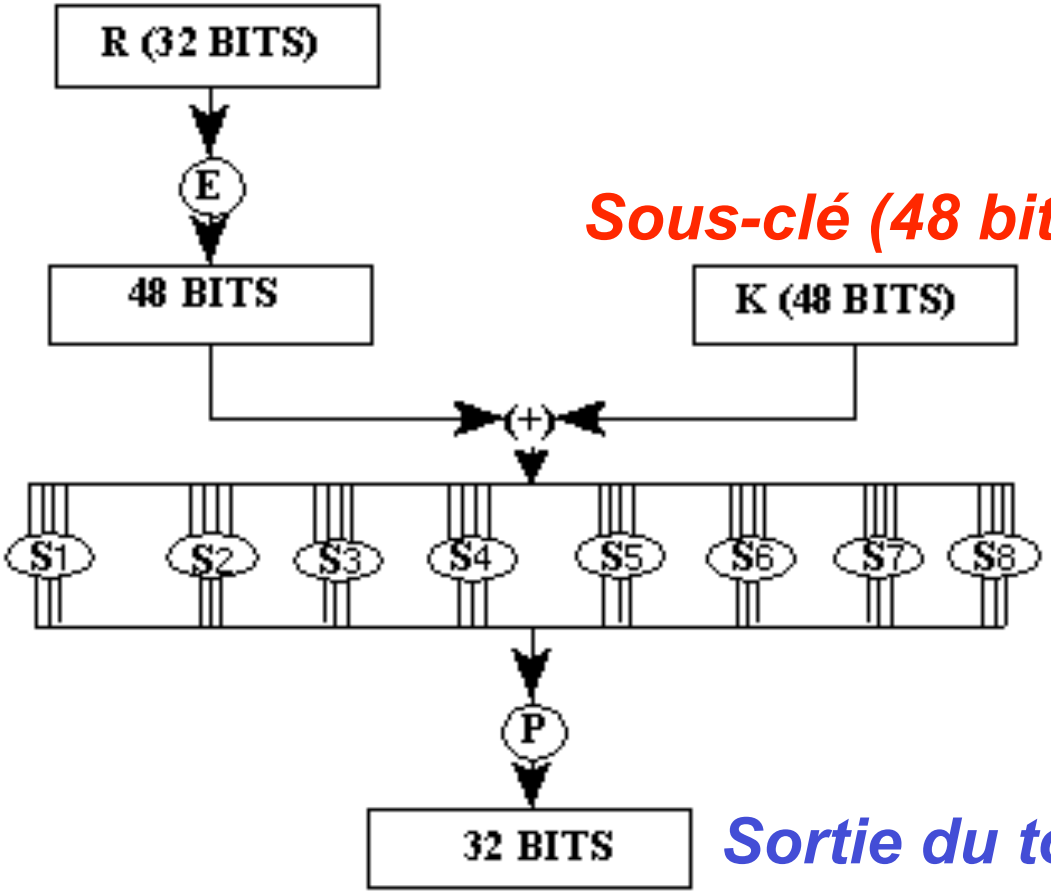


Sous-clé (48 bits)

Sortie du tour (32 bits)

Fonction F du DES

Entrée du tour (32 bits)



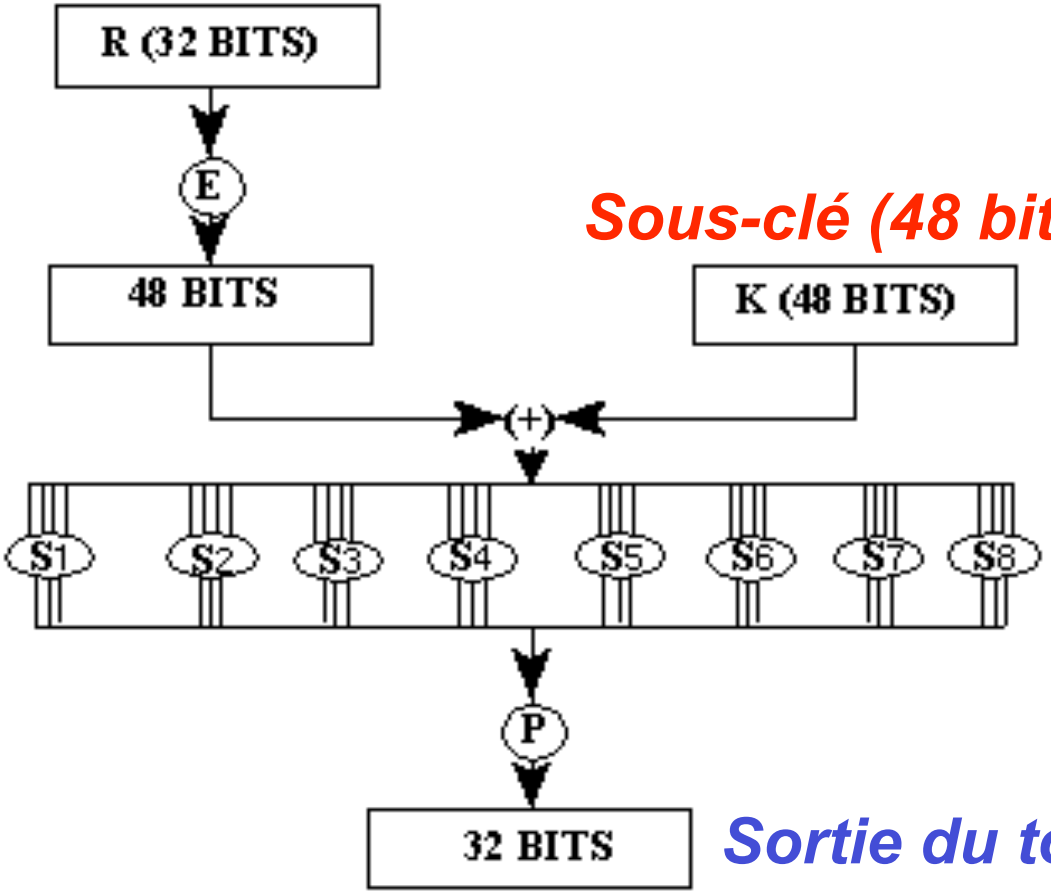
Sous-clé (48 bits)

Boîtes S (fonctions de 6 vers 4 bits)

Sortie du tour (32 bits)

Fonction F du DES

Entrée du tour (32 bits)

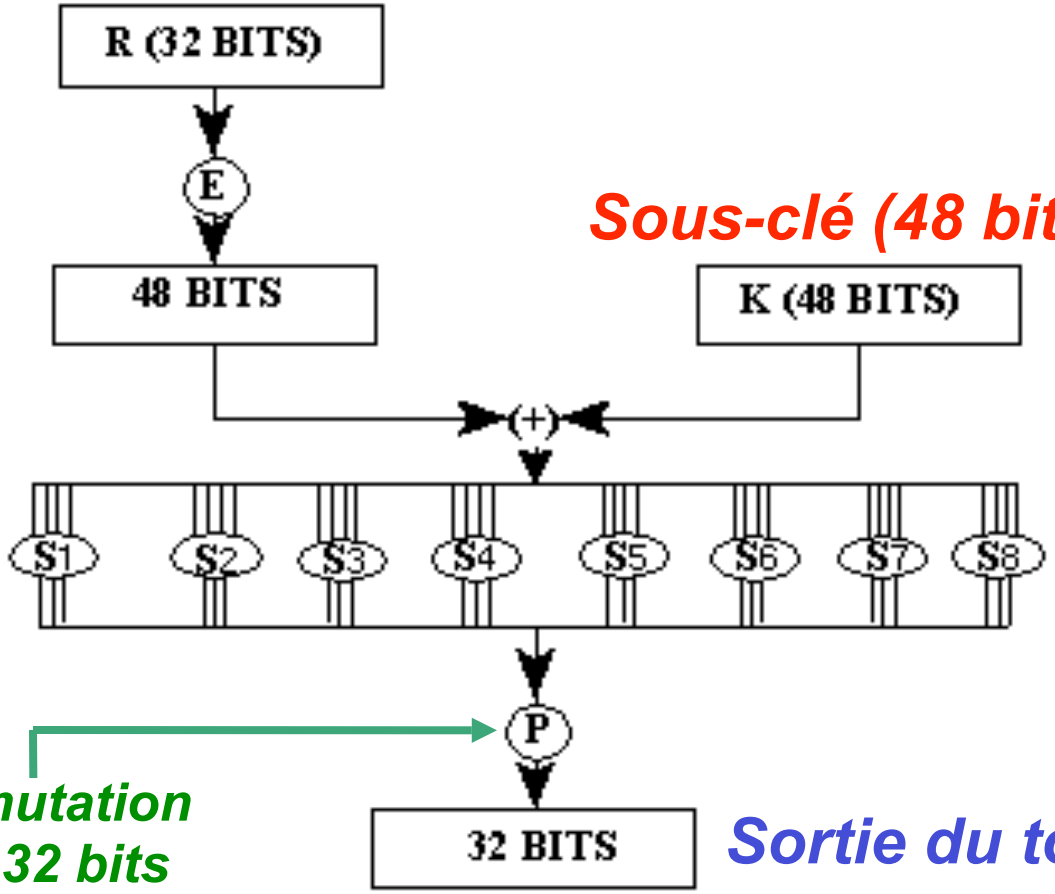


Sous-clé (48 bits)

Sortie du tour (32 bits)

Fonction F du DES

Entrée du tour (32 bits)



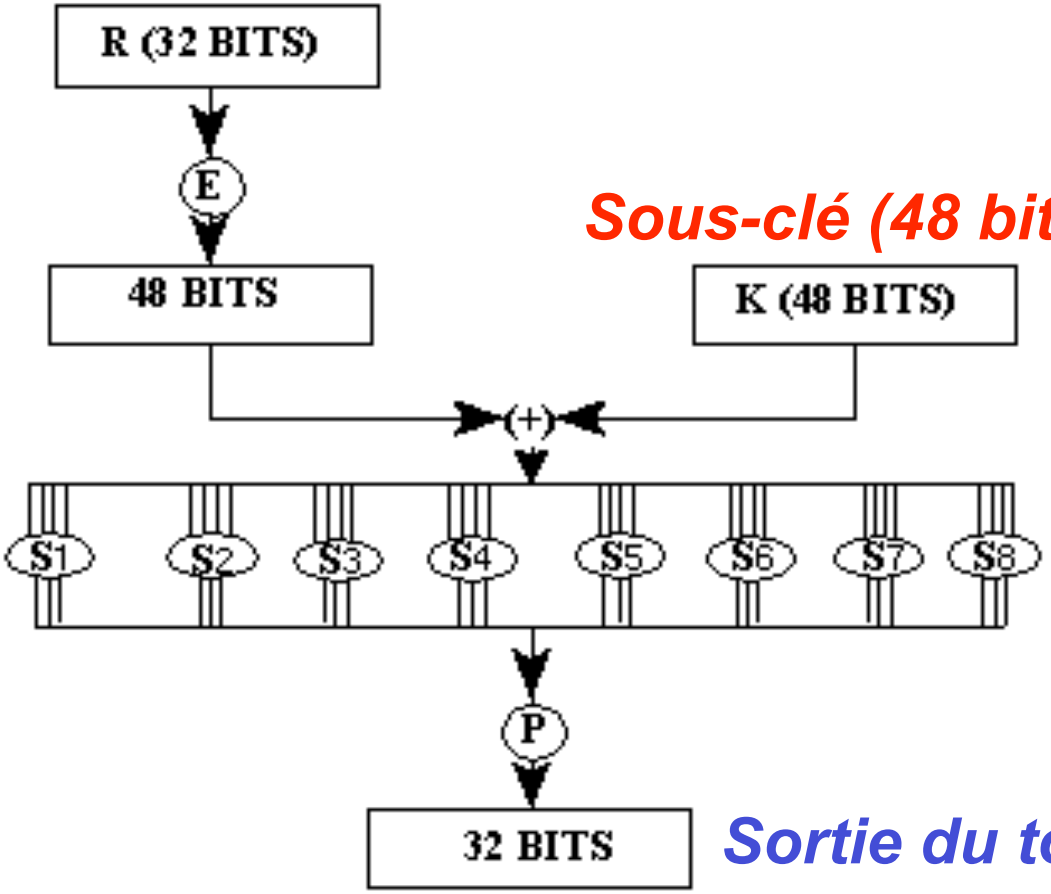
Sous-clé (48 bits)

Permutation sur 32 bits

Sortie du tour (32 bits)

Fonction F du DES

Entrée du tour (32 bits)



Sous-clé (48 bits)

Sortie du tour (32 bits)

Intuition

- La fonction de tour n'a pas besoin d'être inversible
- Expansion de l'état interne de 32 à 48 bits
- Ajout de la sous-clé
- Réduction de l'état interne de 48 à 32 bits grâce aux boîtes S (non-linéarité)
- Permutation (diffusion)

Expansion (32 → 48)

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

Le bit numéro 15 de la sortie
…
provient du bit numéro 10 de l'entrée

Certains bits de l'entrée sont dupliqués (ex: bit 32)

Ajout de la sous-clé

Il s'agit d'un simple XOR, bit à bit, entre

- l'état après expansion (48 bits)
- la sous-clé du tour correspondant (48 bits)

Les boîtes S

- On applique, en parallèle, 8 boîtes (fonction fixe) de 6 bits vers 4 bits
- Ceci réduit donc l'état interne de 8 x 6 = 48 bits à 8 x 4 = 32 bits
- Chaque boîte S est codée comme un tableau avec $2^6 = 64$ entrées

La permutation (32 → 32)

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

Le bit numéro 11 de la sortie
…
provient du bit numéro 23 de l'entrée

Commentaires

Ces choix peuvent paraître arbitraires mais :

- Toutes les briques sont très simples à coder et efficaces en hardware
- Les boîtes S apportent la non-linéarité
- Expansion et permutation garantissent une diffusion rapide

Dérivation des sous-clés

- A chaque tour, on choisit 48 des 56 bits de la clé pour former la sous-clé
- Cette sélection se fait grâce à des permutations circulaires de la clé et des tables d'extraction fixes

Attaques contre le DES

- Avant 1990, attaques contre des versions réduites (t < 16 tours)
- 1990-1992 : cryptanalyse différentielle (Biham et Shamir)
- 1993-1994 : cryptanalyse linéaire (Matsui)
- autres attaques (DaviesMurphy, bilinear …)

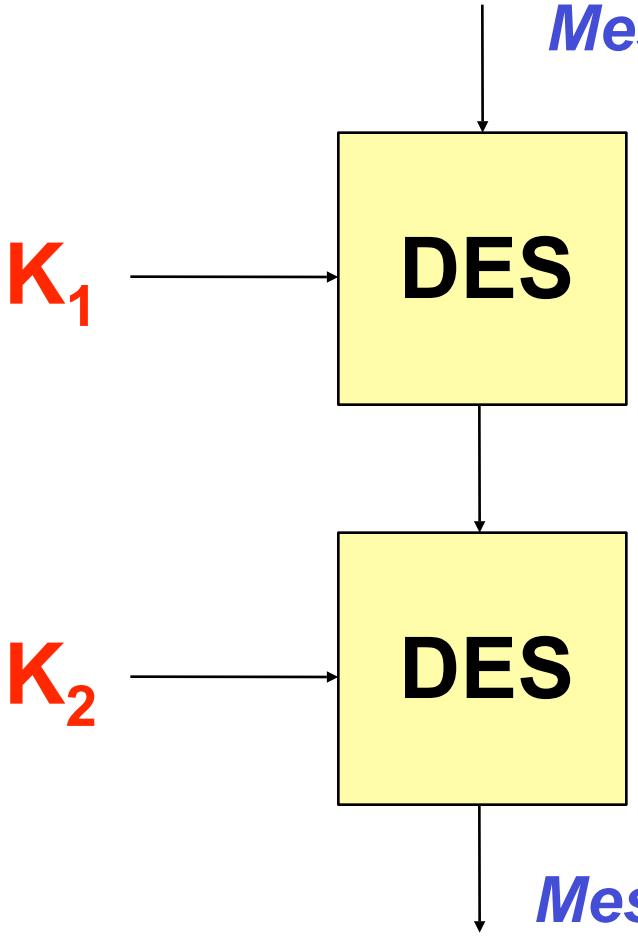
En pratique, le plus efficace reste la recherche exhaustive

Problèmes du DES

- Taille de clé (recherche exhaustive en 2^{56} est réaliste) → utilisation du Triple-DES
- Taille du bloc (attaques avec 2^{32} messages)
- Cryptanalyse linéaire et différentielle

Malgré tout, le DES est un algorithme très bien conçu : il a plutôt bien résisté à 30 ans de cryptanalyse

Double-DES



On obtient un algorithme avec

- taille de bloc 64 bits
- taille de clé 112 bits

Attaque par le milieu

- Attaque pour retrouver les clés secrètes
- L'attaquant doit avoir accès à seulement ***2 couples (clair, chiffré) connus***
- **Objectif** : retrouver les clés secrètes avec la même complexité que pour un simple DES

Attaque par le milieu

- Attaque naïve : recherche exhaustive des 2^{112} clés possibles
- Attaque par le milieu : compromis temps-mémoire pour diminuer la complexité
  - 2^{56} opérations
  - 2^{56} couples (clair,chiffré) en mémoire

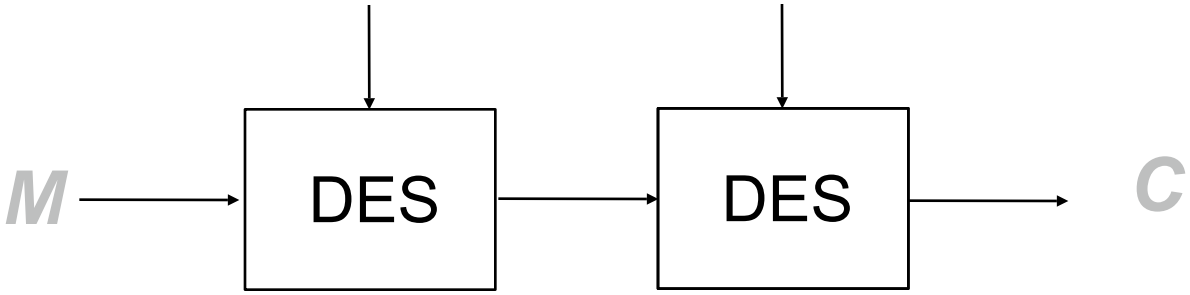
Attaque par le milieu

Étant donné un couple clair-chiffré (M,C):

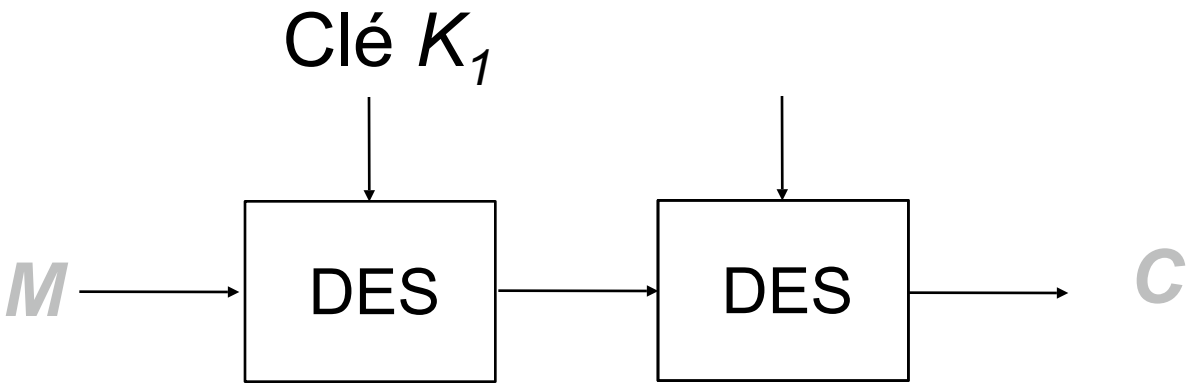
- calculer $N_i = DES_i (M)$ pour $0 \leq i < 2^{56}$ (*i.e.* pour chacune des 2^{56} valeurs possibles de K_1)
- calculer $P_j = DES^{-1}_j (C)$ pour $0 \leq j < 2^{56}$ (*i.e.* pour chacune des 2^{56} valeurs possibles de K_2)
- On cherche les indices (i,j) tels que

$$P_j = N_i$$

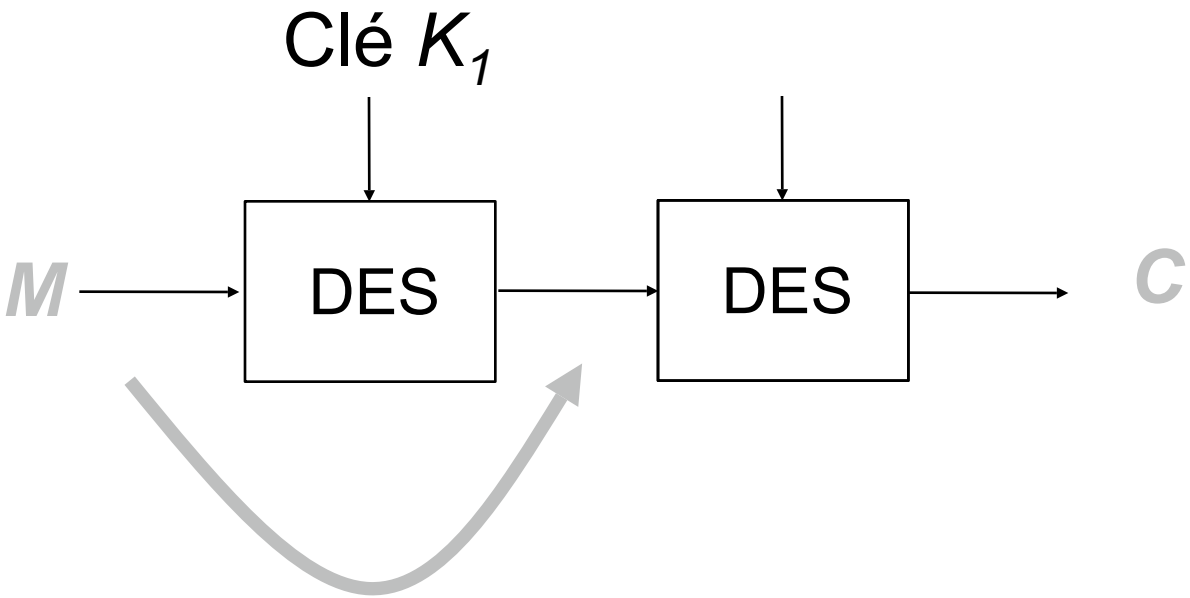
Attaque par le milieu



Attaque par le milieu



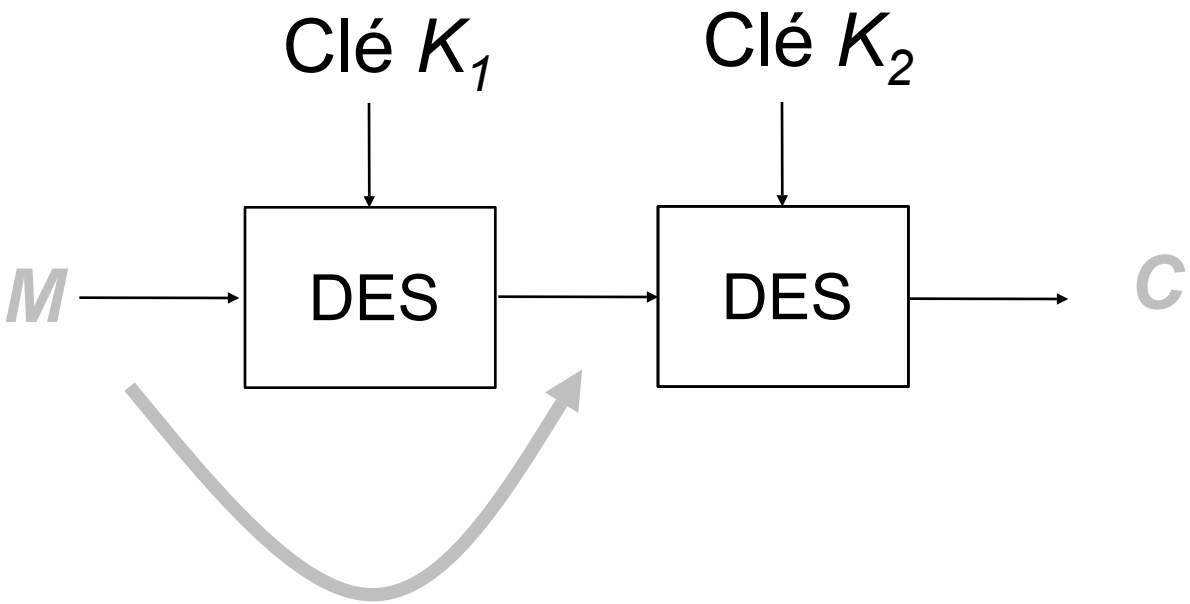
Attaque par le milieu



2^{56} calculs

$N_i = DES(i,M)$

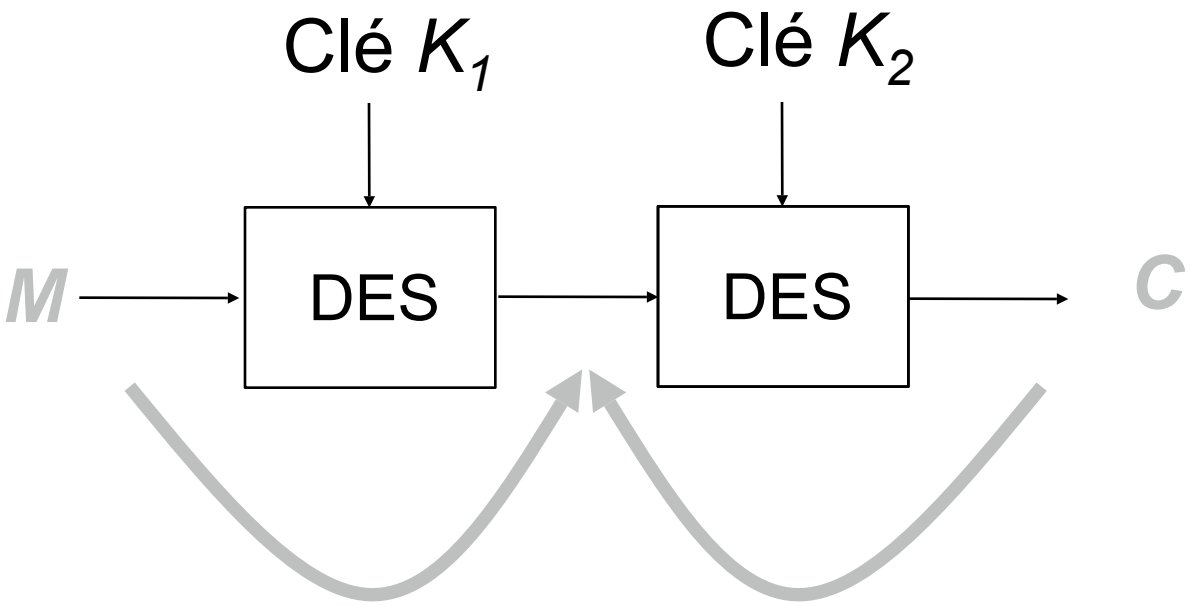
Attaque par le milieu



2^{56} calculs

$N_i = DES(i,M)$

Attaque par le milieu



2^{56} calculs
$N_i = DES(i,M)$

Pour chaque
$P_j = DES^{-1}(j,C)$,
on cherche $N_i = P_j$

Attaque par le milieu

On a :

- 2^{56} chiffrés N_i
- 2^{56} déchiffrés P_j

N_i et P_j font 64 bits, donc on a :

$(2^{56} \times 2^{56}) / 2^{64} = 2^{48}$ collisions en moyenne

Il existe donc 2^{48} couples (i,j) tels que $N_i = P_j$

Donc ***2^{48} bi-clés ($K_1 = i$, $K_2 = j$) possibles***

Attaque par le milieu

- On cherche toutes les collisions

$$N_i = P_j$$

et on obtient 2^{48} bi-clés possibles

- À l'aide d'un second couple (clair, chiffré) connu, on peut alors vérifier quel bi-clé (K_1,K_2) est le bon

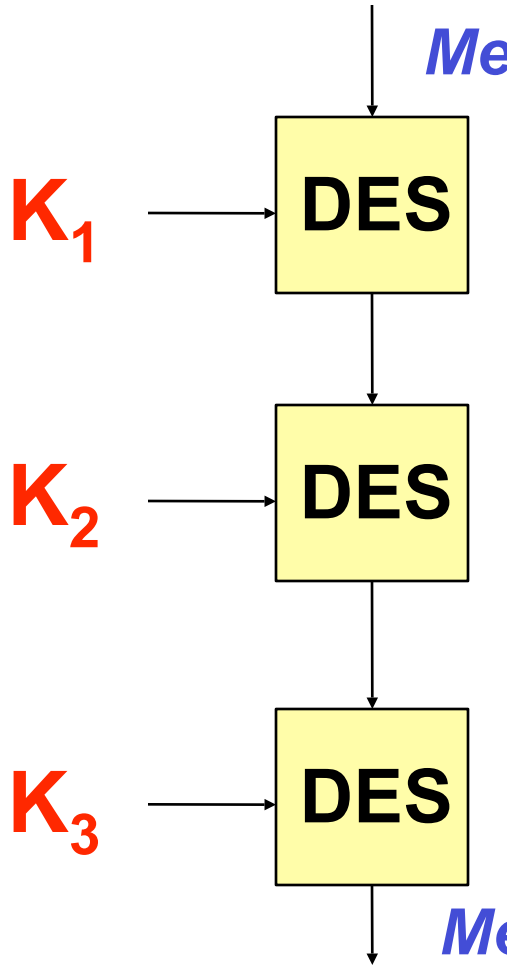
Complexité

- Attaque en 2^{56} en temps et 2^{56} couples (chiffré, clé) en mémoire

- **Par conséquent** :

  la sécurité du double DES n'atteint pas 2^{112} mais seulement 2^{56}, comme le DES

Triple-DES



Plusieurs variantes possibles :

- Utiliser DES ou DES^{-1} ?
- Triple-DES avec 3 clés ou 2 clés (K1 = K3) ?

Problème du Triple-DES

- Certaines variantes ne sont pas sûres (même type d'attaque que contre le Double-DES)
- Version recommandée par le NIST (FIPS 46-3)
  - Triple-DES avec 1,2 ou 3 clés différentes
  - EDE (Encryption-Decryption-Encryption)

Problèmes du Triple-DES

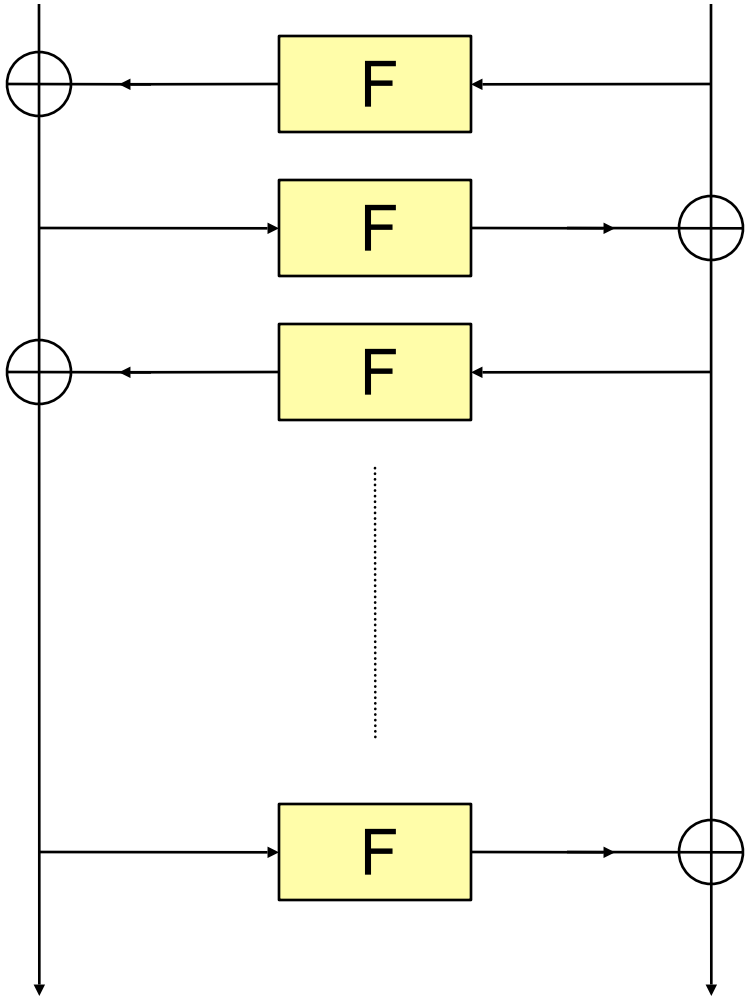
- Le Triple-DES permet d'éviter les problèmes liés à la taille de clé trop courte du DES
- Mais :
  - Le problème de la taille du bloc subsiste
  - Le Triple-DES n'est pas très rapide

⇒ Migration vers un algorithme plus récent. Quelles sont les autres solutions ?

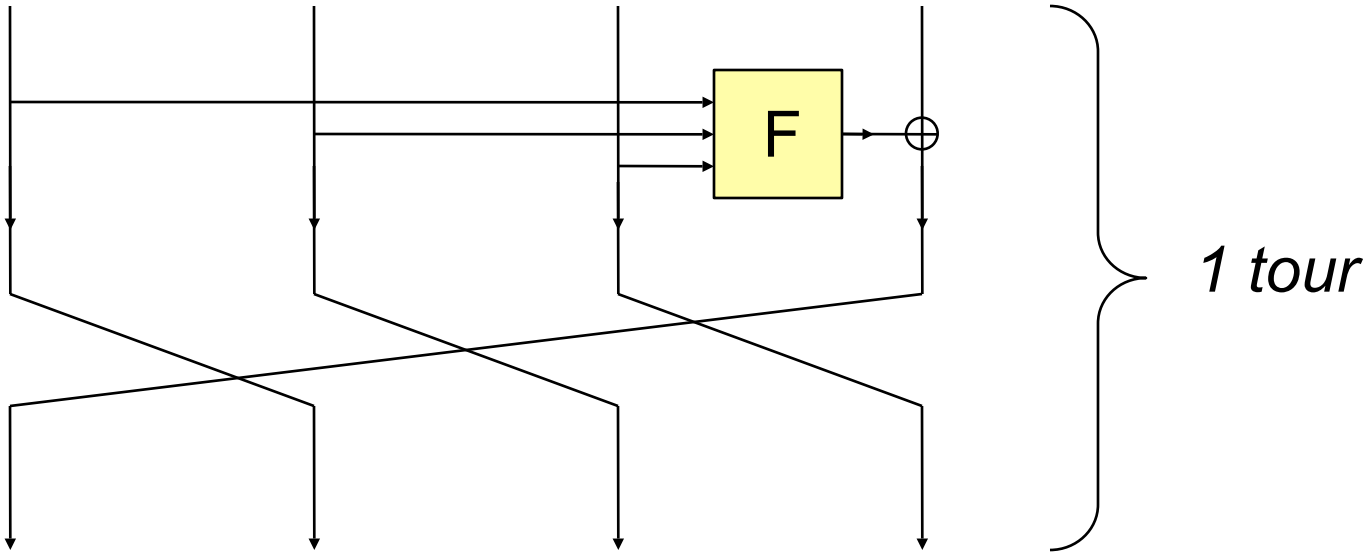
Constructions classiques

- Schéma de Feistel (cas du DES)
- Variations du schéma de Feistel :
  - Schéma de Feistel généralisé (ex : RC6)
  - Schéma de Lai-Massey (ex : IDEA)
- Réseau SP (Substitution-Permutation)
  - Exemple de l'AES

Feistel

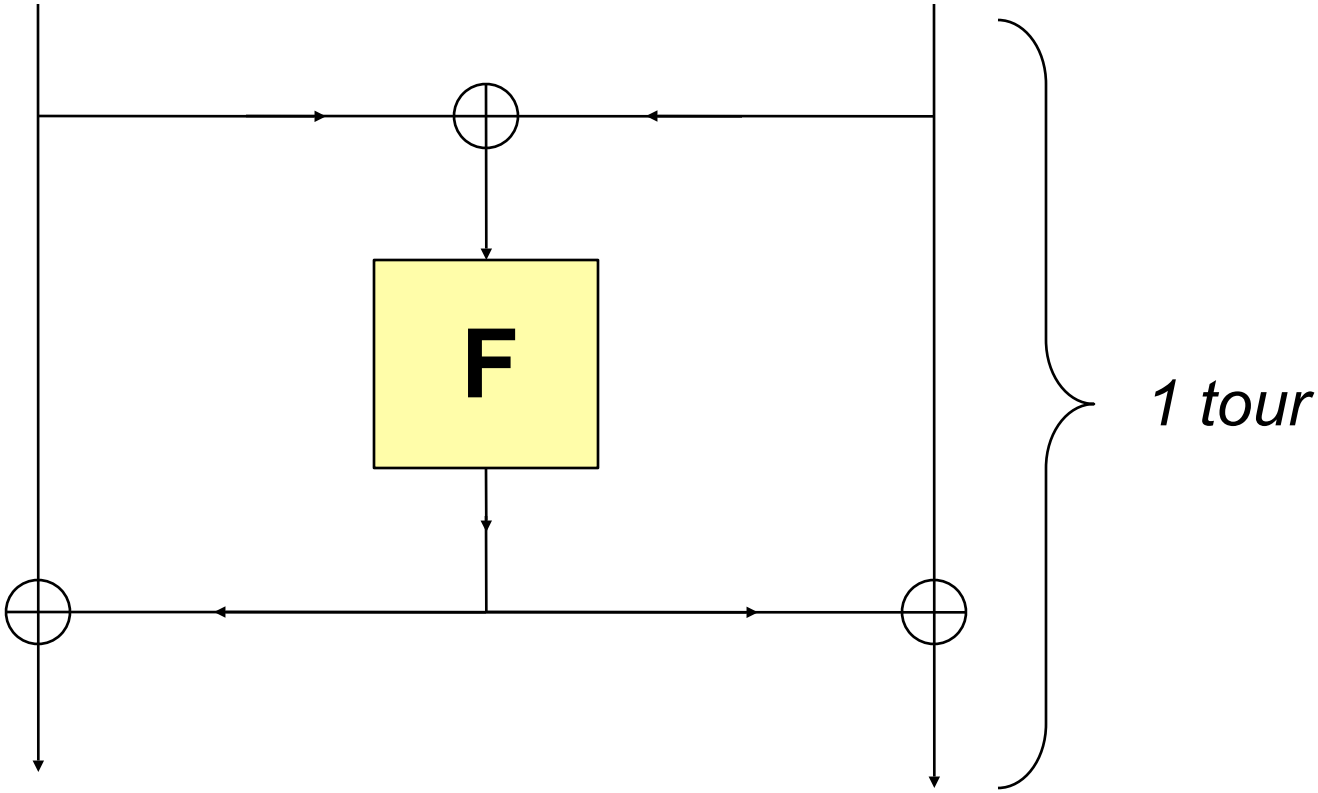


Feistel généralisés



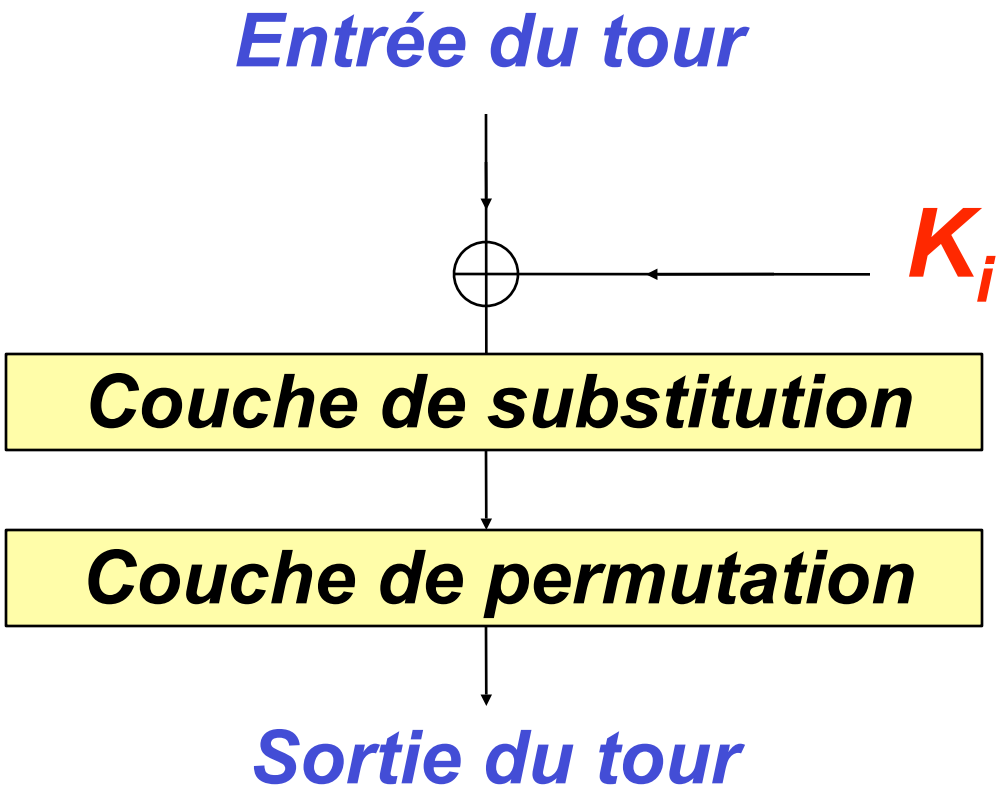
Structure inversible pour toute fonction F

Lai-Massey



Structure inversible pour toute fonction F

Réseau SP



Toutes les couches doivent être inversibles !

AES

- Nouveau standard américain (NIST, 2000), remplaçant du DES
- Processus de sélection (1997-2000) :
  - 15 candidats initiaux
  - 5 retenus pour le second tour
    - Rijndael (Daemen-Rijmen, Belgique)
    - MARS (IBM, USA)
    - SERPENT (Biham-Knudsen-Anderson)
    - RC6 (RSA Labs)
    - Twofish (USA)

AES

- Le 2 octobre 2000, l'algorithme belge Rijndael est retenu par le NIST
- FIPS 197
- Taille de bloc de 128 bits
- Tailles de clé de 128, 192 et 256 bits

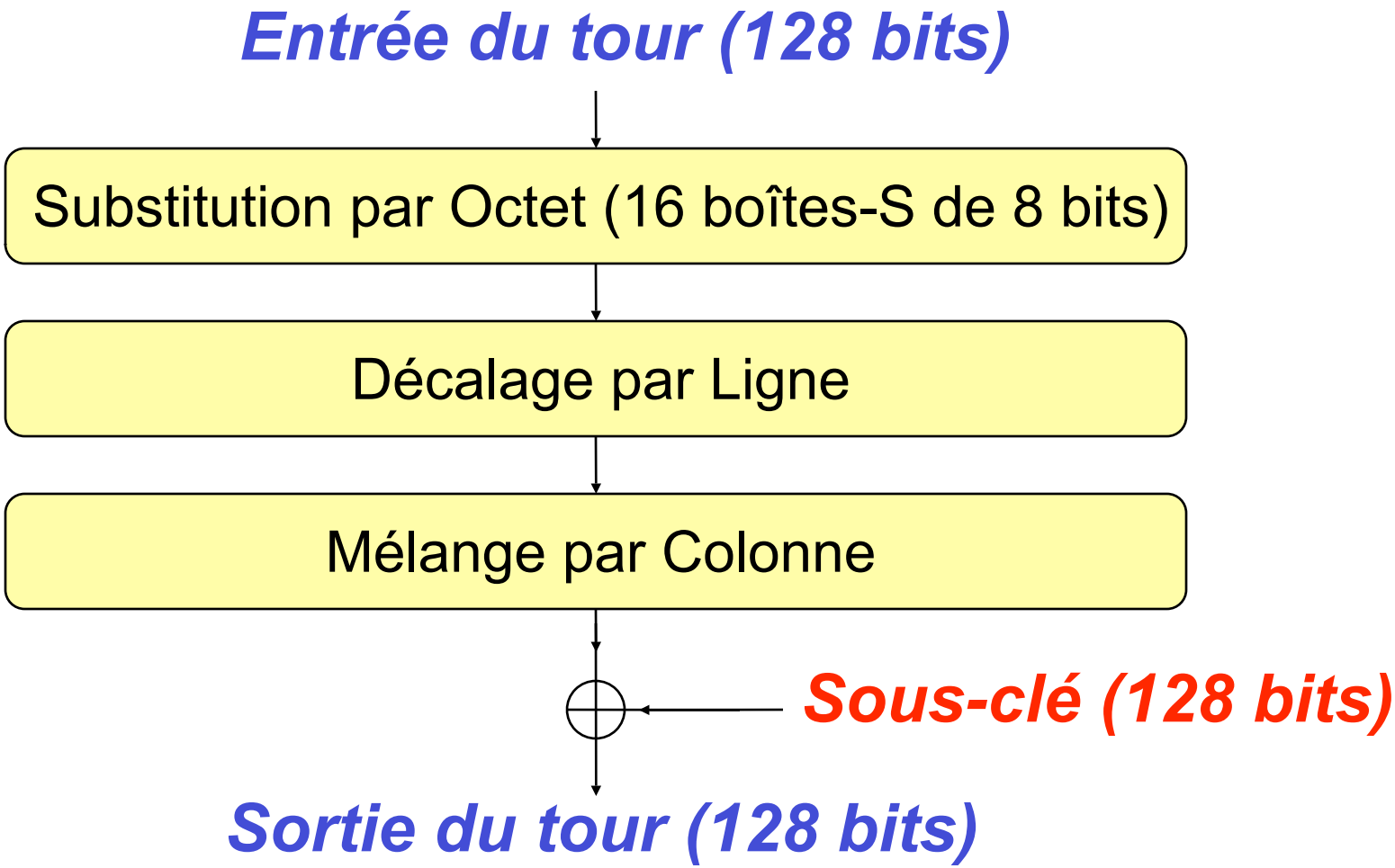
Structure générale

Les données sont stockées dans un « carré »
de 4 x 4 = 16 cases

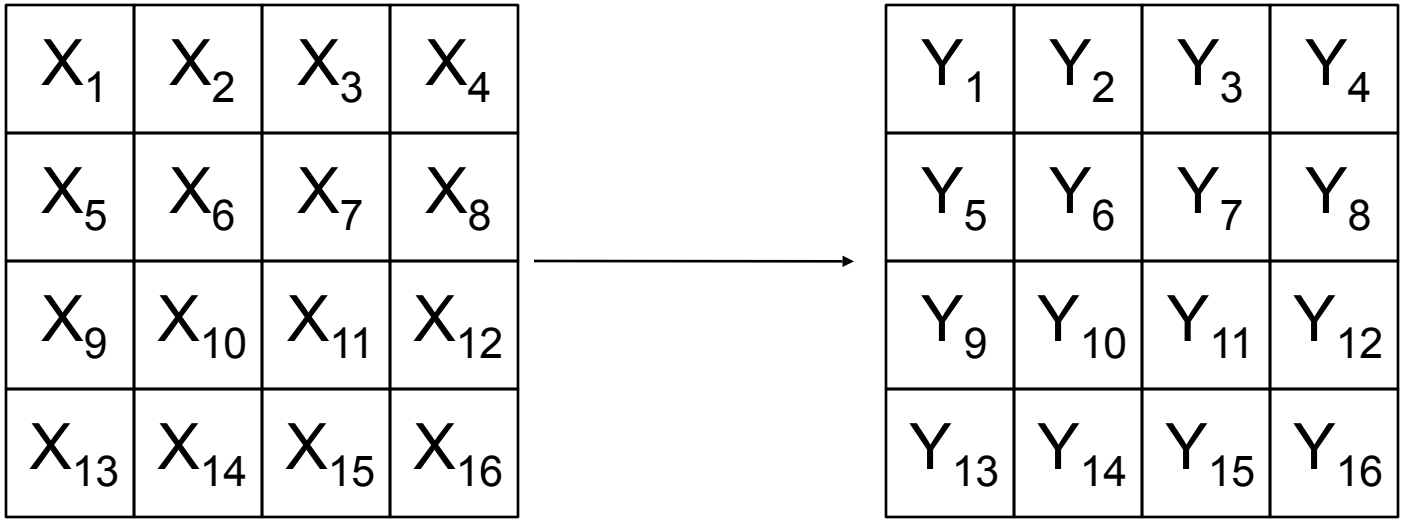
X_1	X_2	X_3	X_4
X_5	X_6	X_7	X_8
X_9	X_{10}	X_{11}	X_{12}
X_{13}	X_{14}	X_{15}	X_{16}

Chaque case contient 1 octet
(8 x 16 = 128 bits d'état interne)

Fonction de tour



Substitution par Octet



Pour tout $1 \leq i \leq 16$, $Y_i = S(X_i)$

Substitution par Octet

- S est une fonction fixe de 8 bits vers 8 bits
  - Définie comme un tableau à $2^8 = 256$ entrées
  - Nécessite donc 256 octets de mémoire
- Basée sur une opération algébrique :

$$S(X) = \text{Affine}(\text{Inverse}(X))$$

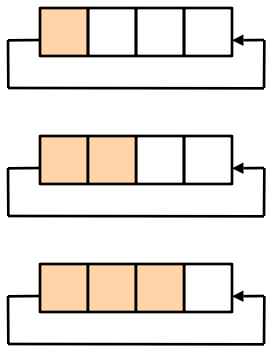
où l'inverse est pris dans $GF(2^8)$

La boîte S

$$Sbox = \begin{bmatrix}
99 & 124 & 119 & 123 & 242 & 107 & 111 & 197 & 48 & 1 & 103 & 43 & 254 & 215 & 171 & 118 \\
202 & 130 & 201 & 125 & 250 & 89 & 71 & 240 & 173 & 212 & 162 & 175 & 156 & 164 & 114 & 192 \\
183 & 253 & 147 & 38 & 54 & 63 & 247 & 204 & 52 & 165 & 229 & 241 & 113 & 216 & 49 & 21 \\
4 & 199 & 35 & 195 & 24 & 150 & 5 & 154 & 7 & 18 & 128 & 226 & 235 & 39 & 178 & 117 \\
9 & 131 & 44 & 26 & 27 & 110 & 90 & 160 & 82 & 59 & 214 & 179 & 41 & 227 & 47 & 132 \\
83 & 209 & 0 & 237 & 32 & 252 & 177 & 91 & 106 & 203 & 190 & 57 & 74 & 76 & 88 & 207 \\
208 & 239 & 170 & 251 & 67 & 77 & 51 & 133 & 69 & 249 & 2 & 127 & 80 & 60 & 159 & 168 \\
81 & 163 & 64 & 143 & 146 & 157 & 56 & 245 & 188 & 182 & 218 & 33 & 16 & 255 & 243 & 210 \\
205 & 12 & 19 & 236 & 95 & 151 & 68 & 23 & 196 & 167 & 126 & 61 & 100 & 93 & 25 & 115 \\
96 & 129 & 79 & 220 & 34 & 42 & 144 & 136 & 70 & 238 & 184 & 20 & 222 & 94 & 11 & 219 \\
224 & 50 & 58 & 10 & 73 & 6 & 36 & 92 & 194 & 211 & 172 & 98 & 145 & 149 & 228 & 121 \\
231 & 200 & 55 & 109 & 141 & 213 & 78 & 169 & 108 & 86 & 244 & 234 & 101 & 122 & 174 & 8 \\
186 & 120 & 37 & 46 & 28 & 166 & 180 & 198 & 232 & 221 & 116 & 31 & 75 & 189 & 139 & 138 \\
112 & 62 & 181 & 102 & 72 & 3 & 246 & 14 & 97 & 53 & 87 & 185 & 134 & 193 & 29 & 158 \\
225 & 248 & 152 & 17 & 105 & 217 & 142 & 148 & 155 & 30 & 135 & 233 & 206 & 85 & 40 & 223 \\
140 & 161 & 137 & 13 & 191 & 230 & 66 & 104 & 65 & 153 & 45 & 15 & 176 & 84 & 187 & 22
\end{bmatrix}$$

Décalage par ligne

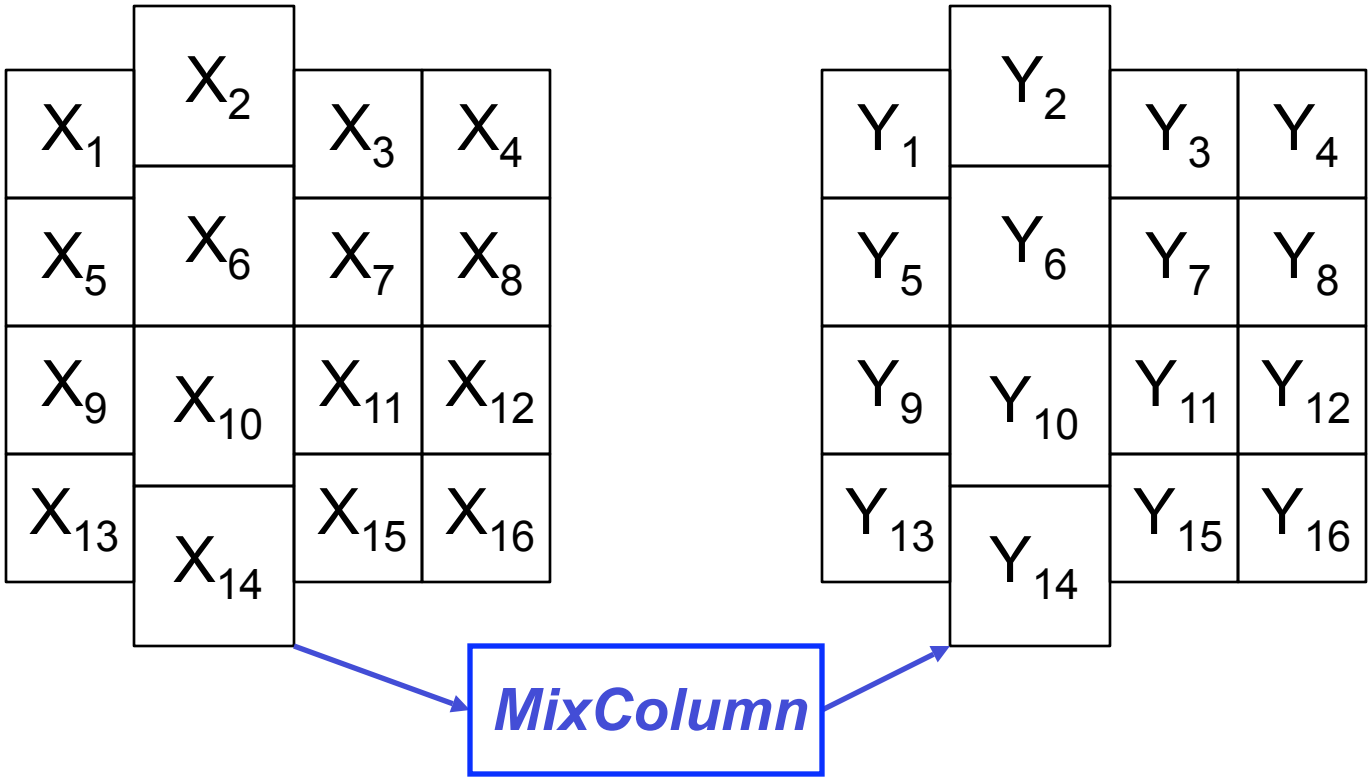
X_1	X_2	X_3	X_4
X_5	X_6	X_7	X_8
X_9	X_{10}	X_{11}	X_{12}
X_{13}	X_{14}	X_{15}	X_{16}



X_1	X_2	X_3	X_4
X_6	X_7	X_8	X_5
X_{11}	X_{12}	X_9	X_{10}
X_{16}	X_{13}	X_{14}	X_{15}

Décalage circulaire (vers la gauche) de i cases pour la ligne numéro i, $0 \leq i \leq 3$

Mélange par colonne



MixColumn() est appliquée à chaque colonne

Mélange par colonne

$$\textit{MixColumn}\left(\begin{bmatrix} X_1 \\ X_5 \\ X_9 \\ X_{13} \end{bmatrix}\right) = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \textit{ x } \begin{bmatrix} X_1 \\ X_5 \\ X_9 \\ X_{13} \end{bmatrix}$$

Opérations linéaires dans GF(2^8)

Corps finis : GF(2^8)

- Cet objet mathématique est utilisé pour définir la boîte S et dans MixColumn()

- Unique corps fini à 256 éléments
  - Addition = XOR
  - Multiplication = ?

GF(2^8) : Définition

- GF(2) = F_2 = unique corps fini à 2 éléments

  = {0,1} avec les opérations booléennes usuelles
- $F_2[X]$ = ensemble des polynômes à coefficients dans F_2
- Soit P(X) un polynôme irréductible de degré 8 appartenant à $F_2[X]$
- Par définition, GF(2^8) = $F_2[X]$ / P

GF(2^8) : Exemple

- Prenons $P(X) = X^8 + X^4 + X^3 + X + 1$
- Éléments dans GF(2^8)

  = polynômes réduits « modulo P »

  = polynômes de degré < 8
- Exemple

$$a = X^6 + X^4 + X^2$$

GF(2^8) : Exemple

- Soient a et b dans GF(2^8)

  $a = X^6 + X^4 + X^2$

  $b = X^2$

- Addition :

  $a+b = (X^6 + X^4 + X^2) + (X^2) = X^6 + X^4$

- Multiplication :

  $a \times b = (X^8 + X^6 + X^4)$ modulo $(X^8 + X^4 + X^3 + X + 1)$

  $a \times b = (X^6 + X^4) + X^4 + X^3 + X + 1$

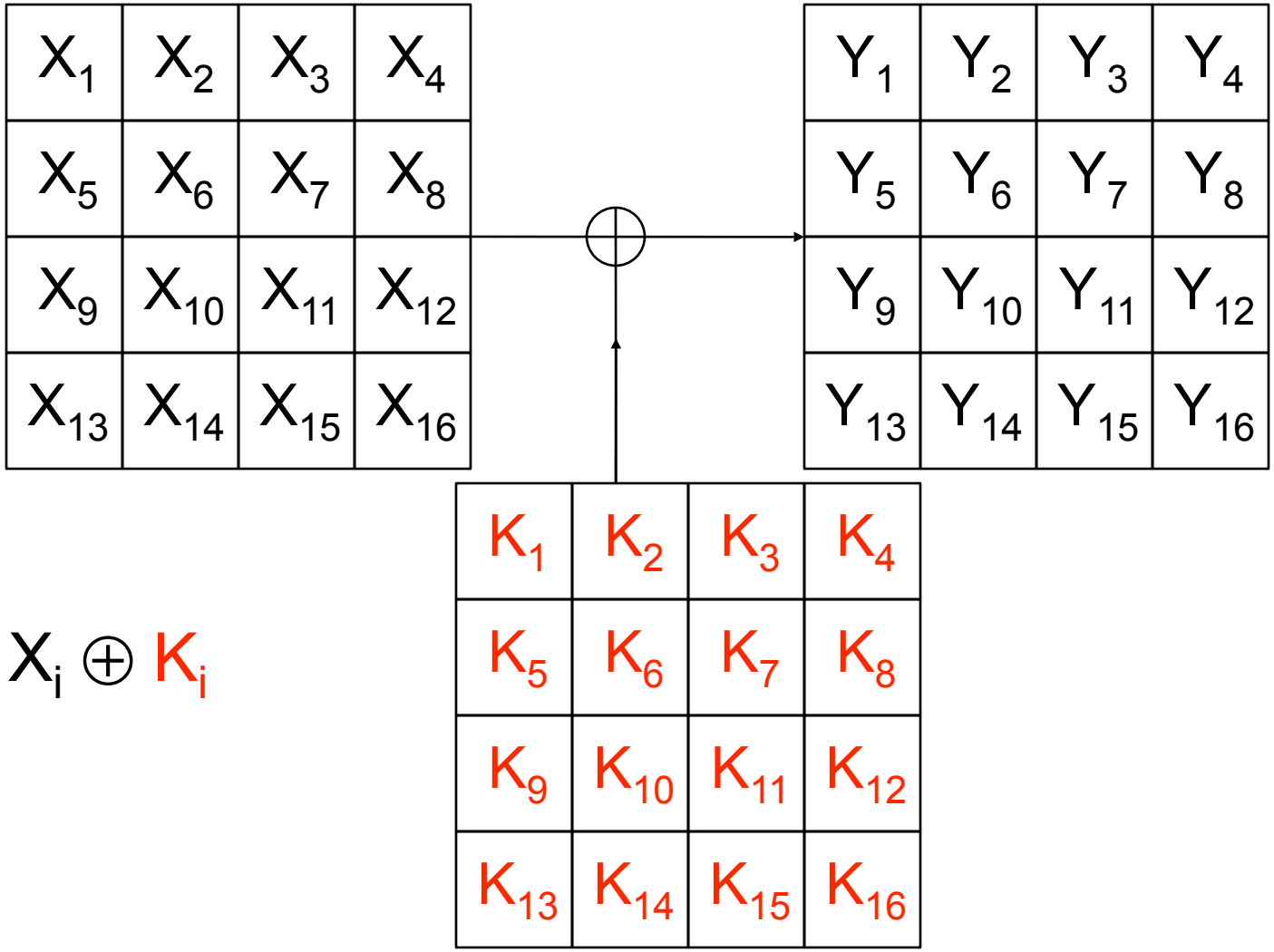
GF(2^8) : représentation

- Chaque élément de GF(2^8) est représenté

$$b_7X^7 + b_6X^6 + \ldots + b_1X + b_0$$

- On le stocke sur l'octet représenté en binaire par (b_7,…,b_0)
- Représentation entre 0 et 255

Addition de la sous-clé



$Y_i = X_i \oplus K_i$

«Key Schedule»

- Algorithme de dérivation des sous-clés à partir de la clé secrète

- Basé sur les mêmes primitives que la fonction de tour

Nombre de tours

- Pour AES-128 (clé de taille 128 bits)

  t = 10 tours

- Pour AES-192

  t = 12 tours

- Pour AES-256

  t = 14 tours

Sécurité de l'AES

- L'algorithme est encore jeune mais
- Il a été conçu pour résister aux attaques classiques (différentielle, linéaire, …)

  → Inversion dans GF(2^8)
- Attaques contre des version réduites (à 6 ou 7 tours) ?
- Attaques algébriques en utilisant la structure mathématique simple de l'AES ?

Synthèse

- Un algorithme de chiffrement par bloc est une **primitive de base** («brique»)
- Reste à se poser la question de son **utilisation** !

  Exemple : modes opératoires pour chiffrer des messages de taille arbitraire

En pratique

- Algorithmes utilisés
  - DES dans les anciens produits
  - AES dans les nouveaux produits
- Autres algorithmes utilisés ponctuellement
  - IDEA (PGP)
  - BlowFish
  - …