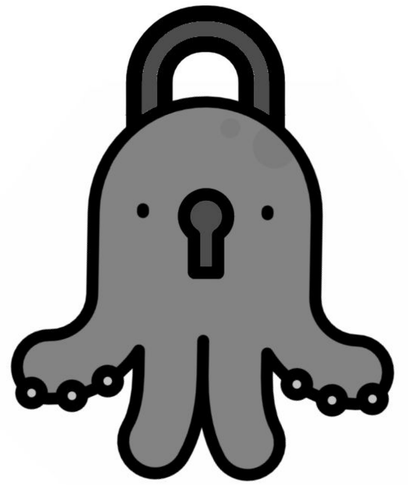
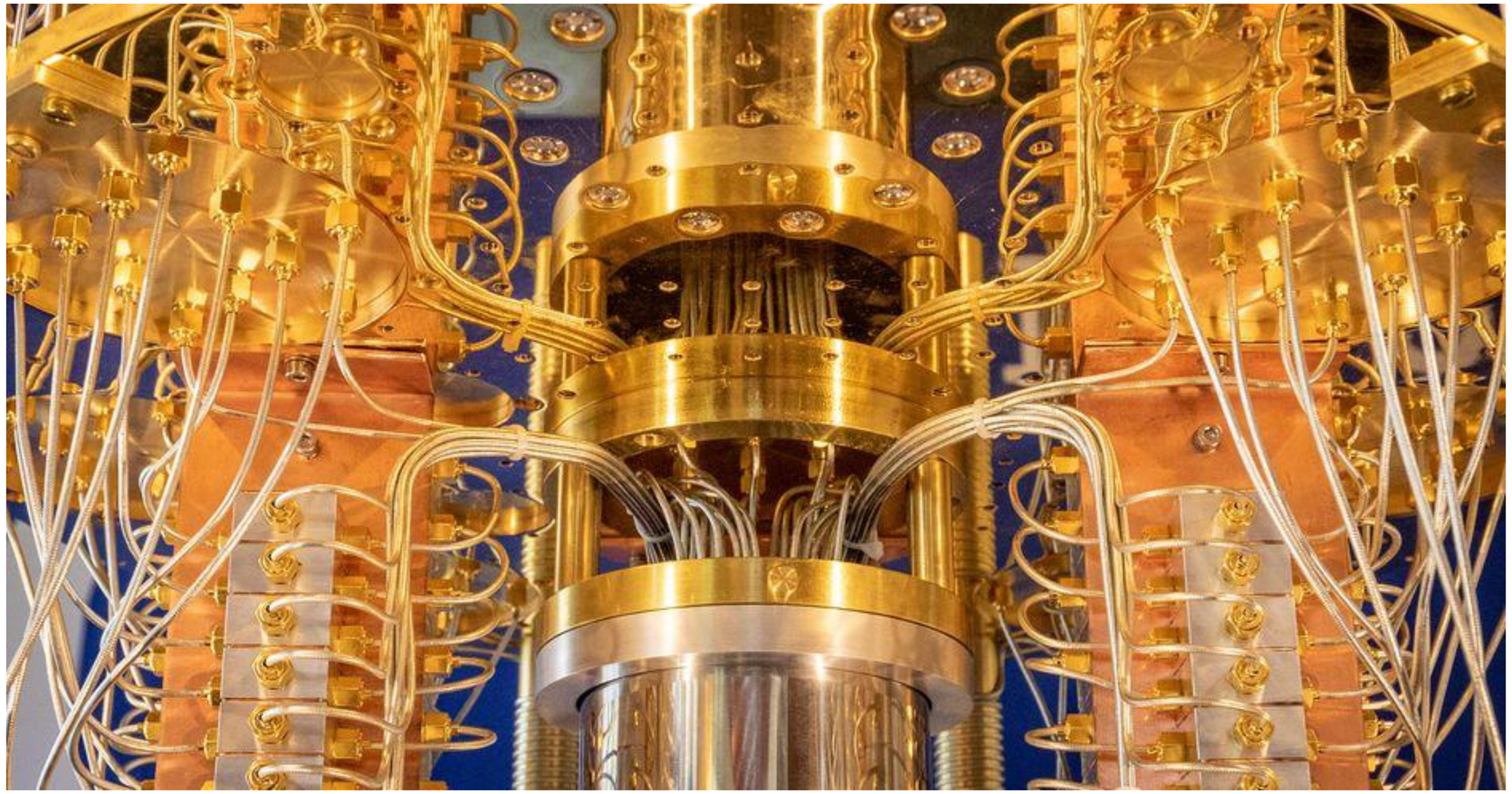
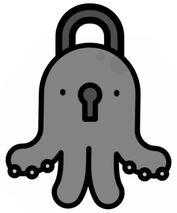
# Chiffrement par Bloc

IFT 3275



**“Computational security”: Bien que la sécurité inconditionnelle n’offre aucune information par rapport au message chiffré, nous utilisons en pratique des cryptosystèmes qui offrent une sécurité qui n’est pas inconditionnelle, mais qui offre une quantité minime d’information à Eve qui possède une puissance de calcul bornée.**





# Chiffrement par bloc

Les chiffrements symmétriques utilisés de nos jours (3DES, AES) sont basés sur le chiffrement par bloc. Voici la définition formelle d'un chiffrage par bloc d'un message $m$ de longueur $n$ utilisant une clef $k$ :

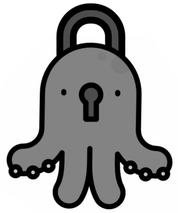
$$ENC_k(m) : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$$

Une encryption $ENC_k(m)$ n'est qu'une bijection des bits d'entrée $\{0,1\}^n$. Chaque clef $k$ produit une des $(2^n)!$ permutations possibles. Il y a trois types d'attaques possibles concernant les chiffrements par bloc :

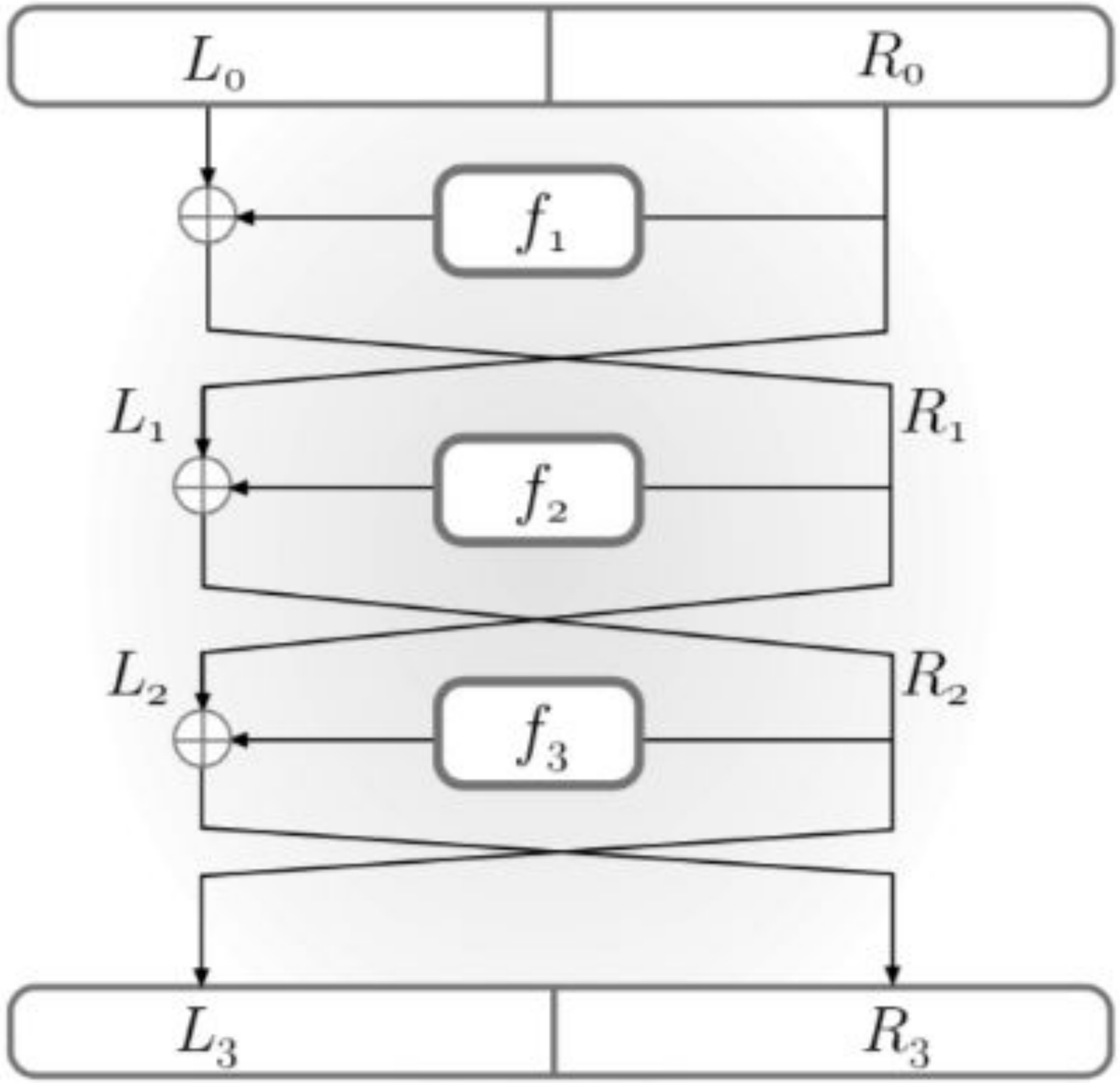
1. **Attaque à texte clair connu :** où l'attaquant possède des paires de messages et chiffres $\{m, ENC_k(m)\}$ où les messages **ne peuvent pas être choisis par l'attaquant**.

2. **Attaque à texte clair choisi :** où l'attaquant possède des paires de messages et chiffres $\{m, ENC_k(m)\}$ où les messages **sont choisis par l'attaquant**.

3. **Attaque à texte chiffré choisi :** où l'attaquant possède $ENC_k(m)$ pour un message $m$ choisi, ainsi que $DEC_k(c)$ pour un chiffre $c$ choisi.

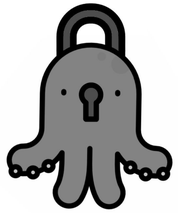


# Réseau de Feisteil:




$$L_1 = R_0$$

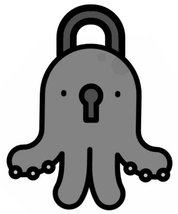
$$R_1 = L_0 \oplus F(R_0, K_0)$$



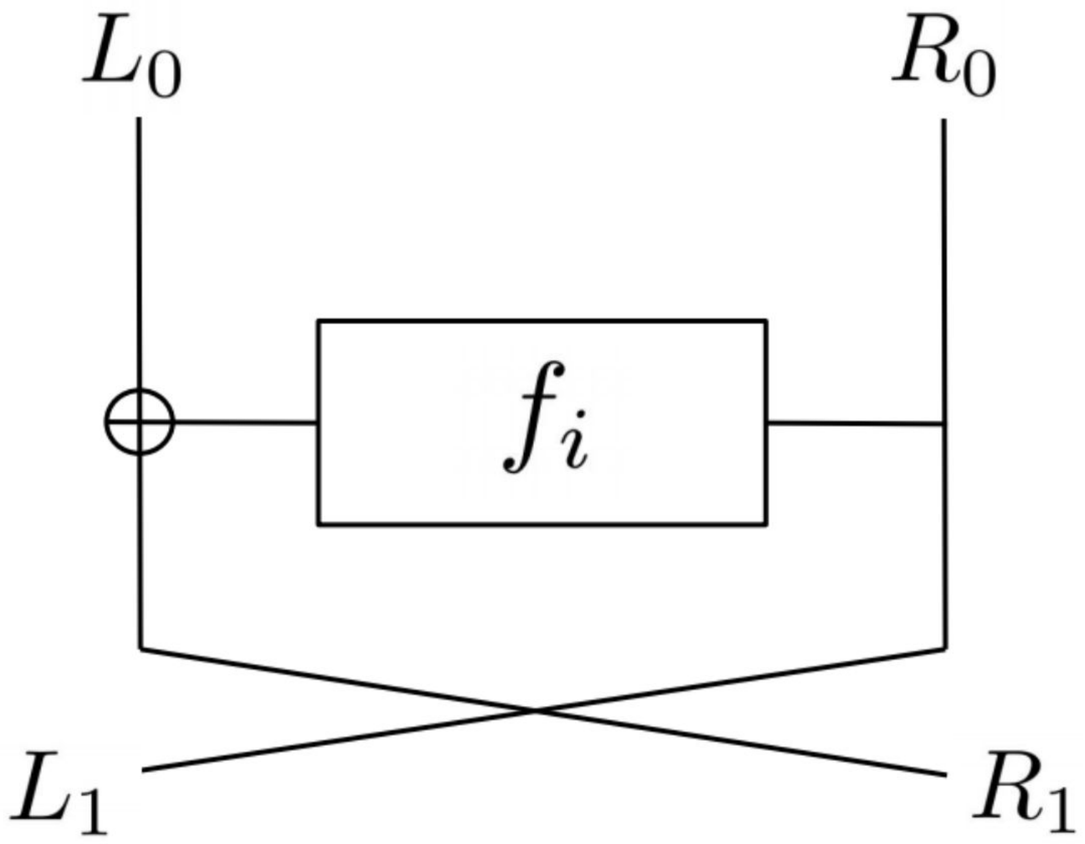
## Paradigme Confusion-Diffusion:

- **Confusion: Chaque bit de notre chiffre doit dépendre de plusieurs parties de la clef pour pouvoir dissimuler les connexions entre notre chiffre et la clef utilisée.**

- **Diffusion: Si nous changeons un bit du message, plusieurs bits du chiffre doivent changer une fois ce nouveau message chiffré. Similairement, si nous changeons un bit du chiffre, plusieurs bits du message doivent changer une fois ce nouveau chiffre déchiffré, Un chiffrement qui ne possède absolument pas cette propriété est la substitution mono-alphabétique!**

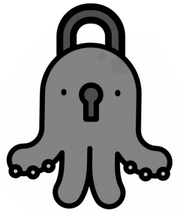


# Inversion du réseau de Feistel:




$$R_0 = L_1$$

$$L_0 = R_1 \oplus f_i(R_0, k_i)$$



# DES / Data Encryption Standard:

- Introduit en 1976.

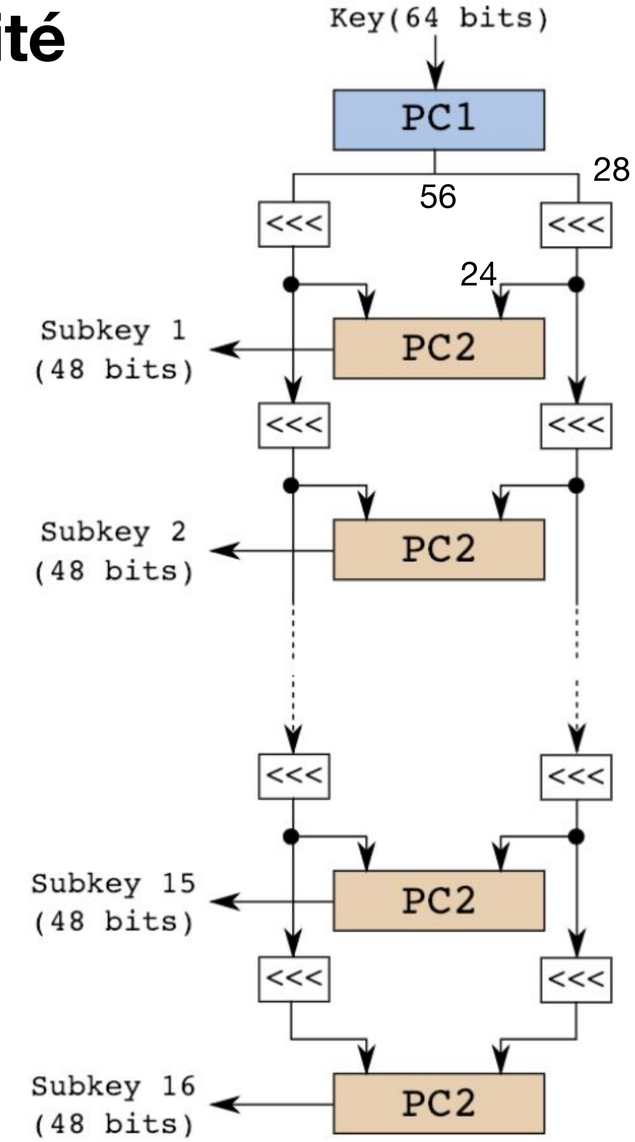
- Est désuet depuis les années 1990.


# Protocole de génération de clefs:

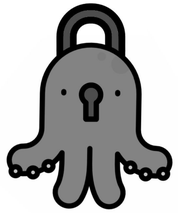
- **64 bits - 8 bits de parité = 56 bits de sécurité**

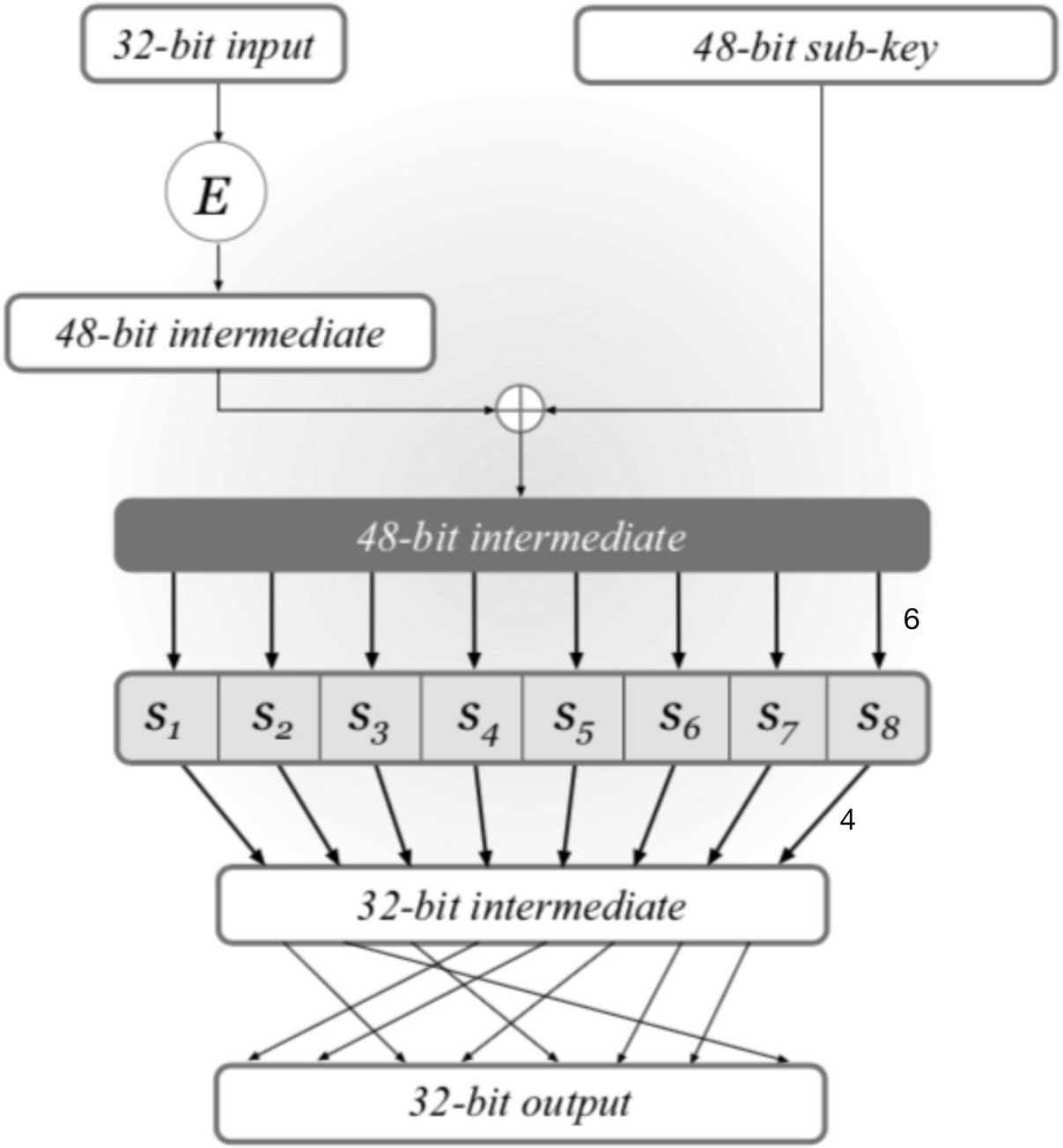
- **2 opérations: Permutation (PC1 et PC2) et décalages circulaires vers la gauche**

| Iteration Number | Number of Left Shifts |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 2 |
| 5 | 2 |
| 6 | 2 |
| 7 | 2 |
| 8 | 2 |
| 9 | 1 |
| 10 | 2 |
| 11 | 2 |
| 12 | 2 |
| 13 | 2 |
| 14 | 2 |
| 15 | 2 |
| 16 | 1 |




$$b_0 b_1 b_2 \ldots b_{25} b_{26} b_{27} \longrightarrow b_1 b_2 \ldots b_{25} b_{26} b_{27} b_0$$

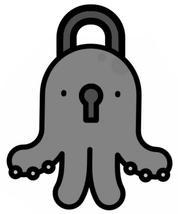



32-bit input
48-bit sub-key
E
48-bit intermediate
48-bit intermediate
6
S1
S2
S3
S4
S5
S6
S7
S8
4
32-bit intermediate
32-bit output

## Boîtes à substitution de DES:

1. Chaque ligne du tableau contient chaque possibilité d'output.
2. Changer un bit dans un input change au moins 2 bits dans l'ouput.

| $S_1$ | x0000x | x0001x | x0010x | x0011x | x0100x | x0101x | x0110x | x0111x | x1000x | x1001x | x1010x | x1011x | x1100x | x1101x | x1110x | x1111x |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0yyyy0 | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 0yyyy1 | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 1yyyy0 | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 1yyyy1 | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |



DES → faiblesse ⇒ fouille exhaustive

2DES ⇒ $ENC_{K_1,K_2}(m) = ENC_{K_2}(ENC_{K_1}(m))$

Pourrait penser $|\mathcal{K}| = 2^{2 \cdot 56} = 2^{112}$ → 112 bits de sécurité ✗

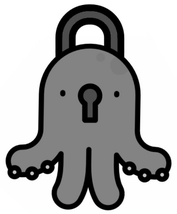
Meet in the middle : MITM $2^1 \cdot 2^{56} = 2^{56+1} = 2^{57}$ → 57 bits de sécurité

→ 1. $\forall K_1 \in \{0,1\}^{56}$, calculer $z = ENC_{K_1}(m)$ et ajouter $(z, K_1)$ à $L_1$

2. $\forall K_2 \in \{0,1\}^{56}$, calculer $z' = DEC_{K_2}(c)$ et ajouter $(z', K_2)$ à $L_2$

3. $\exists (z, K_1) \in L_1 \wedge \exists (z', K_2) \in L_2$ t.q. $z = z'$ ⇒ TRÈS PROBABLE

ON CONNAÎT $K_1$ & $K_2$ !



# 3DES : (TRIPLE DES) :

2DES : $2 \cdot 2^{56} = 2^{56+1} = 2^{57}$ → 57 bits de sécurité

$c = ENC_{K_3}(DEC_{K_2}(ENC_{K_1}(m)))$

$m = DEC_{K_1}(ENC_{K_2}(DEC_{K_3}(c)))$

$2^{56} + 2^{56}$

$2 \cdot 2^{56} = 2^{56+1} = 2^{57}$

## MITM ATTACK :

1 : $\forall K_1 \in \{0,1\}^{56}$, calculer $z = ENC_{K_1}(m)$ et ajouter $(z, K_1)$ à $L_1$

2 : $\forall k_2, k_3 \in \{0,1\}^{56}$, calculer $z' = ENC_{K_2}(DEC_{K_3}(c))$ et ajouter $(z', K_2, K_3)$ à $L_2$

3 : $\exists (z, K_1) \in L_1 \wedge (z', K_2, K_3) \in L_2$ t.q. $z = z'$

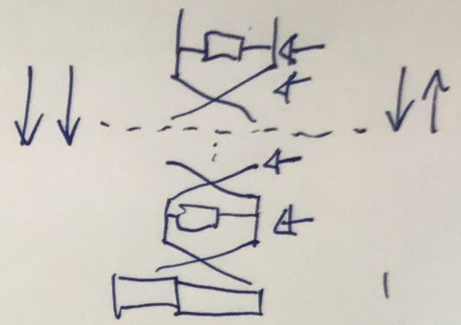
⇒ très probable qu'on a $K_1, K_2, K_3$

# Bits de sécurité : $2^{56} \cdot 2^{56}$

$2^{56} + 2^{56} \cdot 2^{56} = 2^{56} + 2^{112} \approx 2^{112}$

$2^{56}$ inconsidérable

CLEFS FAIBLES : 4 clefs



$ENC_K(ENC_K(m)) = m$ ⇐ Déjà vu ?

- 01010101 01010101
- fefe
- 1f
- e0

Protocol de génération de clef :

$k_{1+i} = K_{16-i}$ $0 \leq i \leq 7$

1
2
15
16

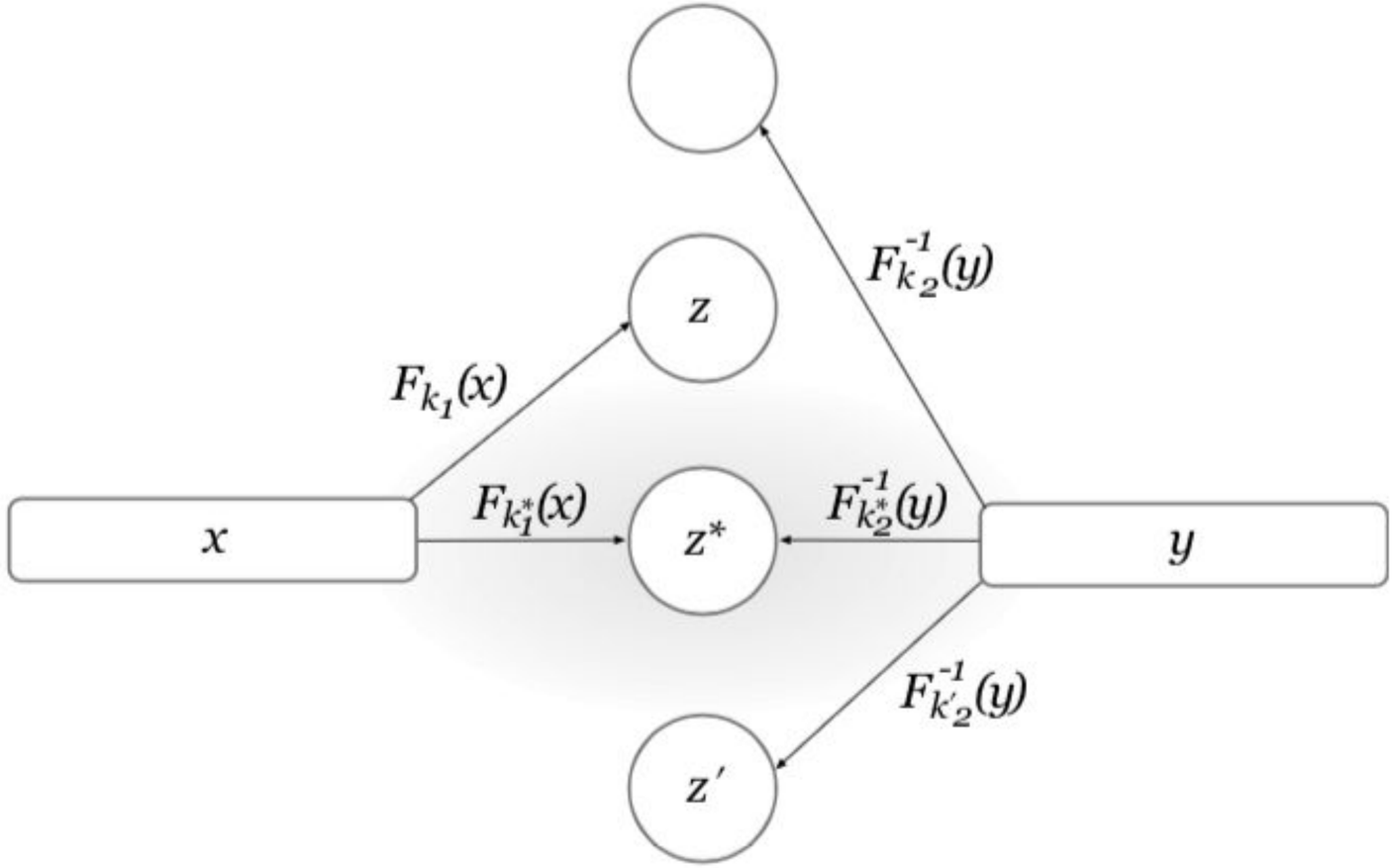
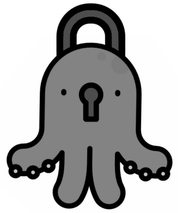



**FIGURE 6.7**: A meet-in-the-middle attack.



# Modes d’opération:

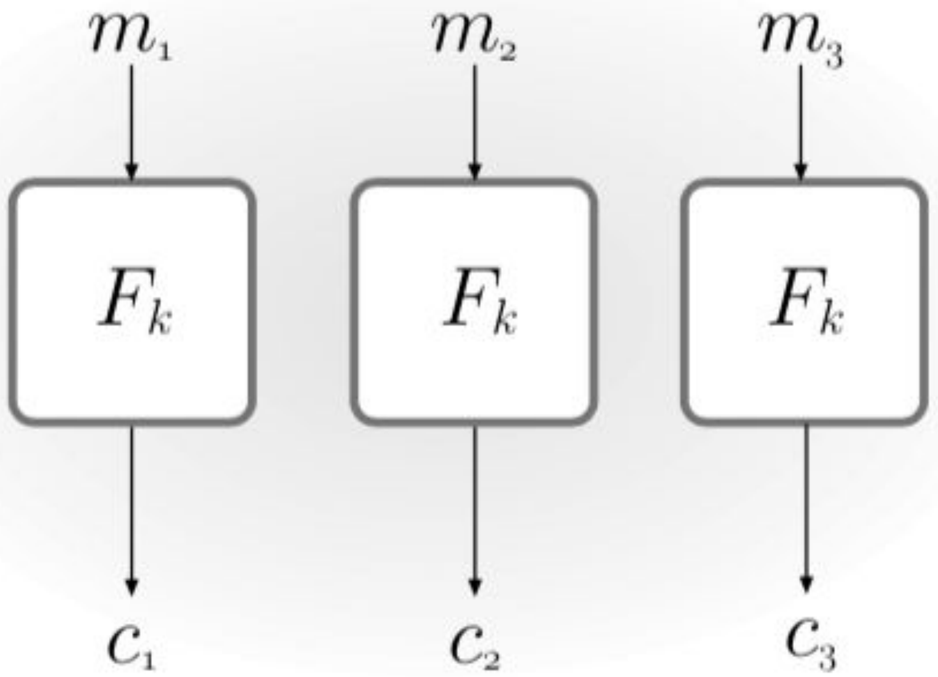
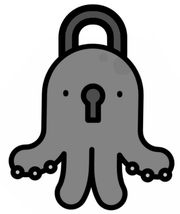
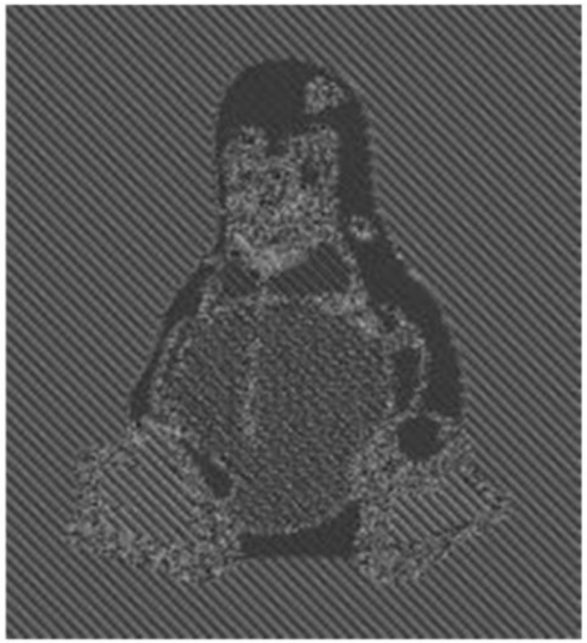
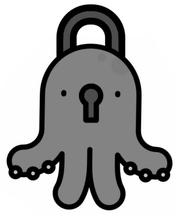



**FIGURE 3.5**: Electronic Code Book (ECB) mode.







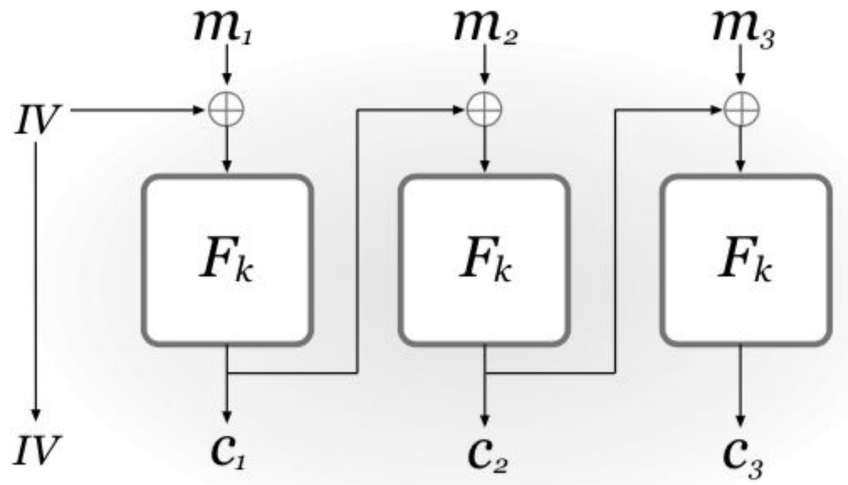



**FIGURE 3.7**: Cipher Block Chaining (CBC) mode.

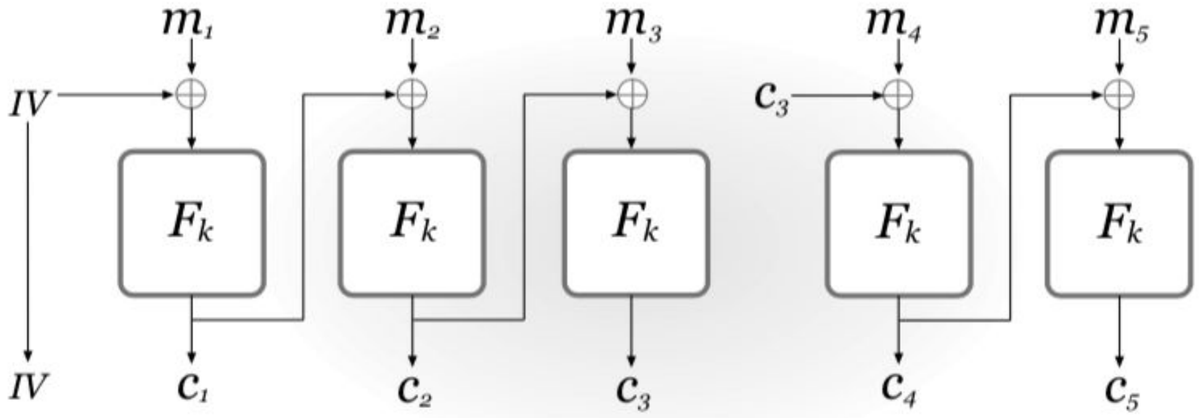
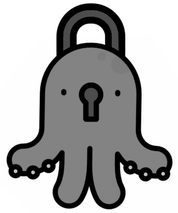



**FIGURE 3.8**: Chained CBC.



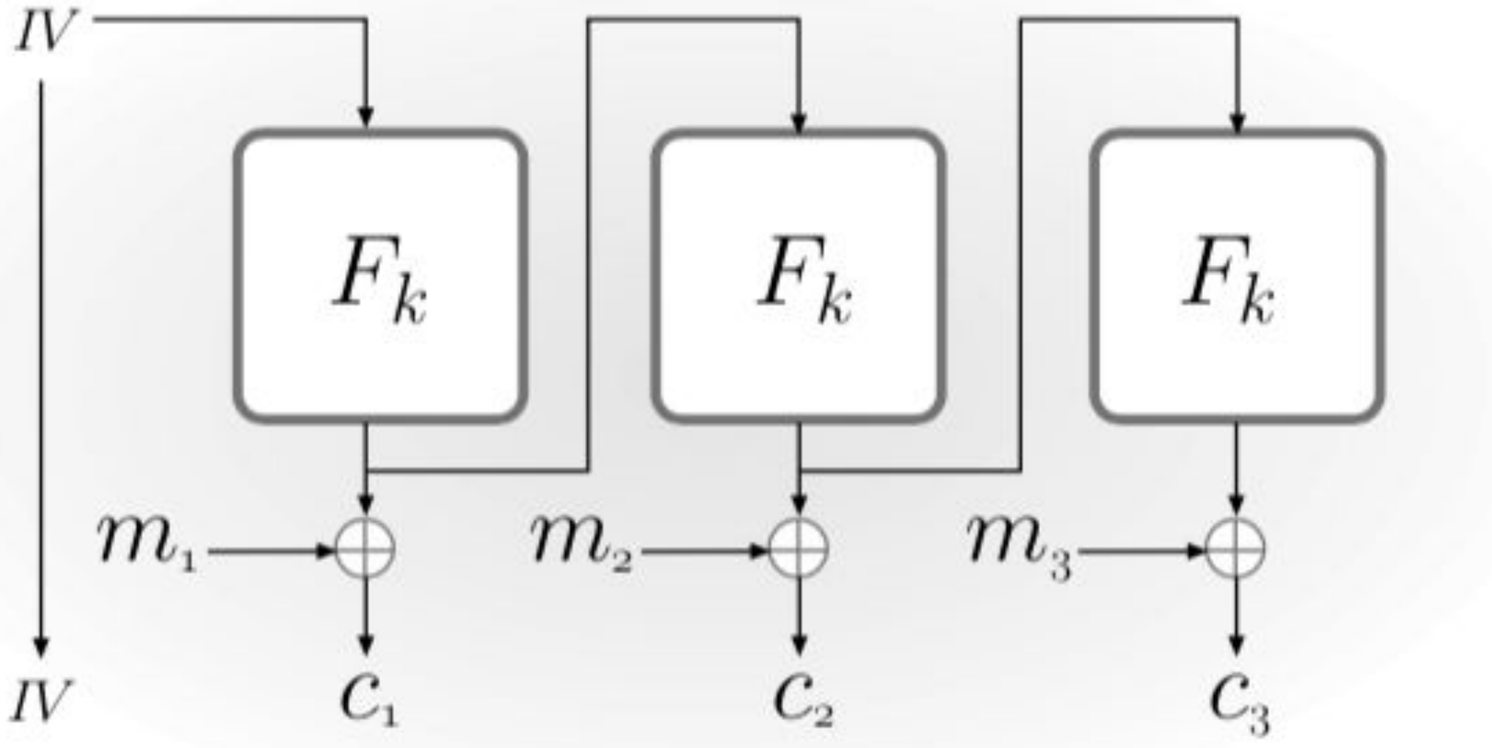



**FIGURE 3.9**: Output Feedback (OFB) mode.

# Réseaux de substitution-permutation:

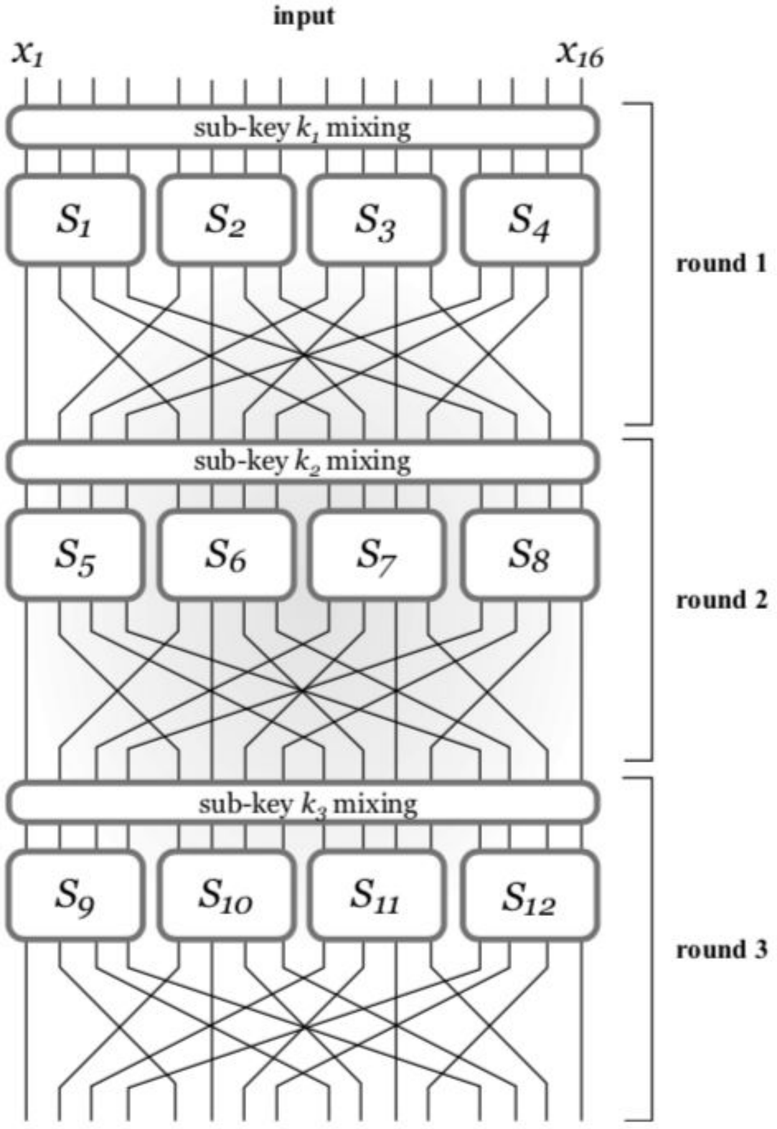



**FIGURE 6.4**: A substitution-permutation network.