

# **Chiffrement par bloc : Cryptanalyse**

Pierre-Alain Fouque

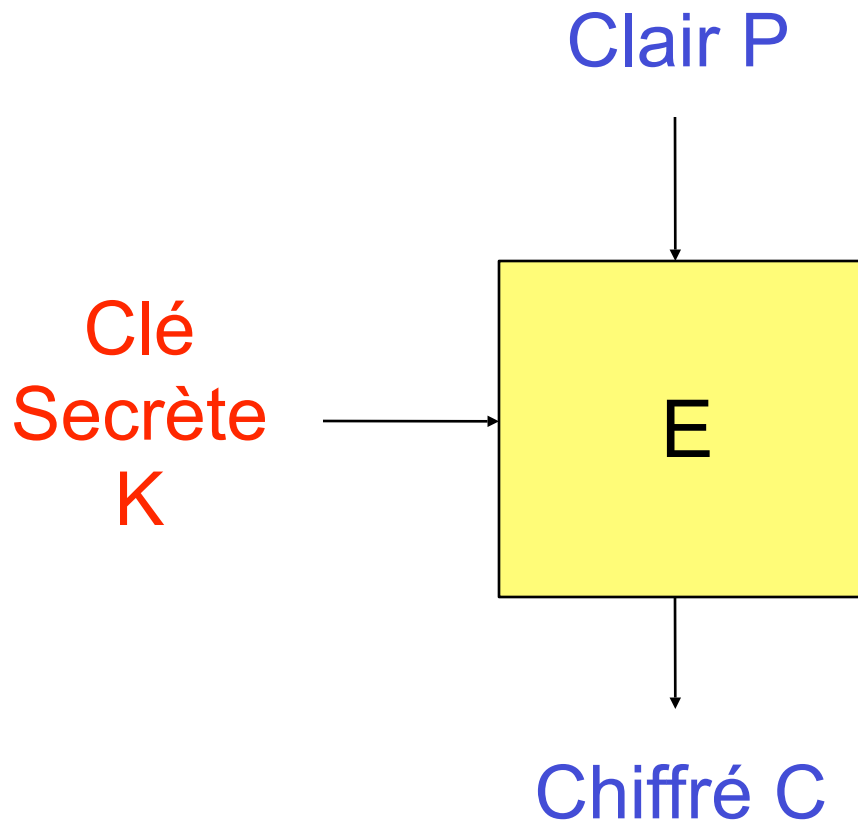
Equipe de cryptographie

Ecole normale supérieure

# Plan du cours

- Principes généraux
  - Rappel : block ciphers
  - Attaques génériques
- Cryptanalyse contre le DES
  - Cryptanalyse différentielle
  - Cryptanalyse linéaire
- Critères de résistance pour l'AES
- Autres techniques de cryptanalyse

# Block Cipher



Notation :

$$E : \{0,1\}^n \times \{0,1\}^k$$

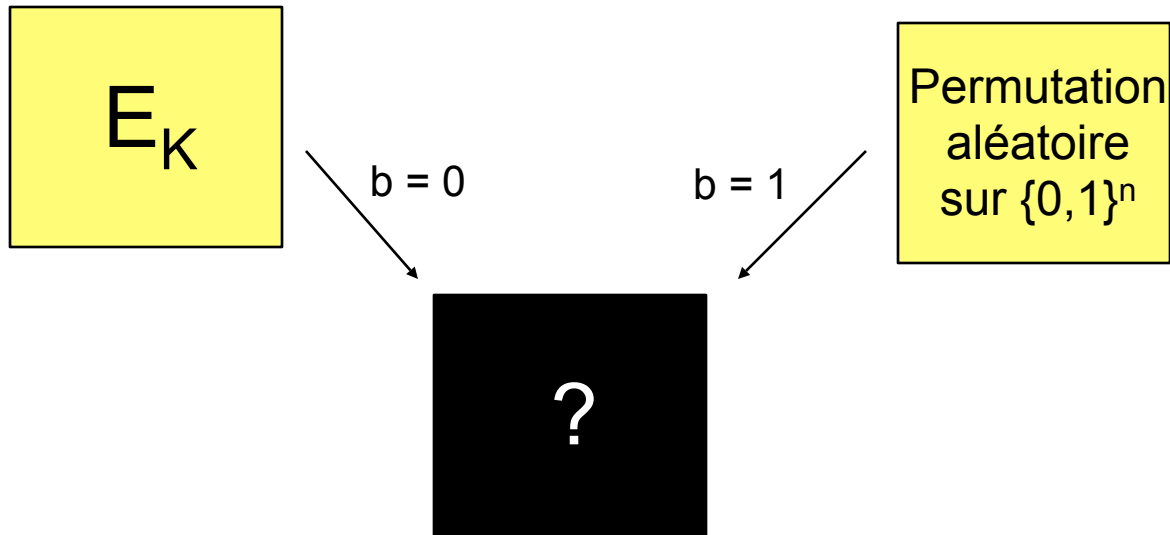
→

$$\{0,1\}^n$$

# Sécurité

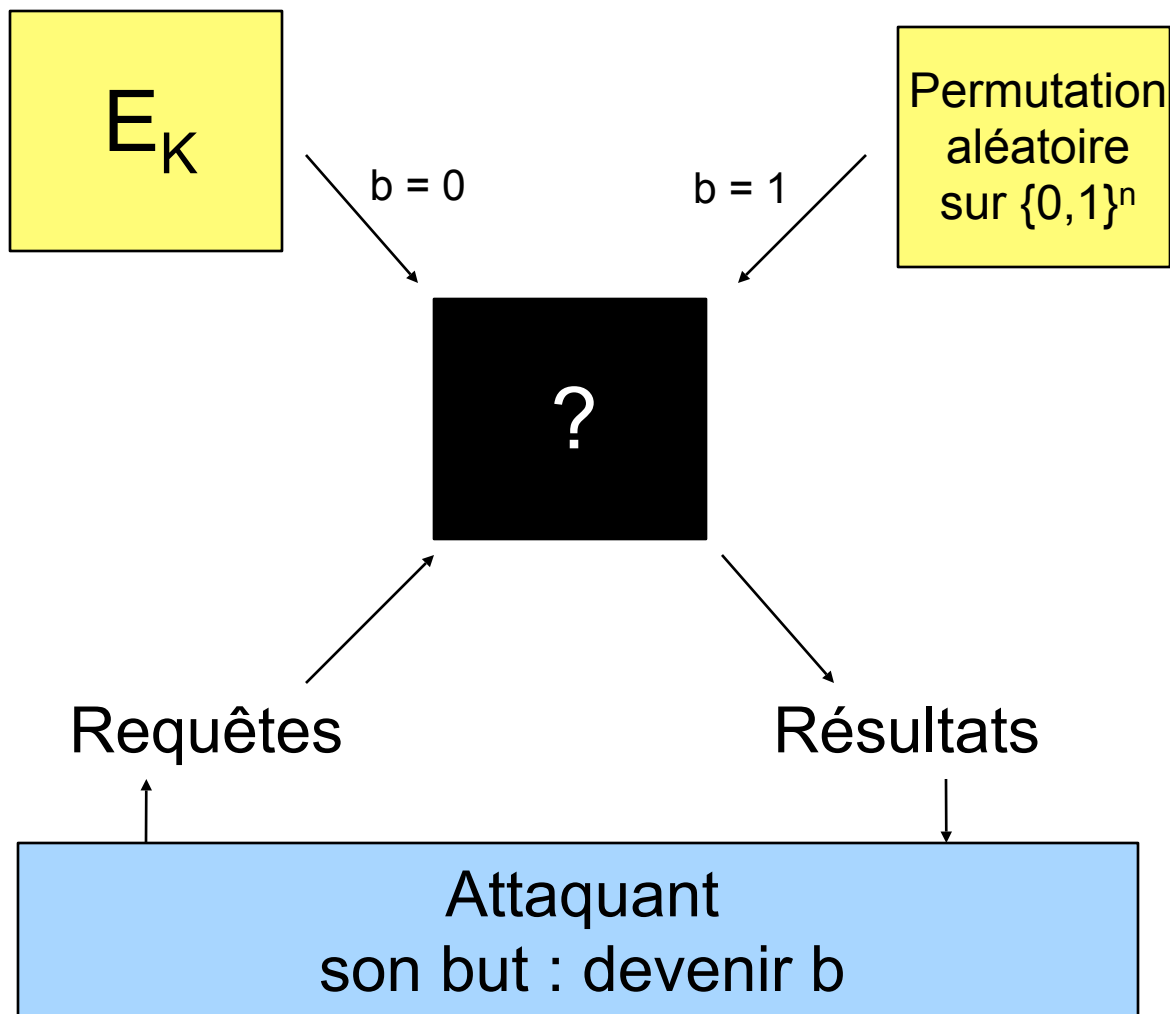
- Idéalement,  $C$  ne doit laisser fuir aucune information sur  $P$  ou sur  $K$
- La sécurité n'est jamais «parfaite». Un attaquant connaissant  $C$  et  $P$  peut « tester toutes les clés »  $\rightarrow 2^k$
- Modèle de la boîte noire

# Modèle de la boîte noire



Attaquant  
son but : devenir  $b$

# Modèle de la boîte noire



# Sécurité

## Techniques de cryptanalyse :

- Attaques génériques (ne dépendent pas de l'algorithme)
- Techniques d'attaque « classiques » (DES)
  - Différentielle
  - Linéaire
- Attaques ciblées contre 1 algorithme particulier

# Recherche exhaustive

$2^{31}$	Cycles / seconde (2GHz)
$2^{56}$	Recherche exhaustive DES (RC5 - 1997 – distributed.net)
$2^{64}$	« Record » de recherche exhaustive (RC5 – 2002- distributed.net)
$2^{72}$	Tentative en cours (RC5 – distributed.net)
$2^{128}$	Sécurité de AES



# Utiliser un précalcul ?

## Compromis temps/mémoire de Hellman

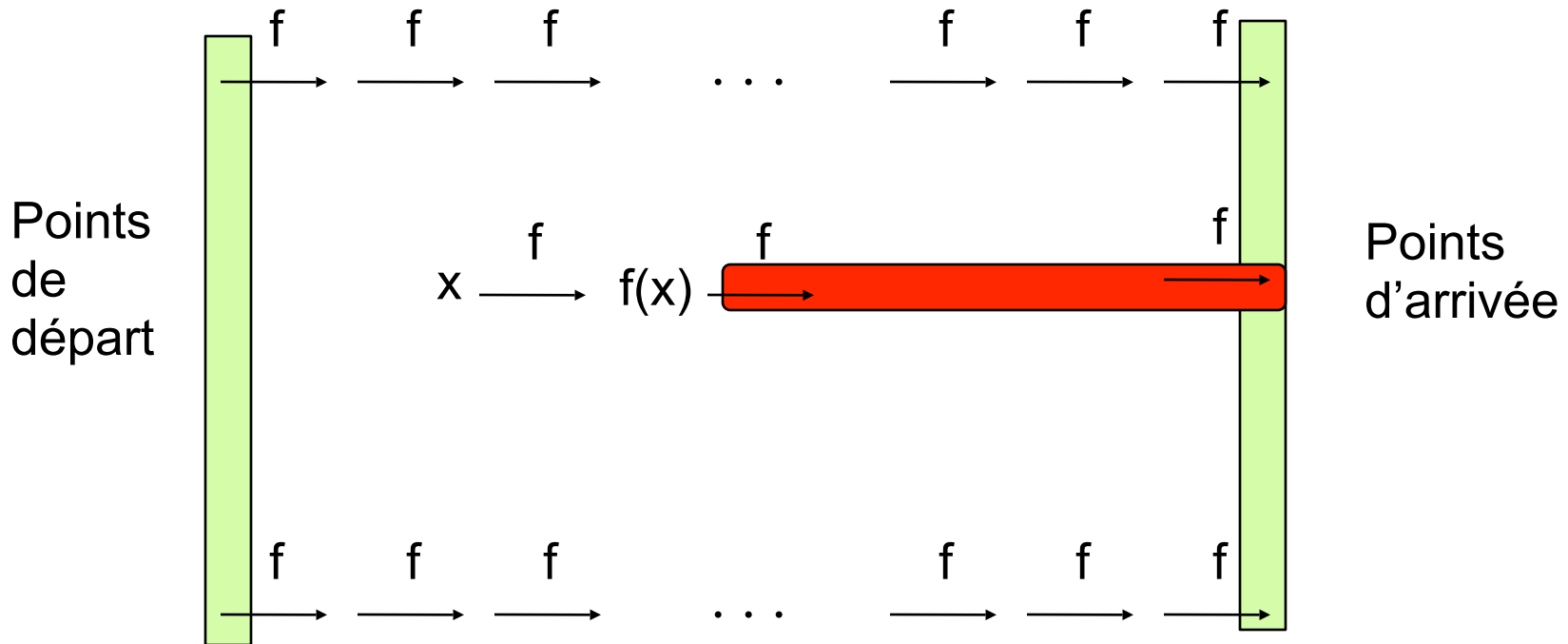
- Clé de  $k$  bits
- Précalcul  $2^k$
- Temps  $2^{2k/3}$
- Mémoire  $2^{2k/3}$

- DES : 56 bits
- Précalcul  $2^{56}$
- Temps  $2^{39}$
- Mémoire  $2^{39}$

« Amortir » le coup d'une recherche exhaustive

« Rainbow table » : application aux mots de passe Windows

# Idée générale



Pour inverser  $f(x)$  : **mémoire nécessaire** = #lignes  
**temps nécessaire** = #colonnes  
(#lignes) x (#colonnes) = espace des clés

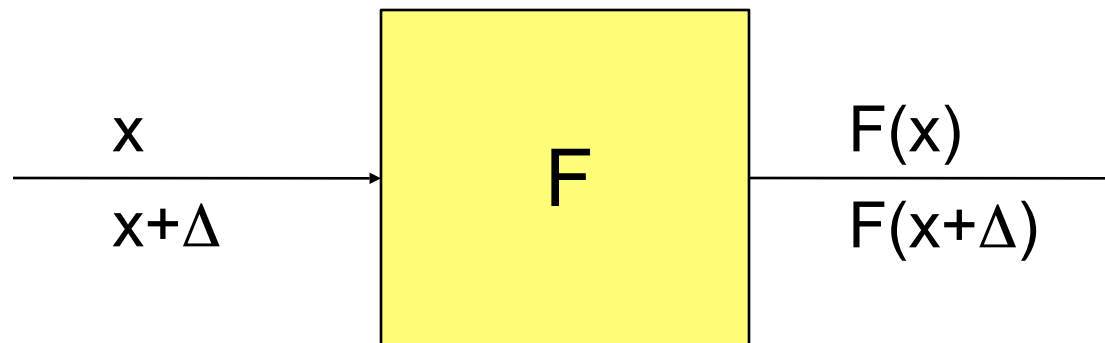
# Attaques génériques

- Développées contre le DES
  - Différentielle (Biham-Shamir 1990-1992)
  - Linéaire (Matsui 1991)
  - **Attaques statistiques** (approximations du comportement du DES)
- Appliquées à de nombreux algorithmes
- Intégrées dans les designs actuels

# Cryptanalyse différentielle

- Soit  $F : \{0,1\}^n \rightarrow \{0,1\}^n$
- Idée générale, on étudie la « différentielle »  $F^*$  d'une fonction  $F$  au point  $x$

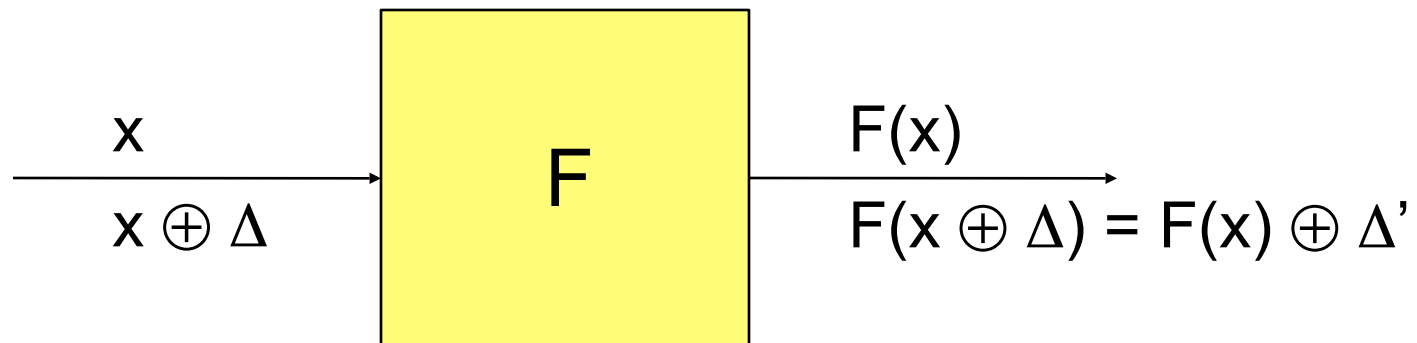
$$F^* : \Delta \rightarrow F(x+\Delta) - F(x)$$



# Cas des block ciphers

On travaille bit à bit : Addition == Soustraction == XOR

$$(A \oplus B) \oplus B = A$$



Observation : les block ciphers contiennent beaucoup de fonctions linéaires (uniquement des XOR)....

# Fonctions linéaires

Soit  $L$  une fonction **linéaire**  $\{0,1\}^n \rightarrow \{0,1\}^n$

# Fonctions linéaires

Soit  $L$  une fonction **linéaire**  $\{0,1\}^n \rightarrow \{0,1\}^n$

$$L(x \oplus y) = L(x) \oplus L(y)$$

# Fonctions linéaires

Soit  $L$  une fonction **linéaire**  $\{0,1\}^n \rightarrow \{0,1\}^n$

$$L(x \oplus y) = L(x) \oplus L(y)$$

Alors la différentielle de  $L$  est très simple, en tout point  $x$  :

$$\begin{aligned} L^*(\Delta) &= L(x \oplus \Delta) \oplus L(x) \\ &= L(\Delta) \end{aligned}$$



# Fonctions linéaires

Soit  $L$  une fonction **linéaire**  $\{0,1\}^n \rightarrow \{0,1\}^n$

$$L(x \oplus y) = L(x) \oplus L(y)$$

Alors la différentielle de  $L$  est très simple, en tout point  $x$  :

$$\begin{aligned} L^*(\Delta) &= L(x \oplus \Delta) \oplus L(x) \\ &= L(\Delta) \end{aligned}$$

**Donc  $L^* = L$  en tout point**

# Fonction « affine »

- Ajout d'une sous-clé  $K$

$$L_K(x) = x \oplus K$$

- $L_K^*(\Delta) = L_K(x \oplus \Delta) \oplus L_K(x)$   
 $= (x \oplus \Delta \oplus K) \oplus (x \oplus K)$   
 $= \Delta$

- La différentielle  $L_K^*$  en tout point est indépendant de  $K$  !

# Fonction non-linéaire

La différentielle  $F^*$  dépend du point  $x$  concerné

Exemple  $F : \{0,1\}^2 \rightarrow \{0,1\}^2$   
 $F(x,y) = (x.y, x)$

# Fonction non-linéaire

La différentielle  $F^*$  dépend du point  $x$  concerné

Exemple  $F : \{0,1\}^2 \rightarrow \{0,1\}^2$

$$F(x,y) = (x.y, x) \longrightarrow$$

$$F(0,0) = (0,0)$$

$$F(0,1) = (0,0)$$

$$F(1,0) = (0,1)$$

$$F(1,1) = (1,1)$$

# Fonction non-linéaire

La différentielle  $F^*$  dépend du point  $x$  concerné

Exemple  $F : \{0,1\}^2 \rightarrow \{0,1\}^2$   
 $F(x,y) = (x.y, x)$

# Fonction non-linéaire

La différentielle  $F^*$  dépend du point  $x$  concerné

Exemple  $F : \{0,1\}^2 \rightarrow \{0,1\}^2$   
 $F(x,y) = (x.y, x)$

Différentielle en  $(0,0)$   $\longrightarrow$

$$\begin{aligned} F^*(0,0) &= (0,0) \\ F^*(0,1) &= (0,0) \\ F^*(1,0) &= (0,1) \\ F^*(1,1) &= (1,1) \end{aligned}$$

# Fonction non-linéaire

La différentielle  $F^*$  dépend du point  $x$  concerné

Exemple  $F : \{0,1\}^2 \rightarrow \{0,1\}^2$   
 $F(x,y) = (x.y, x)$

Différentielle en  $(0,0)$   $\longrightarrow$

$$\begin{aligned} F^*(0,0) &= (0,0) \\ F^*(0,1) &= (0,0) \\ F^*(1,0) &= (0,1) \\ F^*(1,1) &= (1,1) \end{aligned}$$

Différentielle en  $(1,1)$   $\longrightarrow$

$$\begin{aligned} F^*(0,0) &= (0,0) \\ F^*(0,1) &= (1,0) \\ F^*(1,0) &= (1,1) \\ F^*(1,1) &= (1,1) \end{aligned}$$

# Pour résumer

- Fonctions linéaires
  - Différentielle **prévisible de façon exacte**
- Fonctions affines (XOR de sous-clé)
  - Différentielle **indépendante de la clé**
- Fonctions non-linéaires
  - **On ne peut rien dire** de façon générale
  - Donc on ne peut pas calculer directement la différentielle pour tout le block cipher
- On adopte une approche **statistique**



# Probabilité

- **Caractéristique** différentielle
  - Différence  $\Delta$  en entrée de F
  - Différence  $\Delta'$  en sortie de F
  - Probabilité  $p$  associée (moyennée sur tous les  $x$  possibles)
- Notation :  $\Delta \rightarrow \Delta'$  [proba =  $p$ ]

# Table des différences

$$F : \{0,1\}^2 \rightarrow \{0,1\}^2 : F(x,y) = (x.y, x)$$

Somme = 1  
Sur la ligne

$\Delta \backslash \Delta'$	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	0	0	0
(0,1)	0,5	0	0,5	0
(1,0)	0	0,5	0	0,5
(1,1)	0	0,5	0	0,5

La somme ne vaut pas 1

# Fonction linéaire

$$F : \{0,1\}^2 \rightarrow \{0,1\}^2 : F(x,y) = (x \oplus y, x)$$

Somme = 1

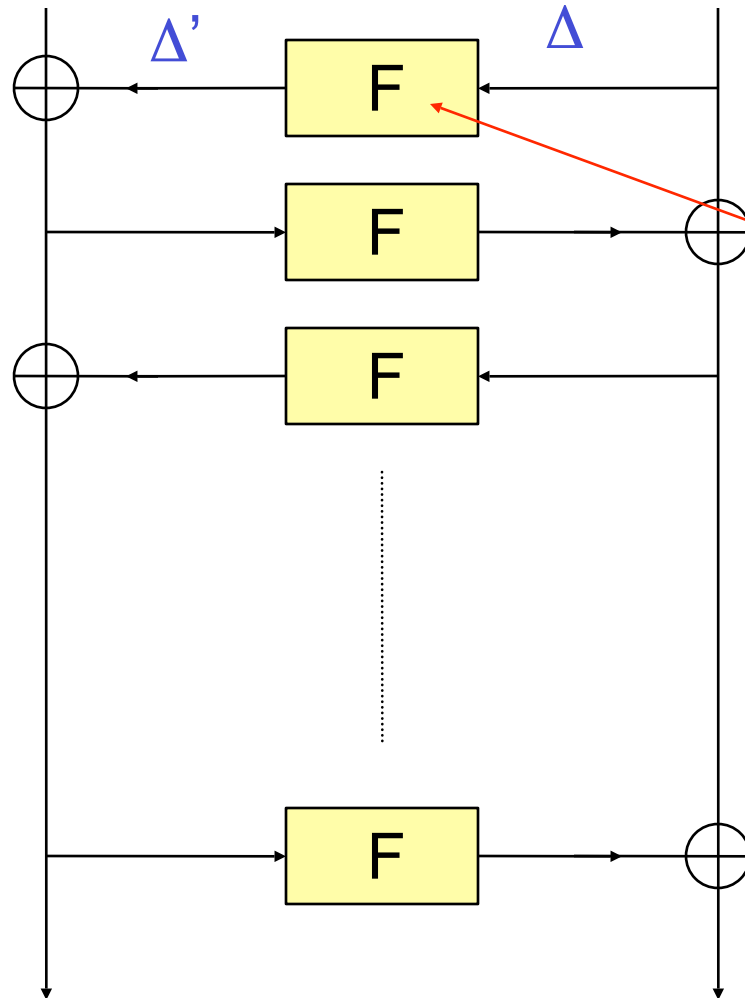
$\Delta \backslash \Delta'$	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	0	0	0
(0,1)	0	0	1	0
(1,0)	0	0	0	1
(1,1)	0	1	0	0

Somme = 1

# En pratique

- Calculer la table des différences coûte environ  $2^n * 2^n = 2^{2n}$  pour une fonction sur n bits
- Impossible pour le block cipher entier !
- Approche plus subtile :
  - Étude des fonctions élémentaires
  - Composition des caractéristiques différentielles
  - On cherche à trouver les meilleurs  $\Delta \rightarrow \Delta'$  [proba = p]

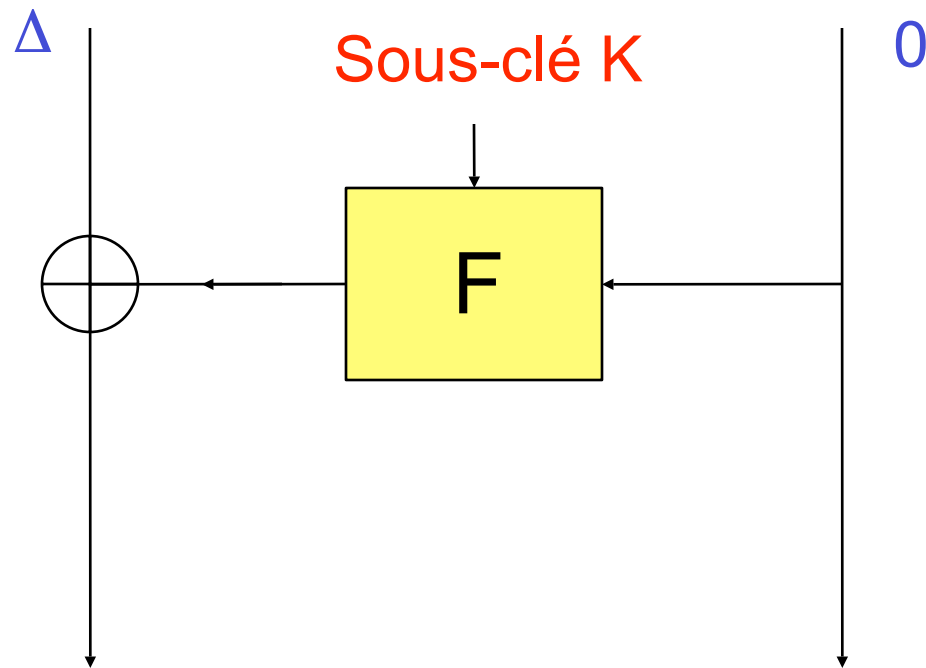
# Algorithme DES



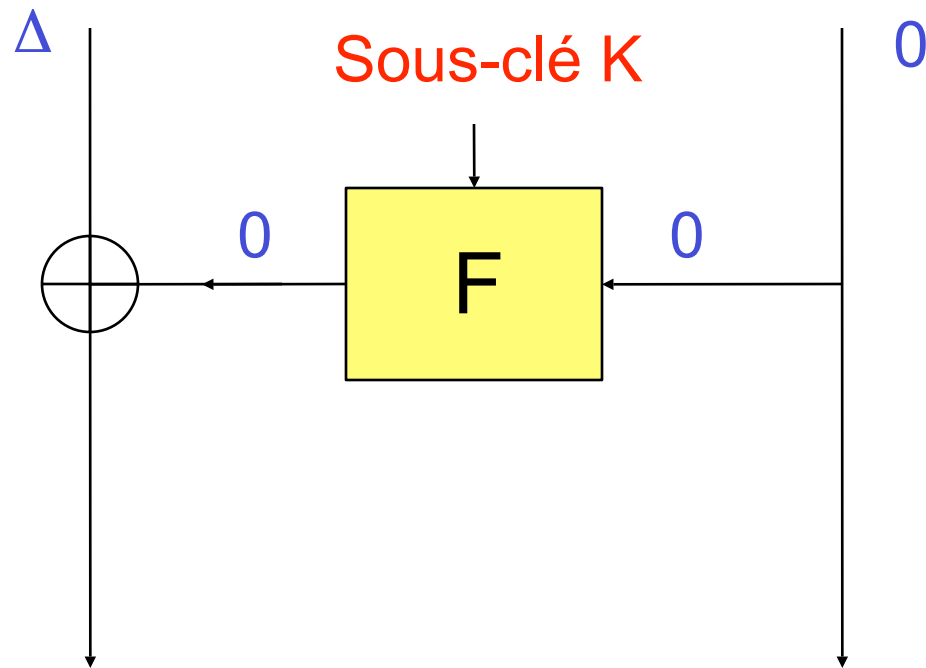
Propriétés  
différentielles  
de  $F$

$\Delta \rightarrow \Delta'$  [proba =  $p$ ]

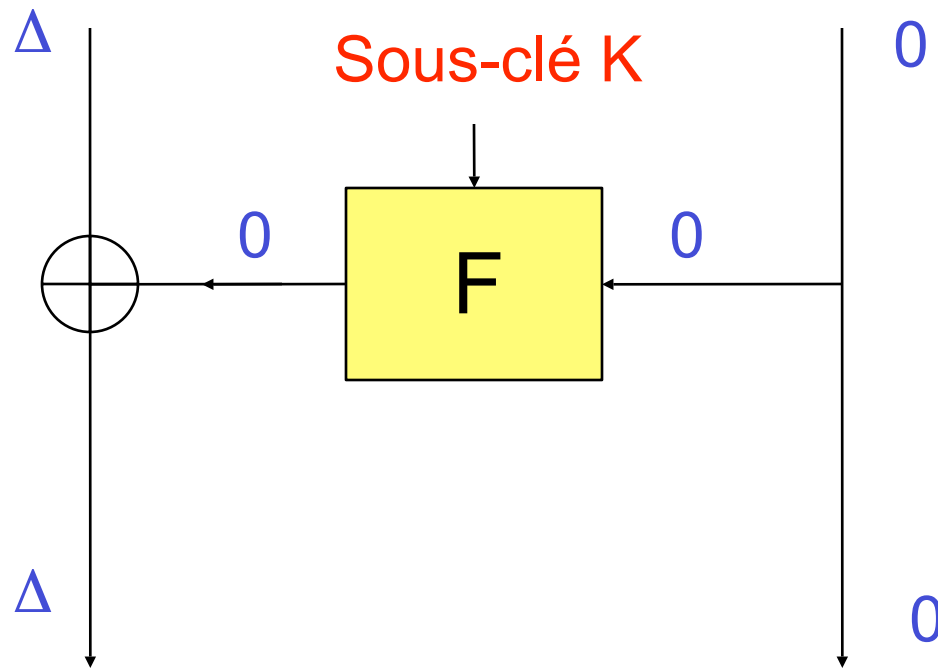
# 1 tour



# 1 tour

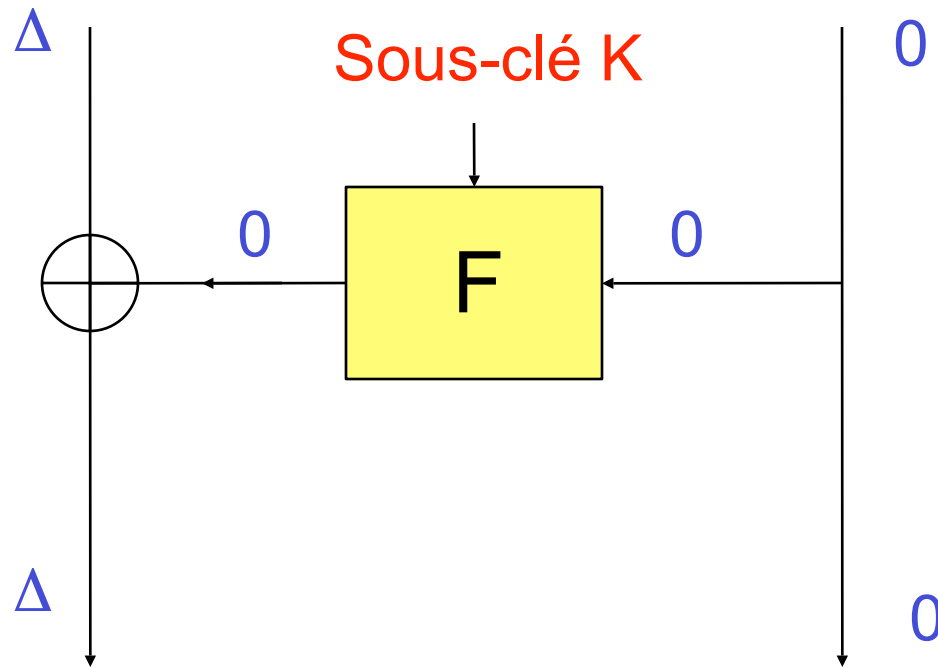


# 1 tour





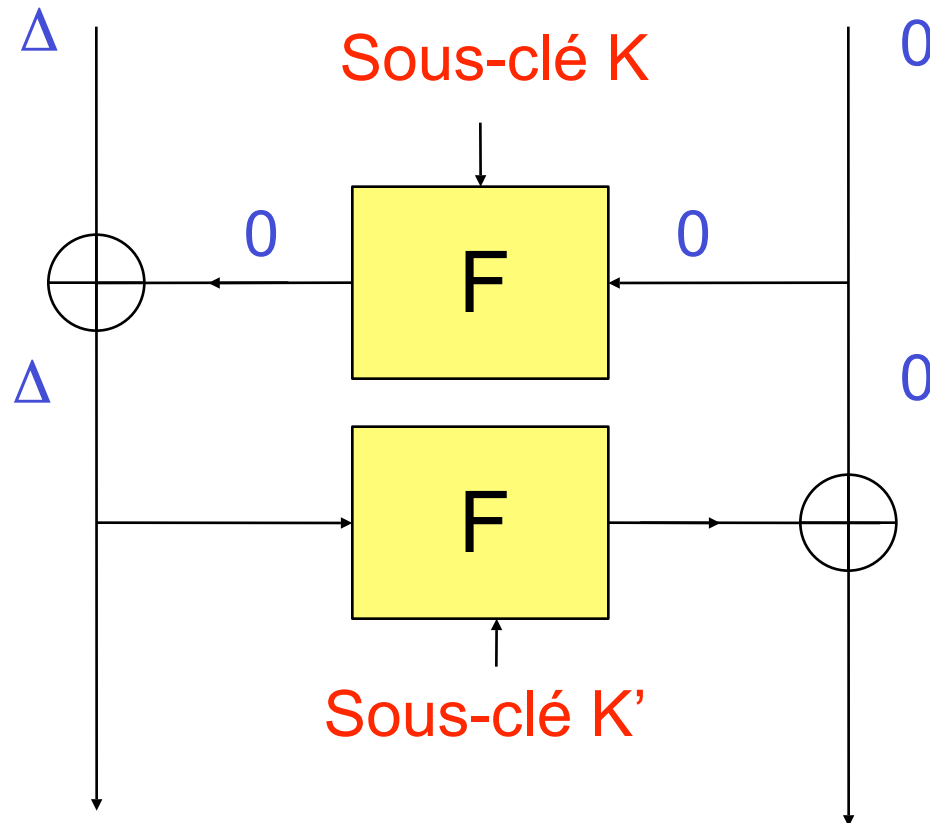
# 1 tour



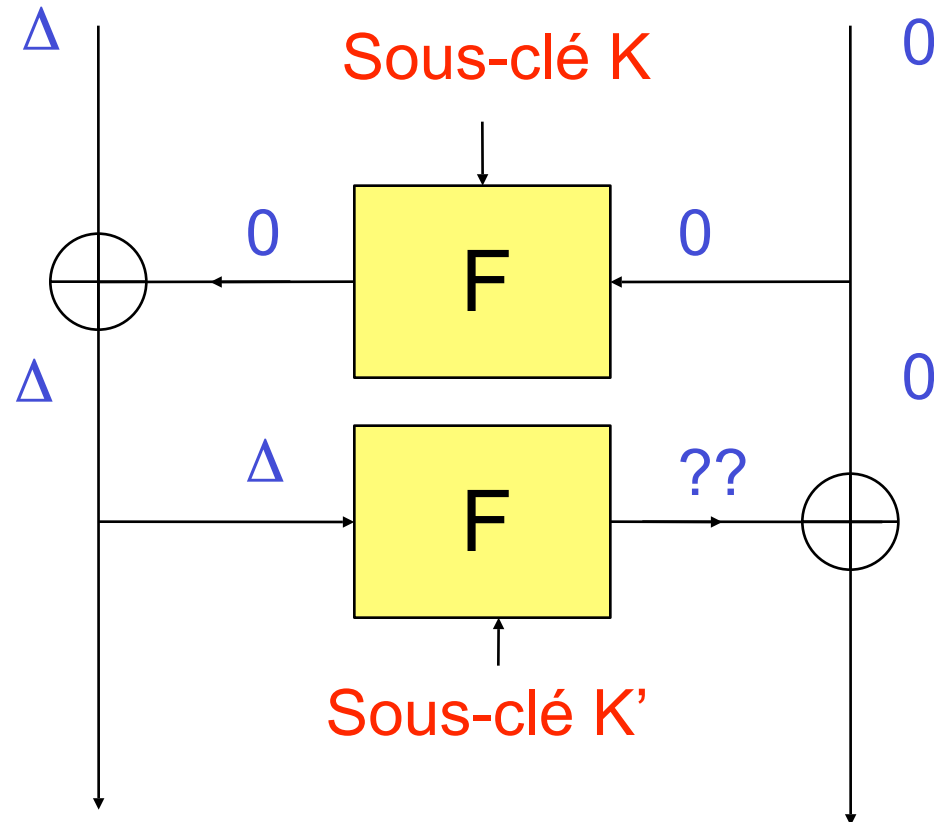
Donc  $\forall \Delta$  et  $\forall K$ , on a :

$$(\Delta, 0) \rightarrow (\Delta, 0) \quad [\text{proba} = 1]$$

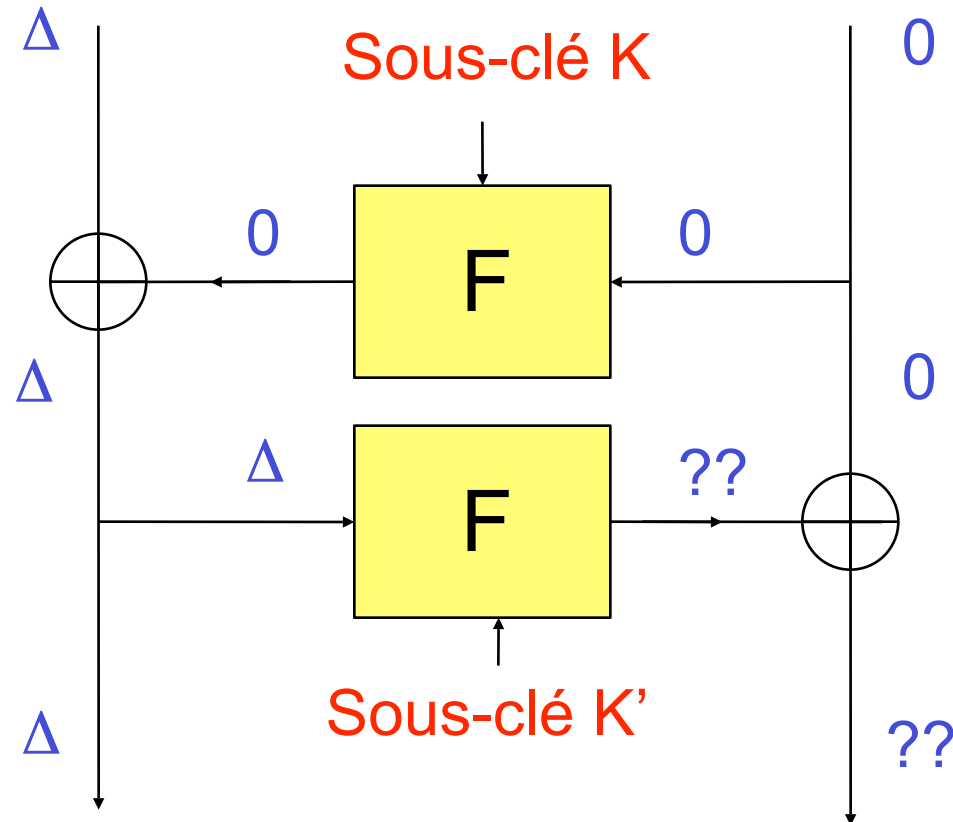
# 2 tours



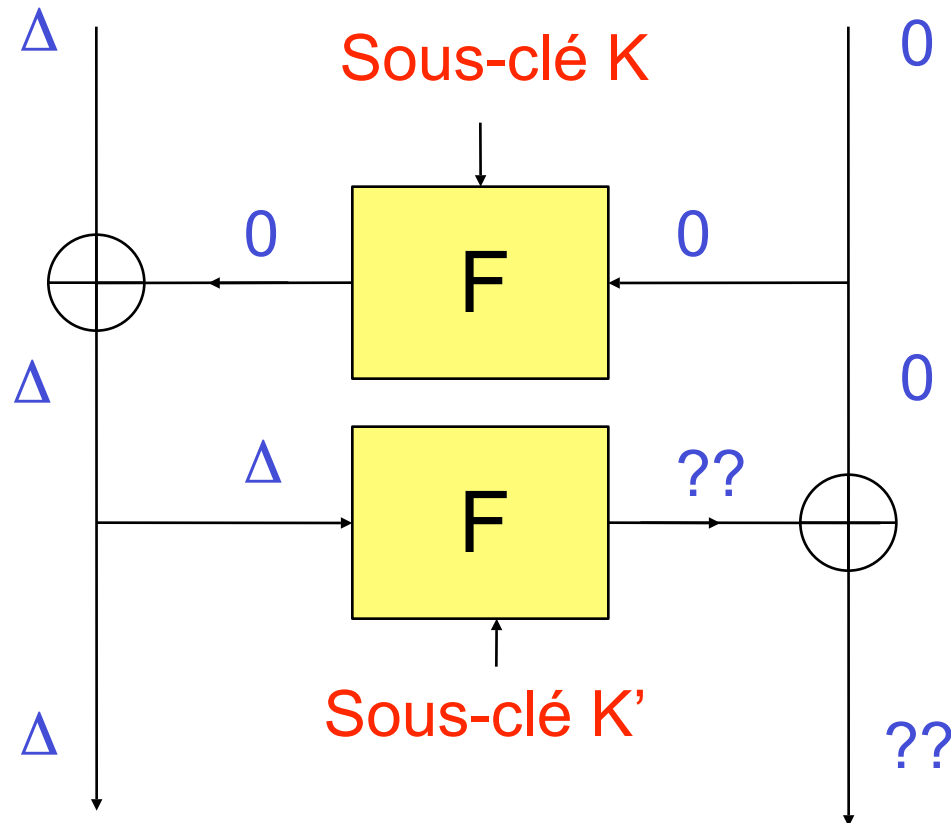
# 2 tours



# 2 tours

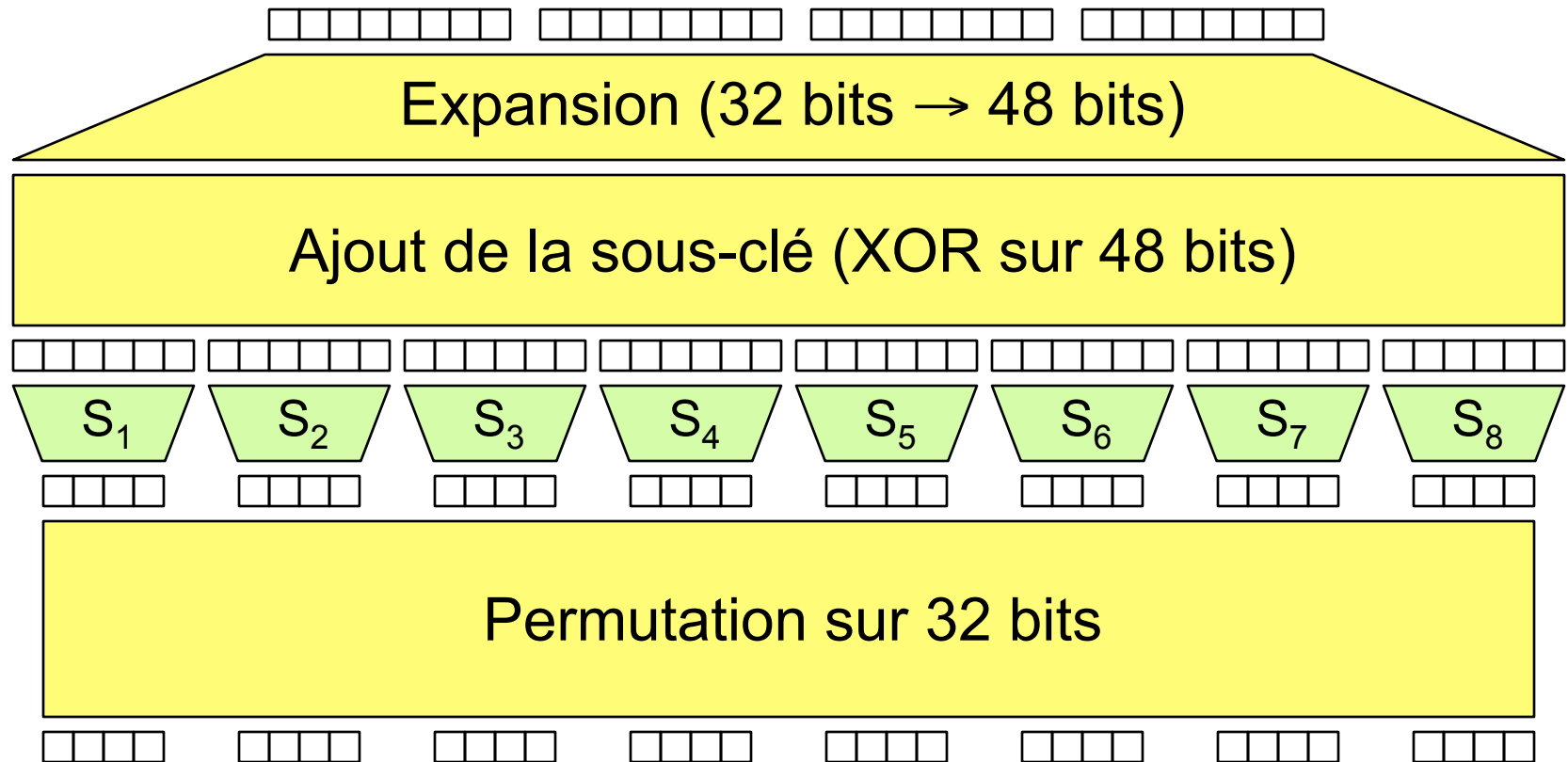


# 2 tours

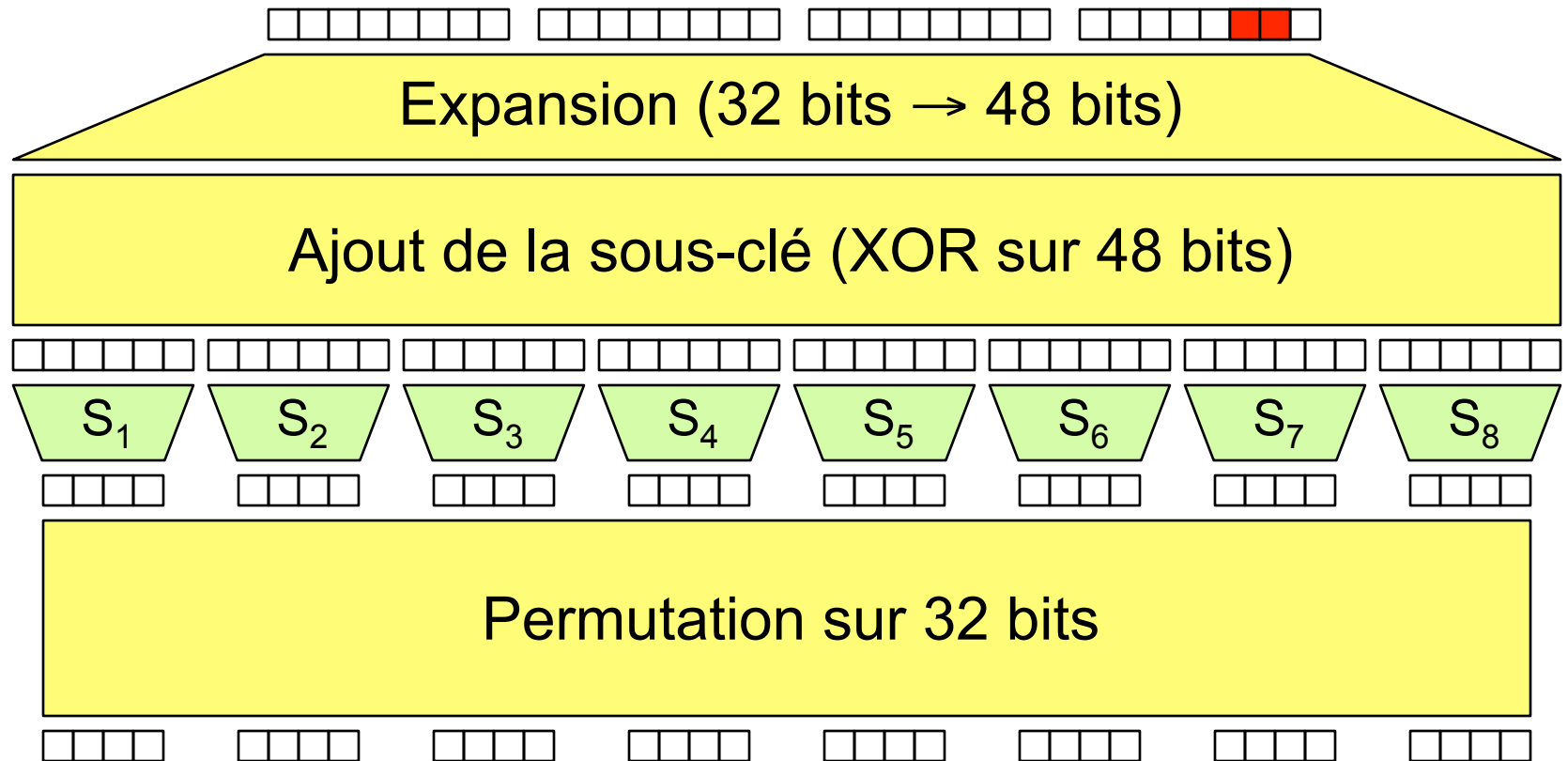


Nécessite d'étudier  $F$  plus en détails

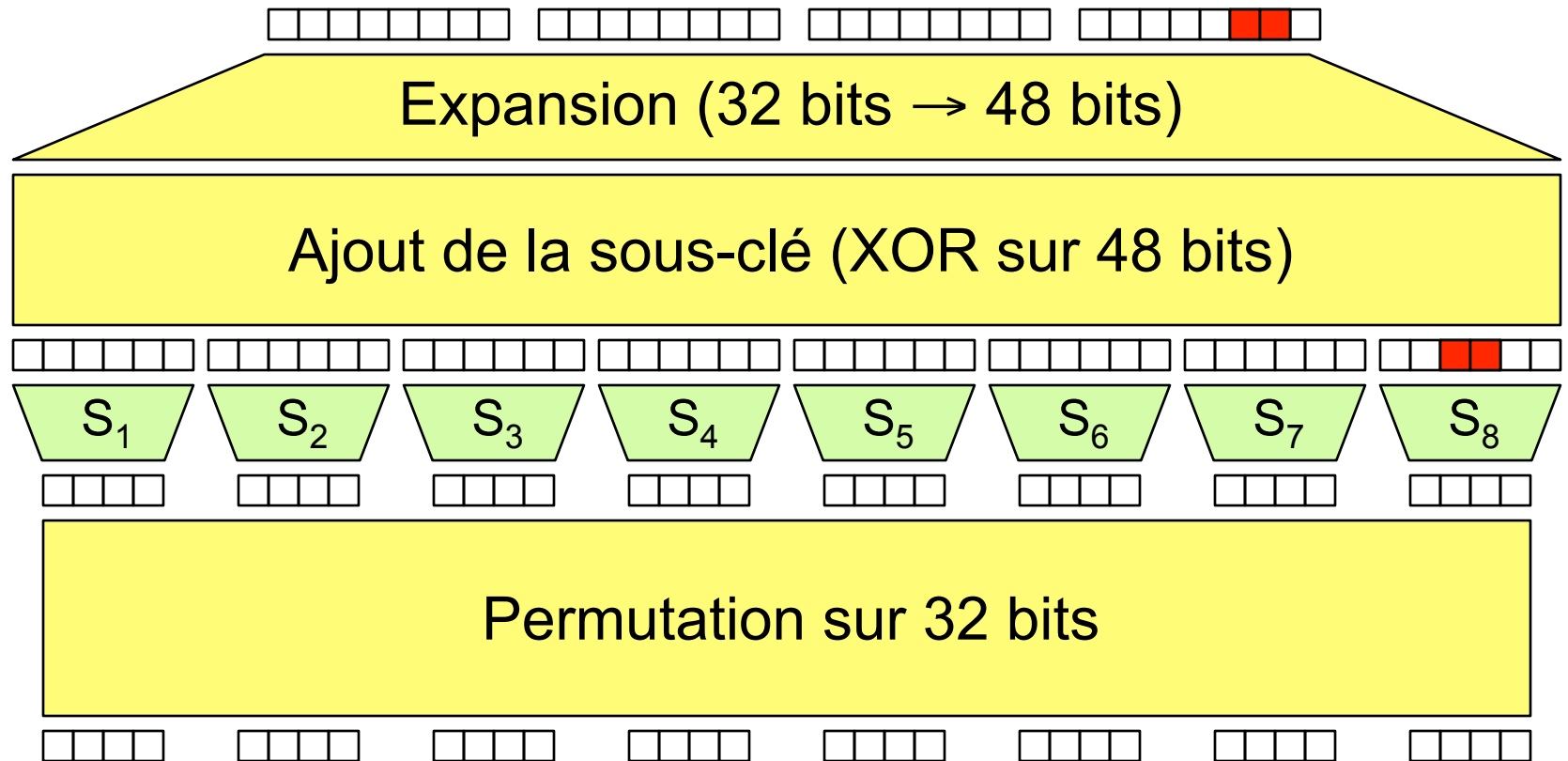
# Différentielle de F



# Différentielle de F

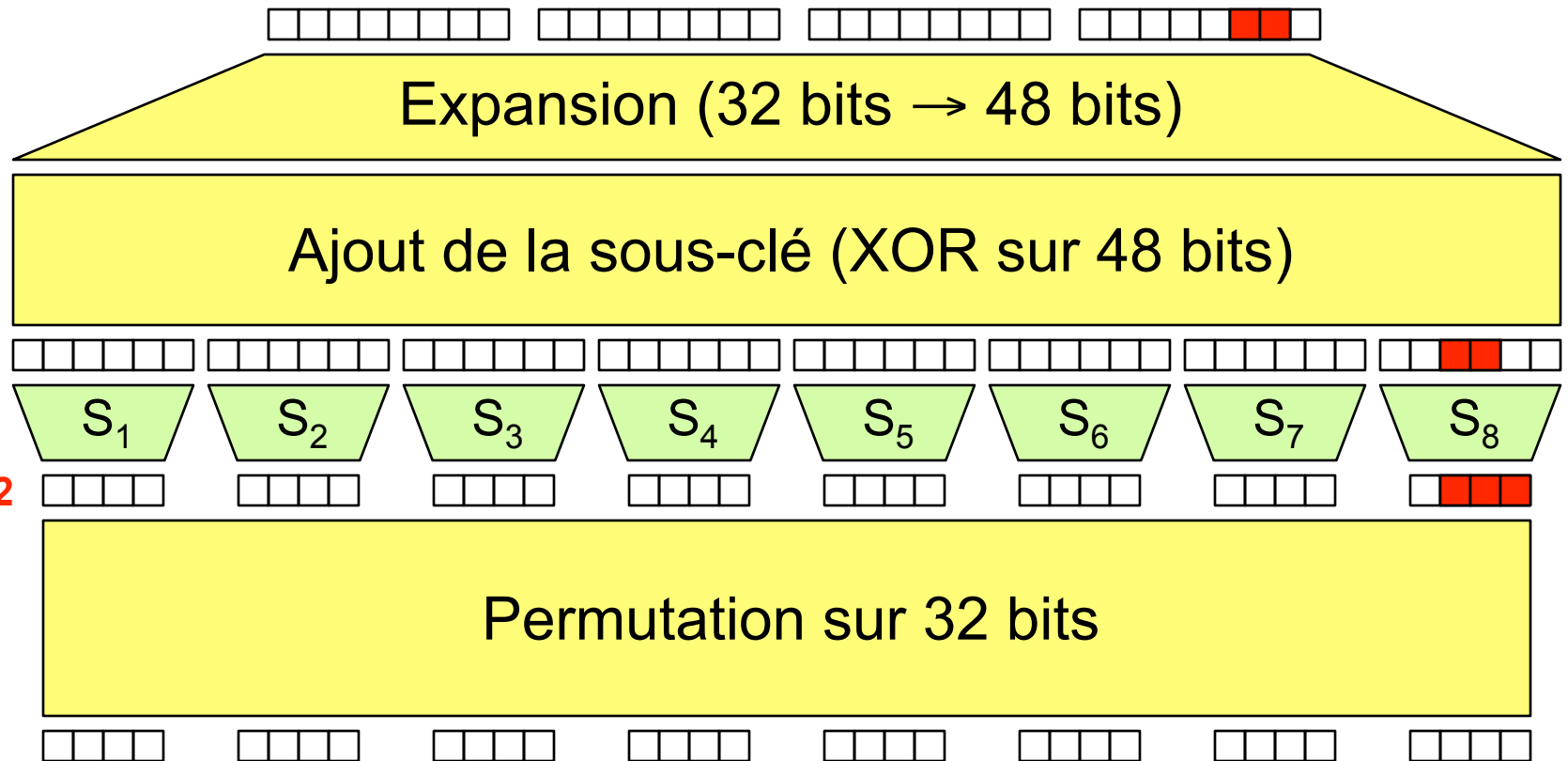


# Différentielle de F



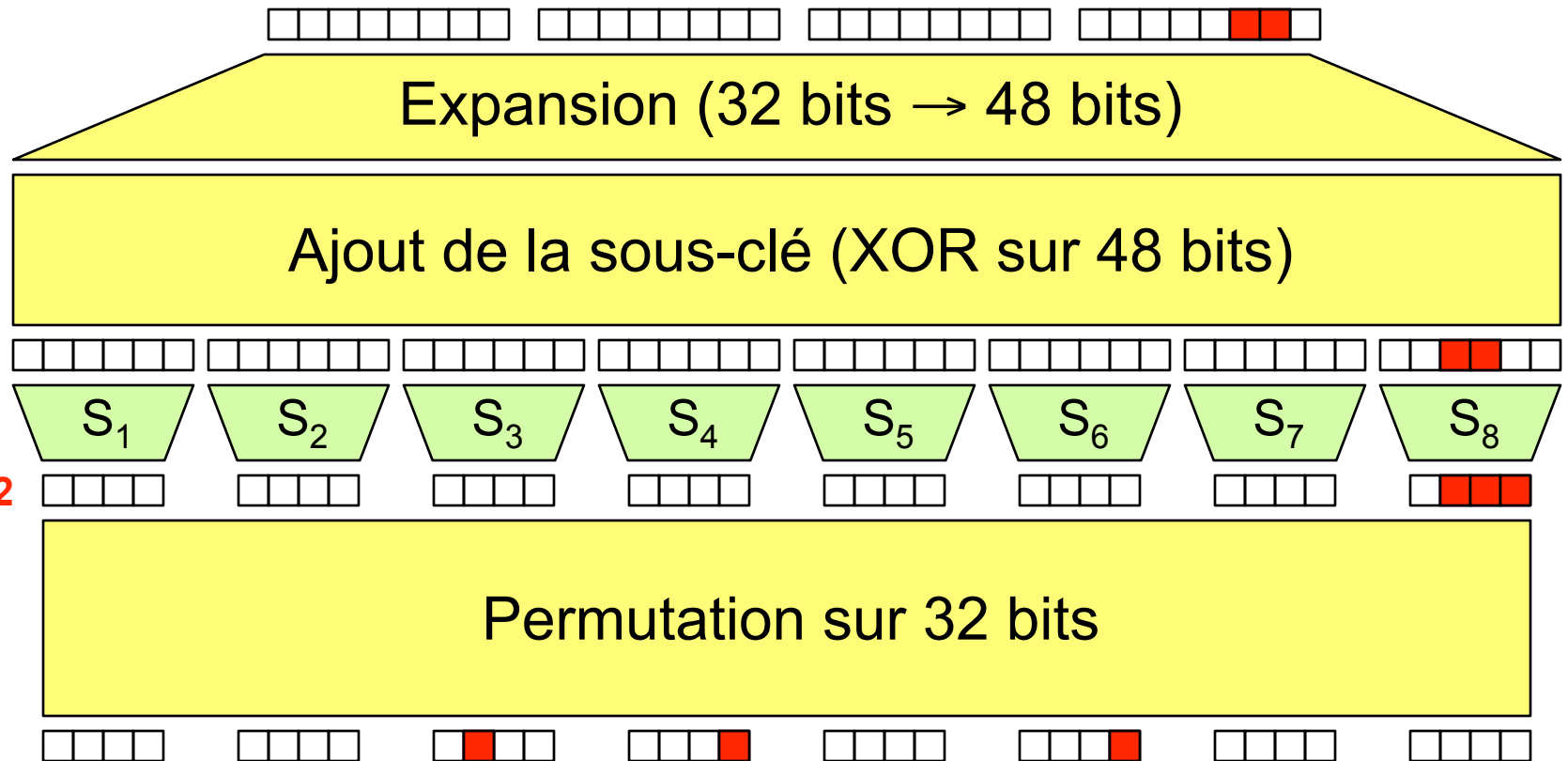


# Différentielle de F

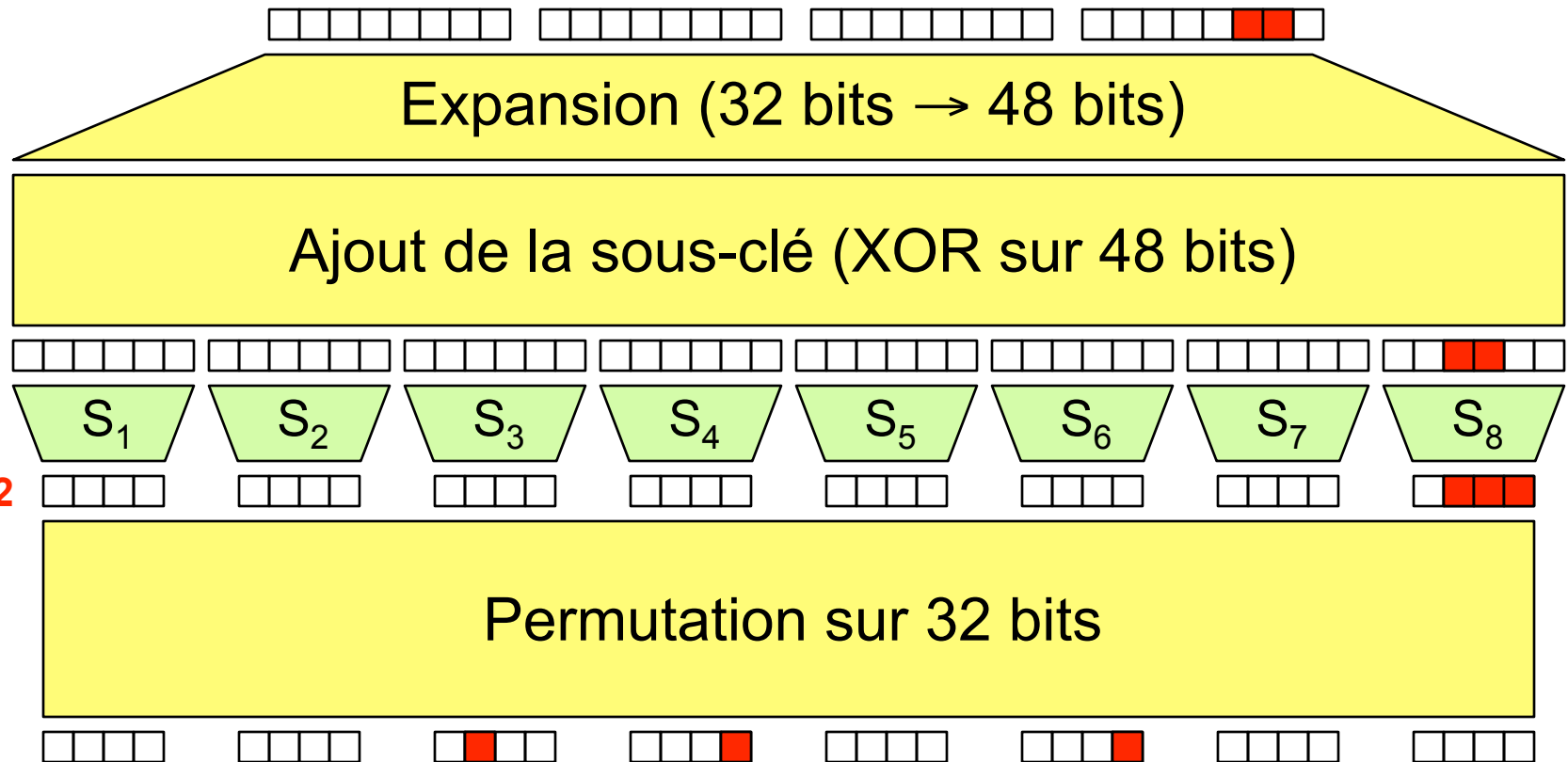


$p = 7/32$

# Différentielle de F

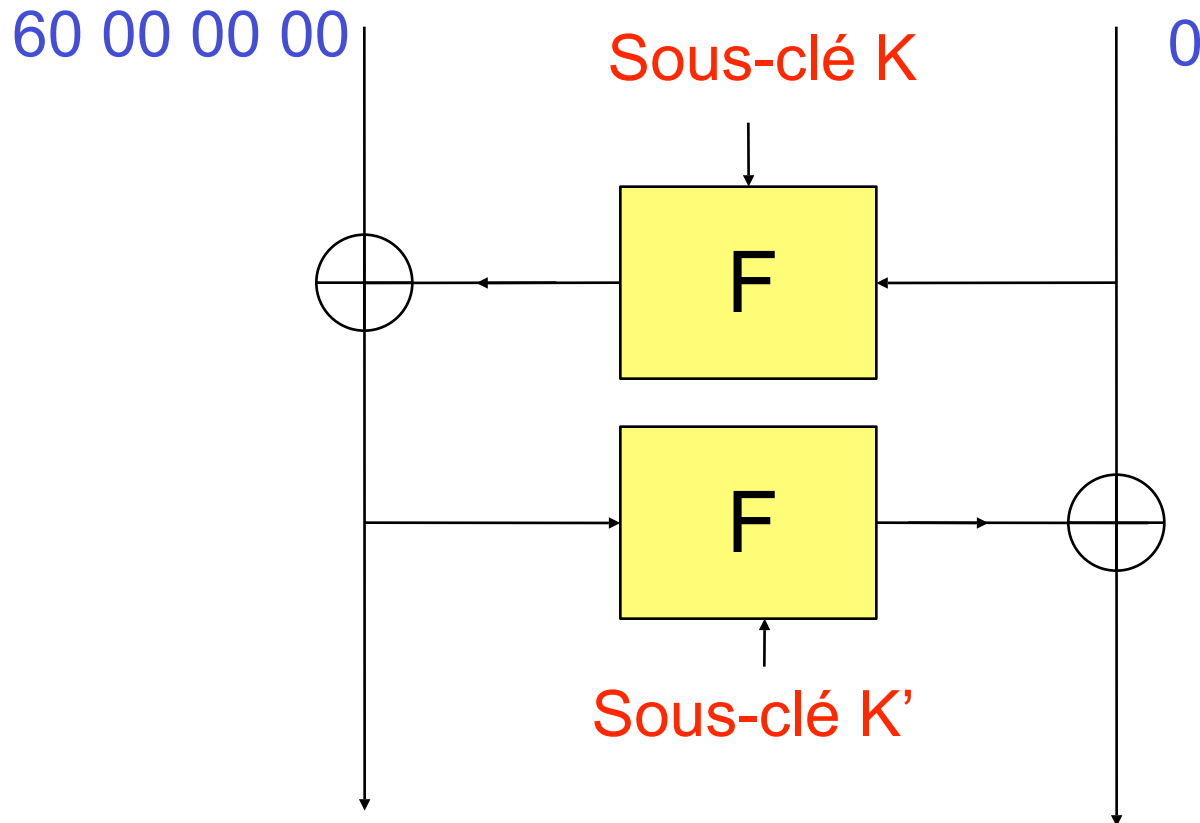


# Différentielle de F

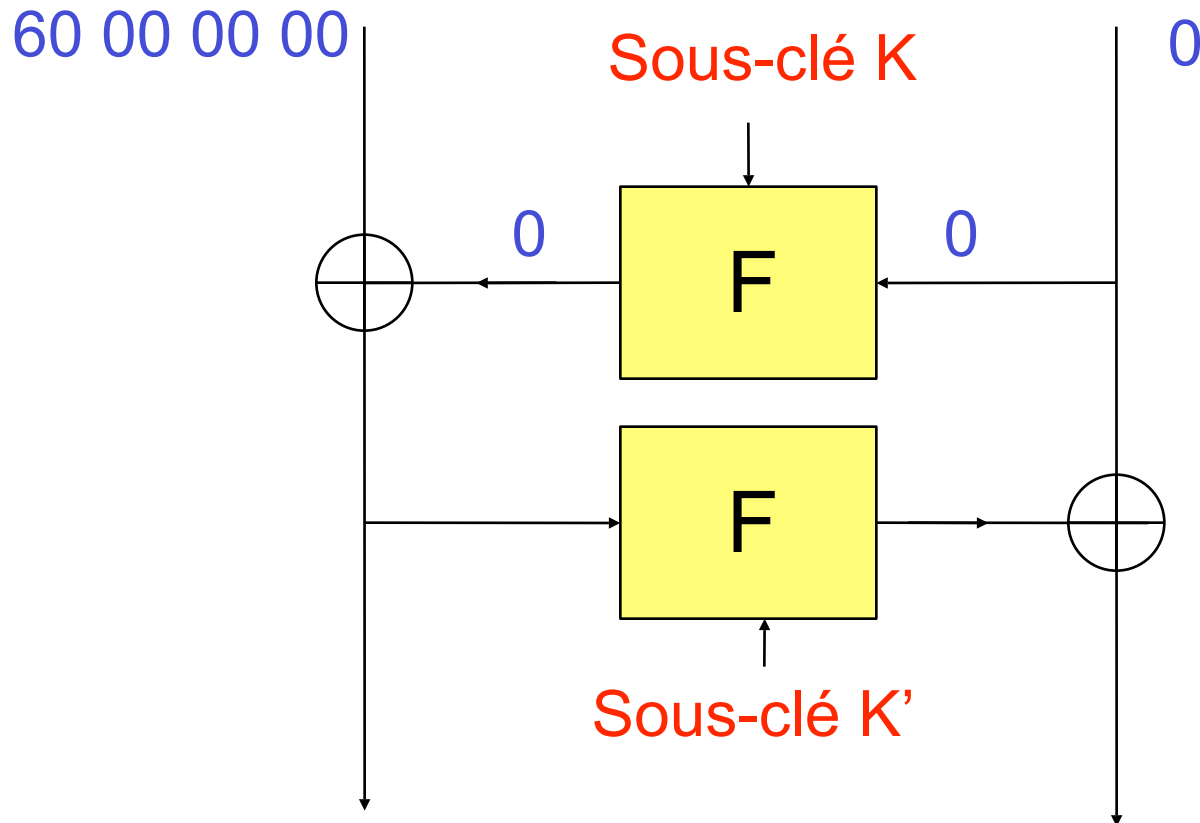


Donc : 60 00 00 00 → 00 80 82 00 avec probabilité  $7/32$

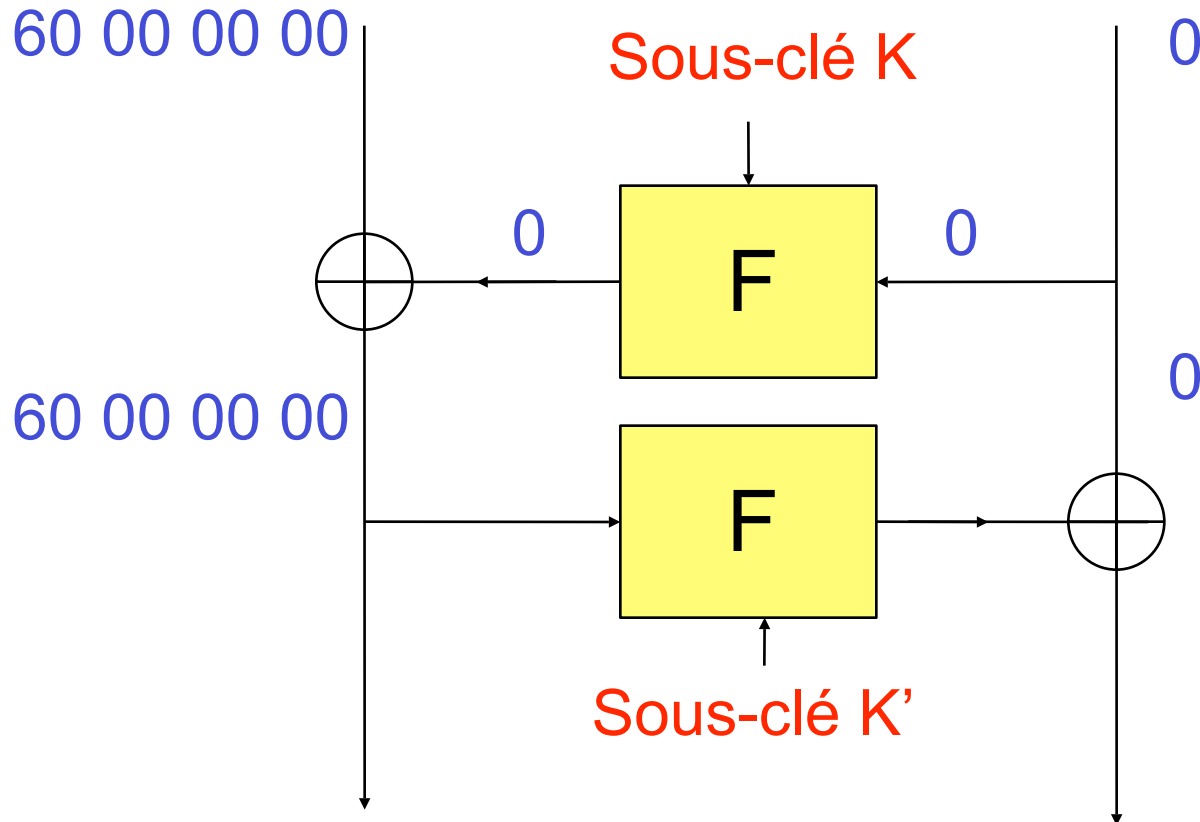
# 2 tours



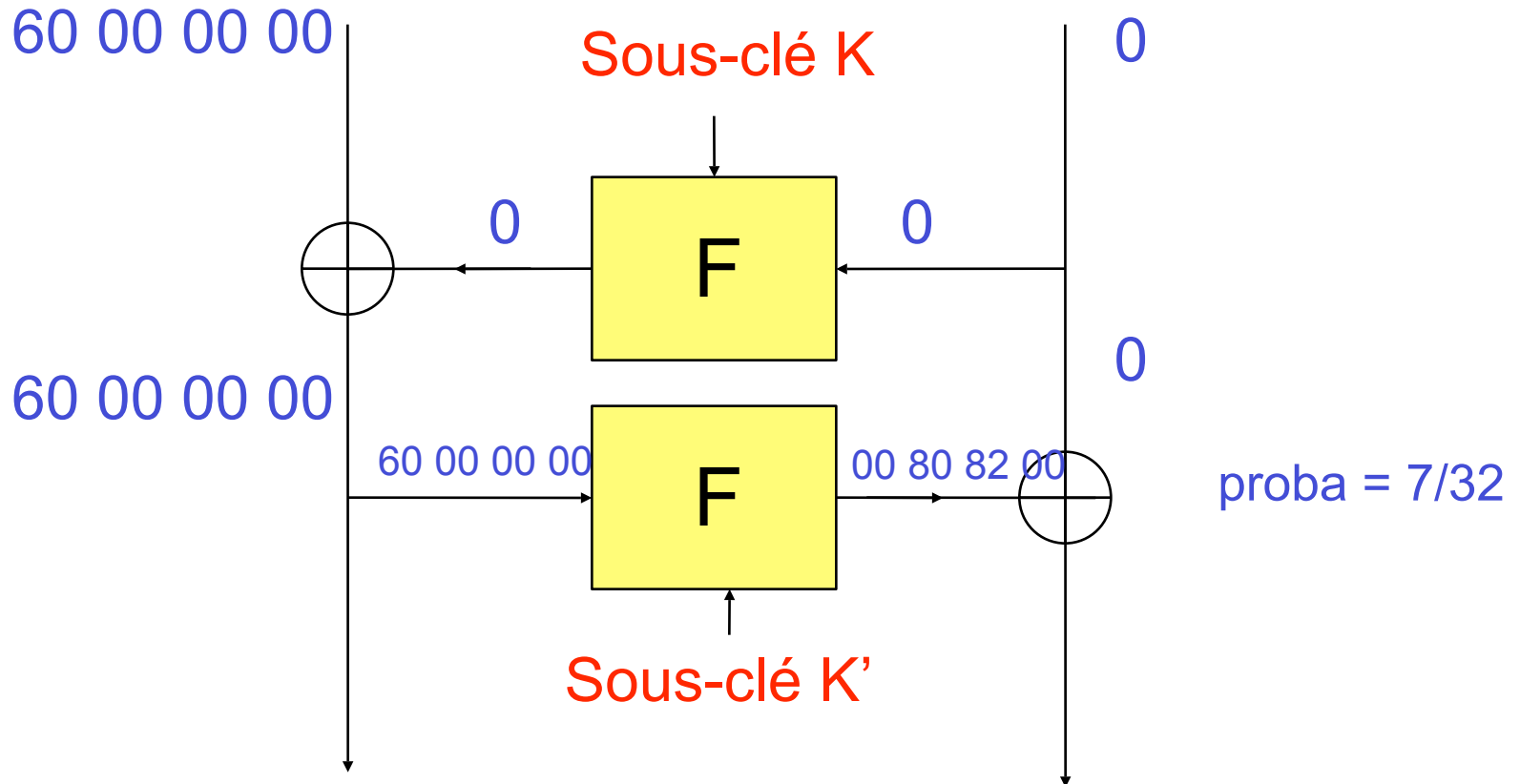
# 2 tours



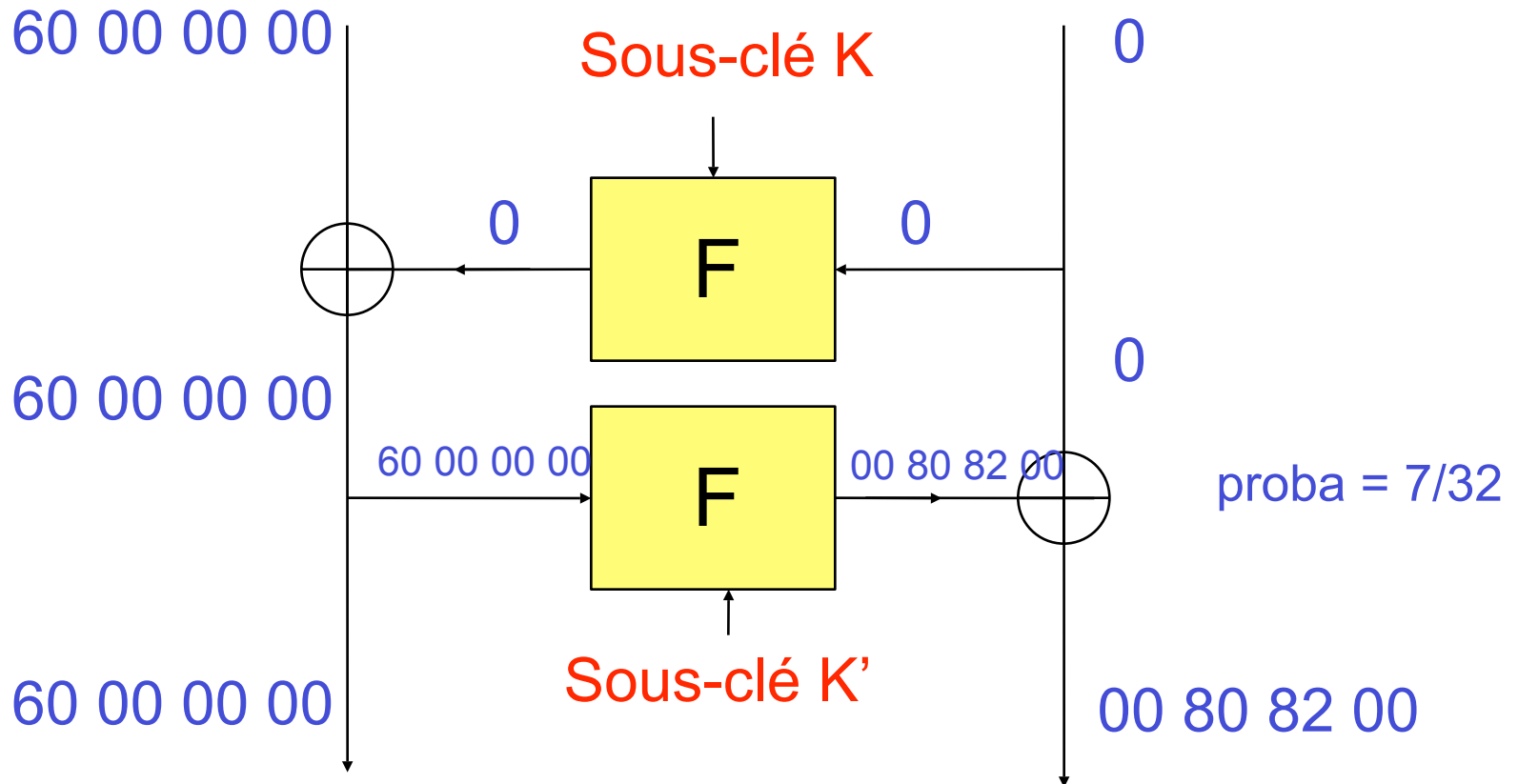
# 2 tours



# 2 tours

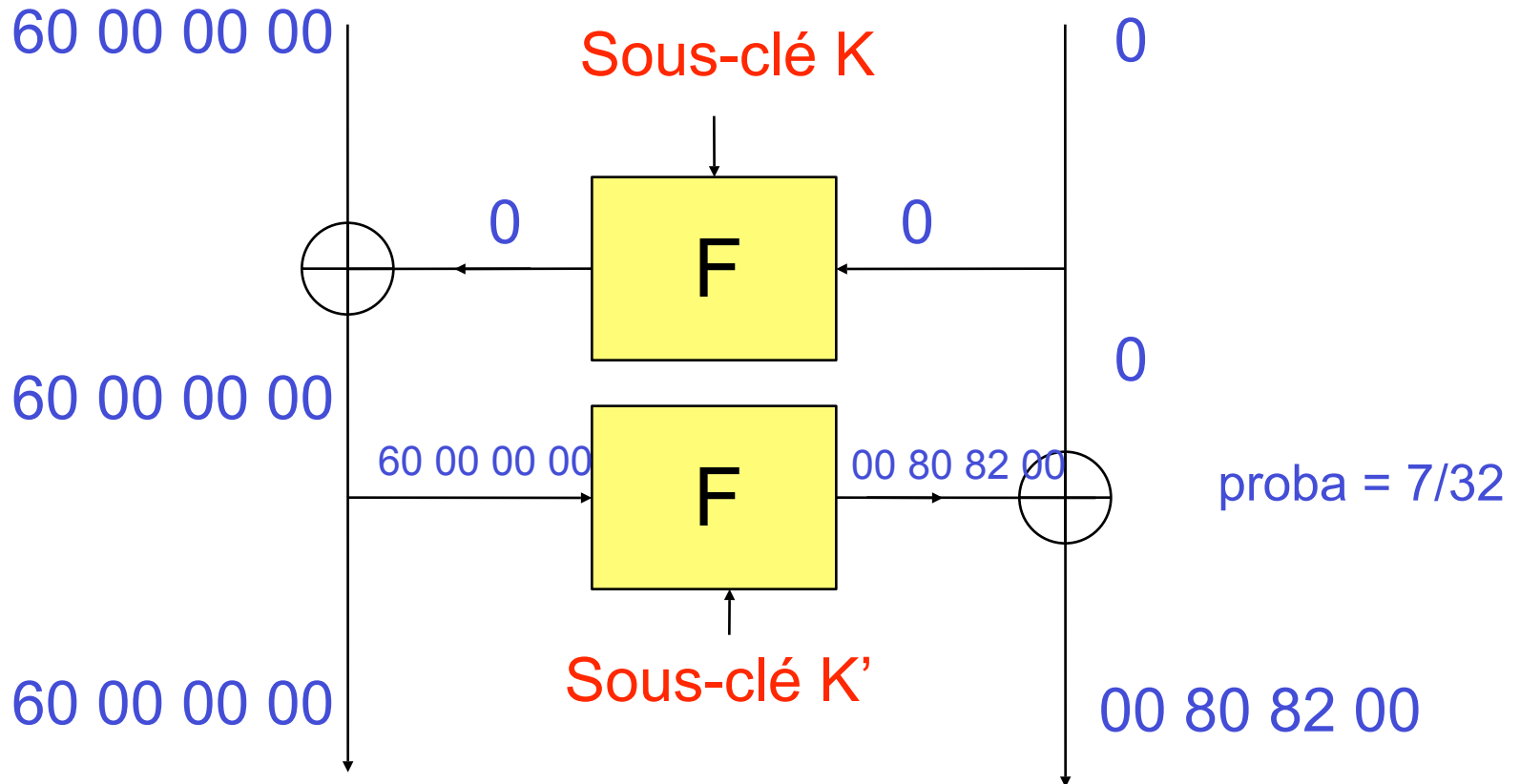


# 2 tours





# 2 tours



$\forall K, K'$  on a  $(60\ 00\ 00\ 00, 0) \rightarrow (60\ 00\ 00\ 00, 00\ 80\ 82\ 00)$  [proba = 7/32]

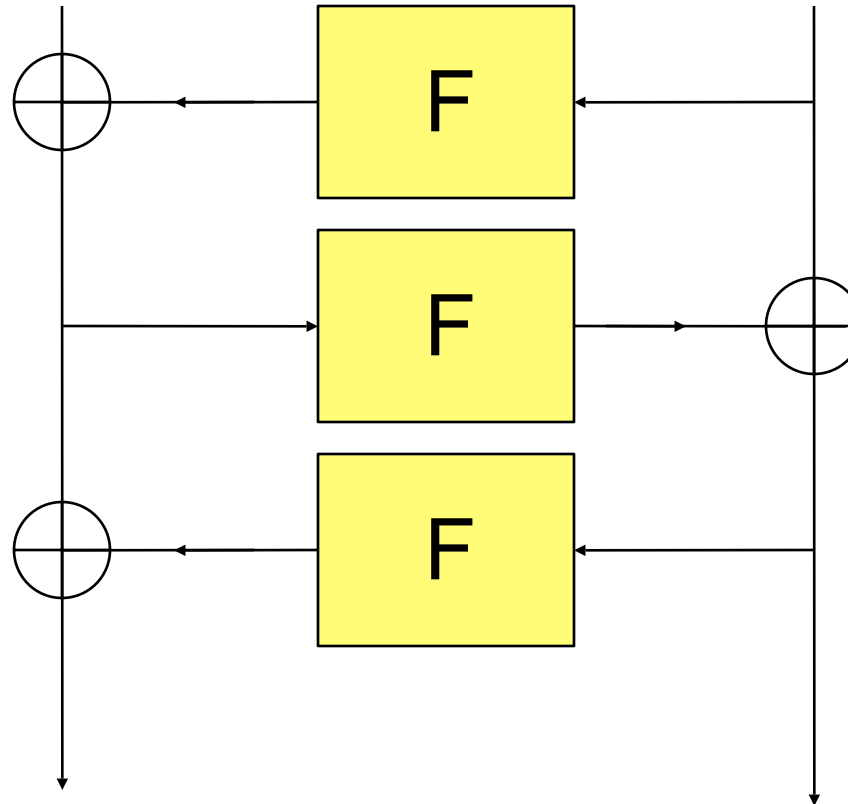
**$\geq 3$  tours**

- L'idée est de combiner les caractéristiques différentielles
- On souhaite conserver une probabilité la plus grande possible

# 3 tours

40 08 00 00

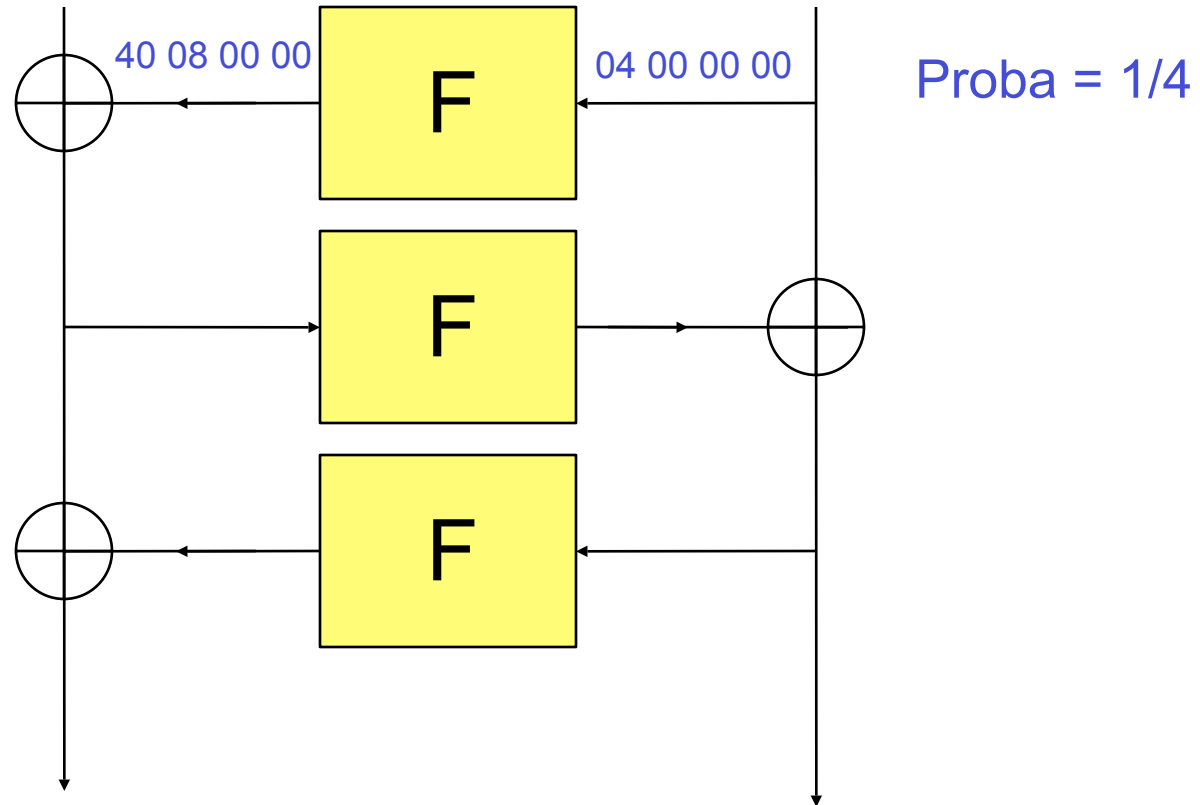
04 00 00 00



# 3 tours

40 08 00 00

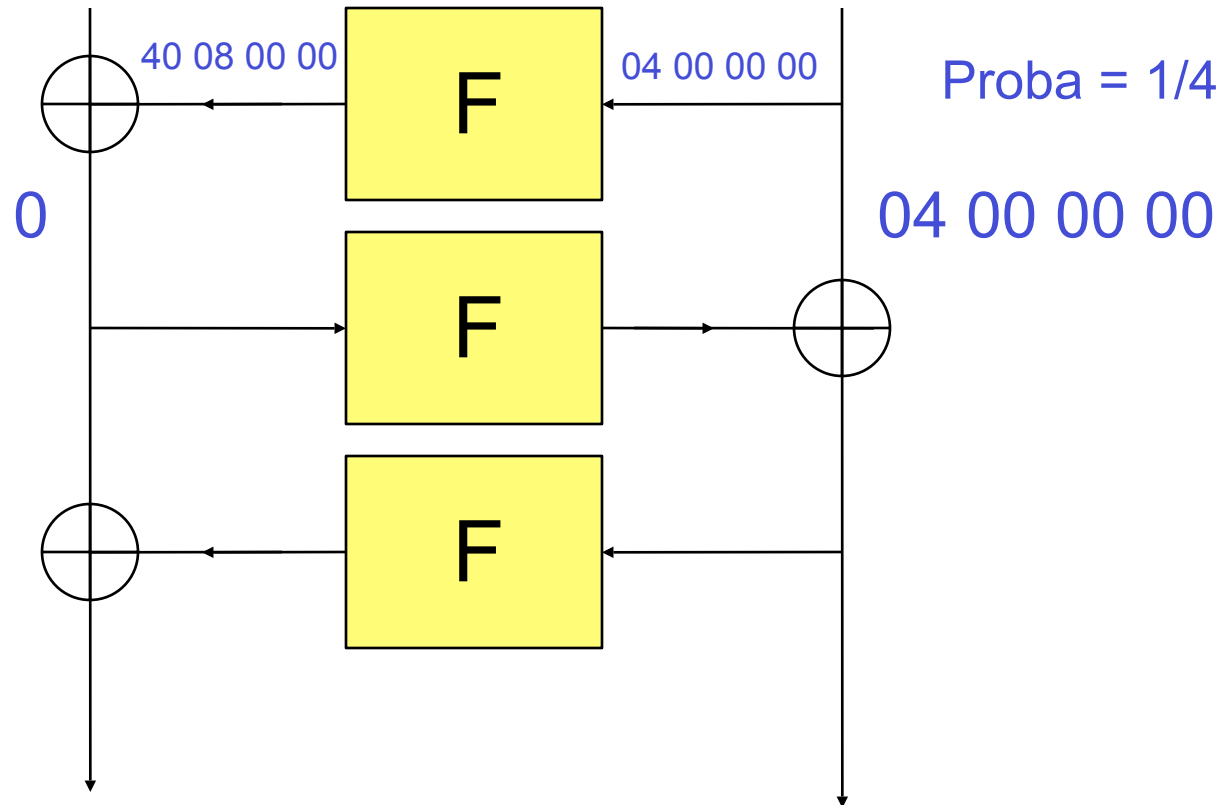
04 00 00 00



# 3 tours

40 08 00 00

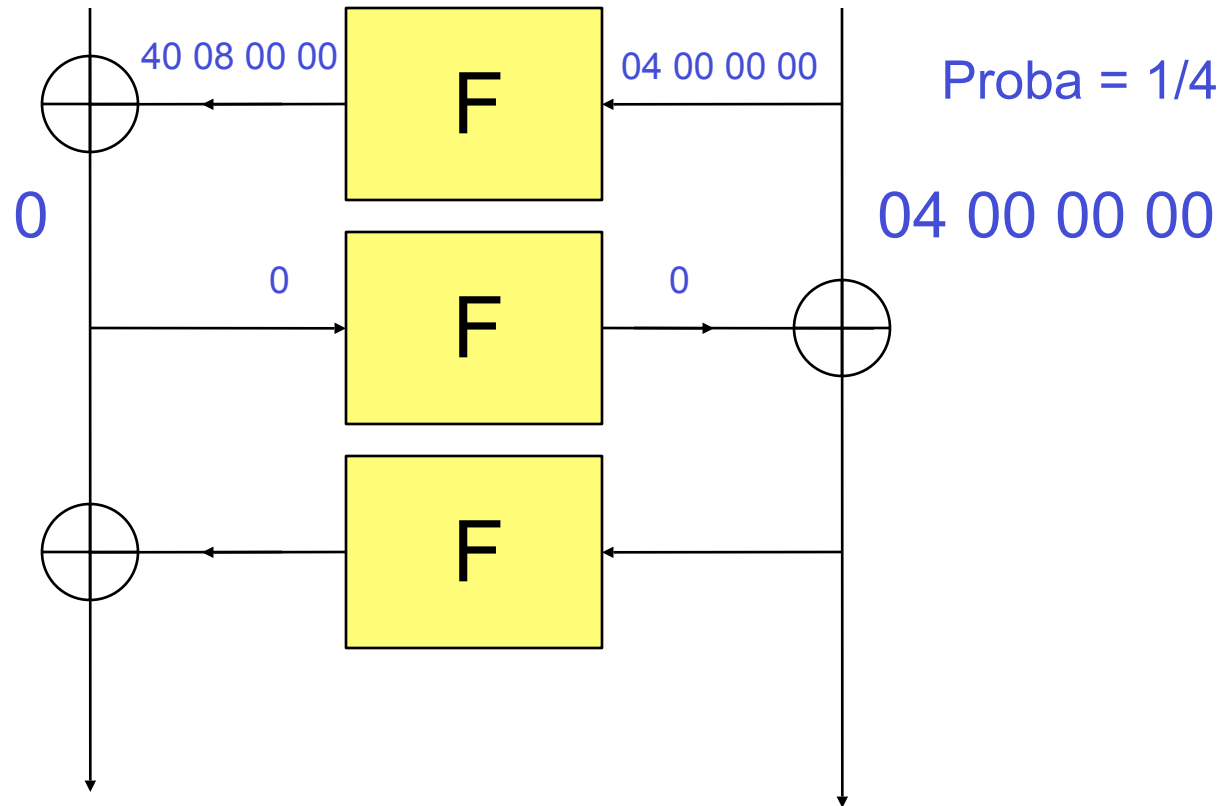
04 00 00 00



# 3 tours

40 08 00 00

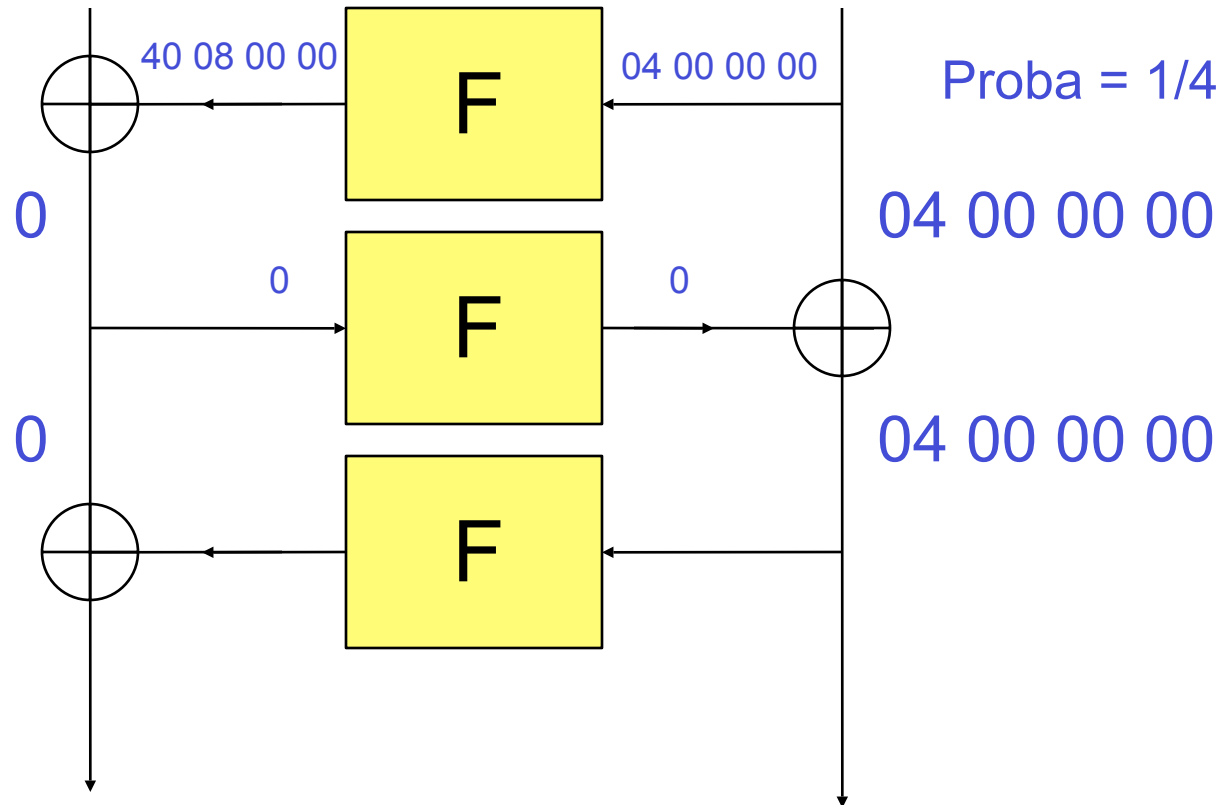
04 00 00 00



# 3 tours

40 08 00 00

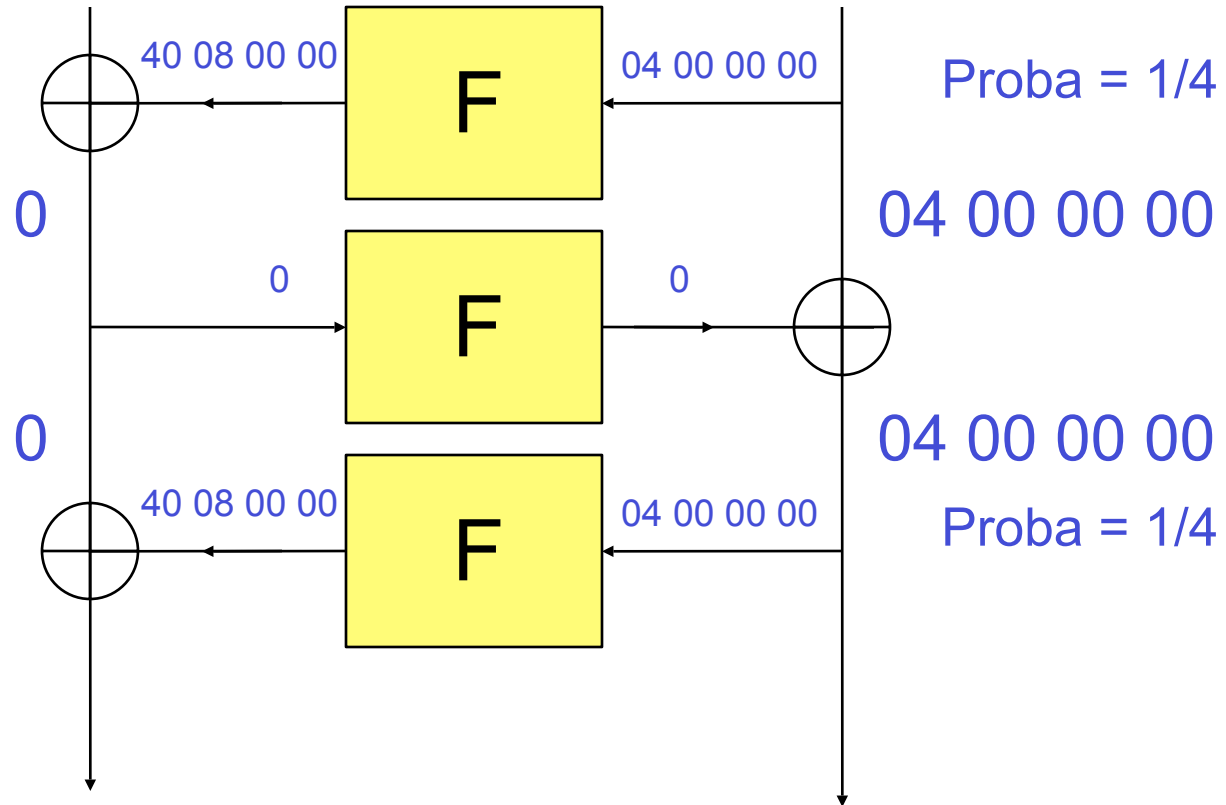
04 00 00 00



# 3 tours

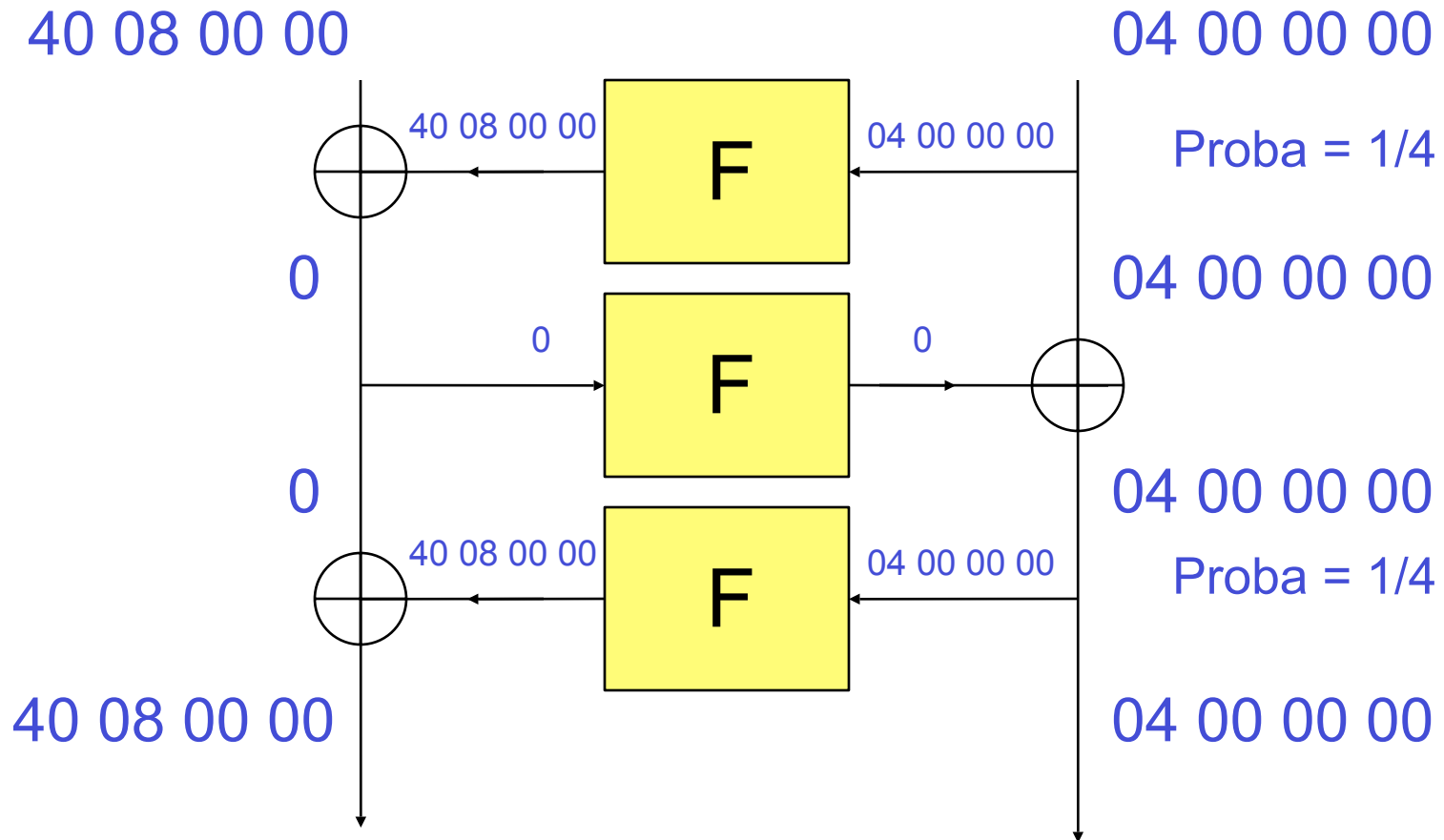
40 08 00 00

04 00 00 00

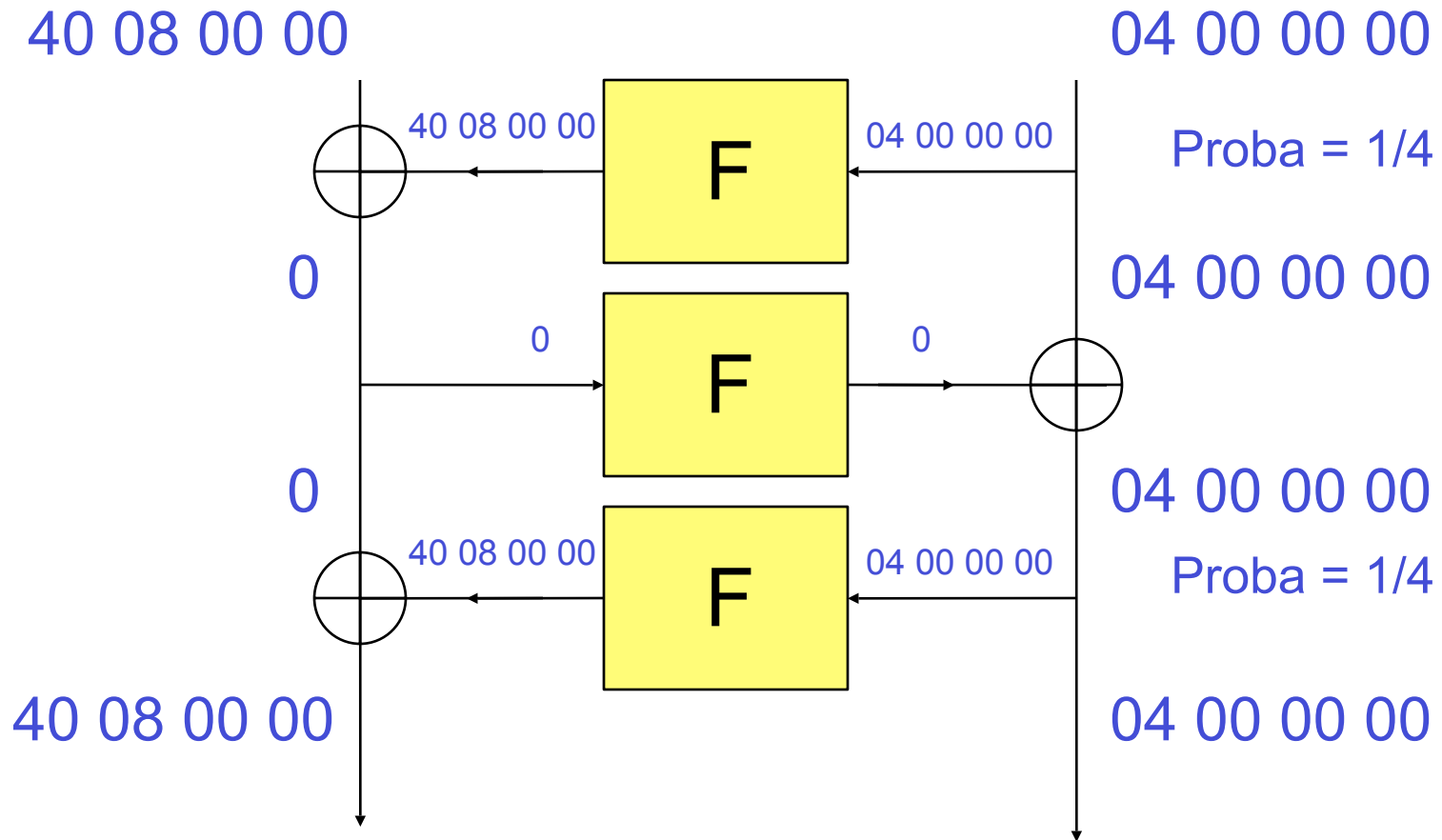




# 3 tours



# 3 tours

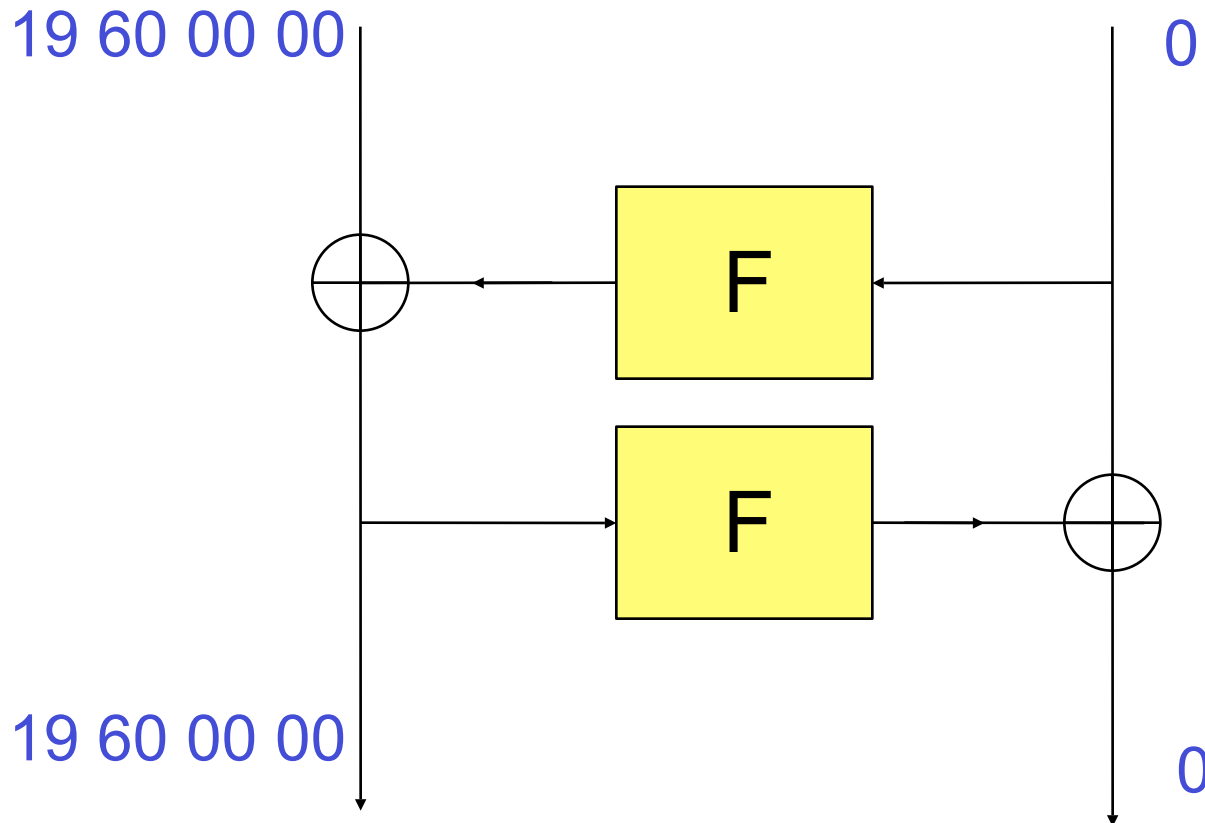


$(40080000, 04000000) \rightarrow (40080000, 04000000)$  [proba = 1/16]

# Caractéristique itérative

- Il s'agit d'une caractéristique sur 1 tour qui se combine avec elle-même
- Typiquement  $\Delta \rightarrow \Delta$
- Pour le DES, on cherche des caractéristiques itératives sur 2 tours, car les tours pairs et impairs sont différents (structure de Feistel)

# 2 tours



Probabilité de cette caractéristique =  $1 / 2^{34}$

# Analyse

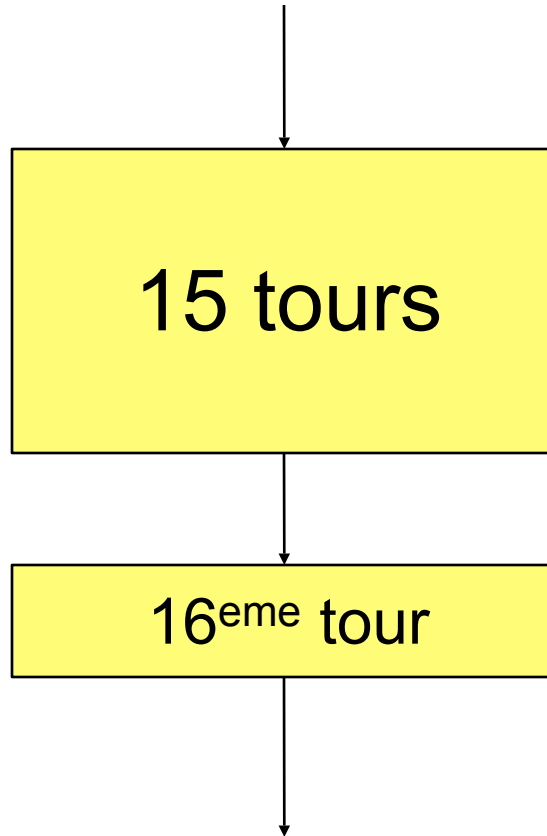
- Probabilité  $1/234$  pour 2 tours  $\rightarrow (1/234)^t$  pour  $2t$  tours
- Ex:  $t = 8 \rightarrow \text{probabilité} = 2^{-62.96}$
- **Attaque en distingueur** :
  - chiffrer  $> 2^{62.96}$  paires de message avec la différence (19 60 00 00, 0) en entrée
  - Observer que la différence (19 60 00 00, 0) apparaît plus fréquemment que les autres en sortie
- Attaque **théorique** et moins bonne que la recherche exhaustive ( $2^{56}$ )

# Retrouver la clé ?

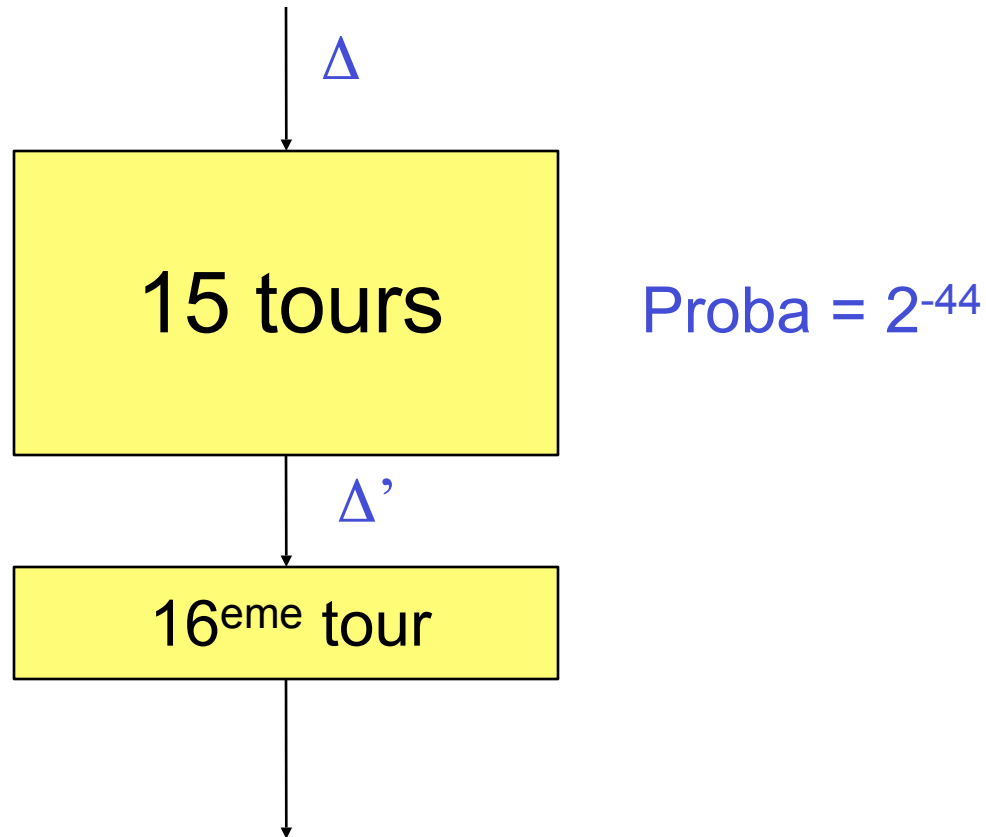
Deux méthodes possibles :

- 1 – Utiliser des « astuces » algorithmiques pour éliminer le premier ou le dernier tour
- 2 – Exploiter le fait que les différences intermédiaires sont connues (→ équations explicites faisant intervenir la clé)

# Optimisations

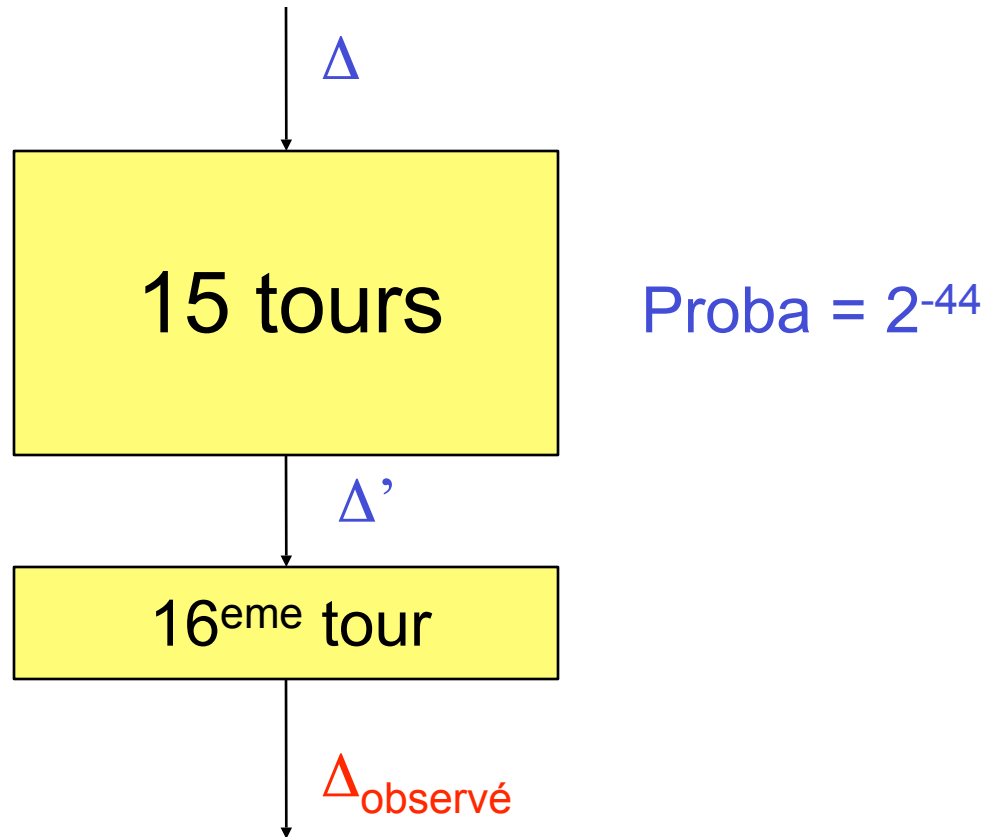


# Optimisations

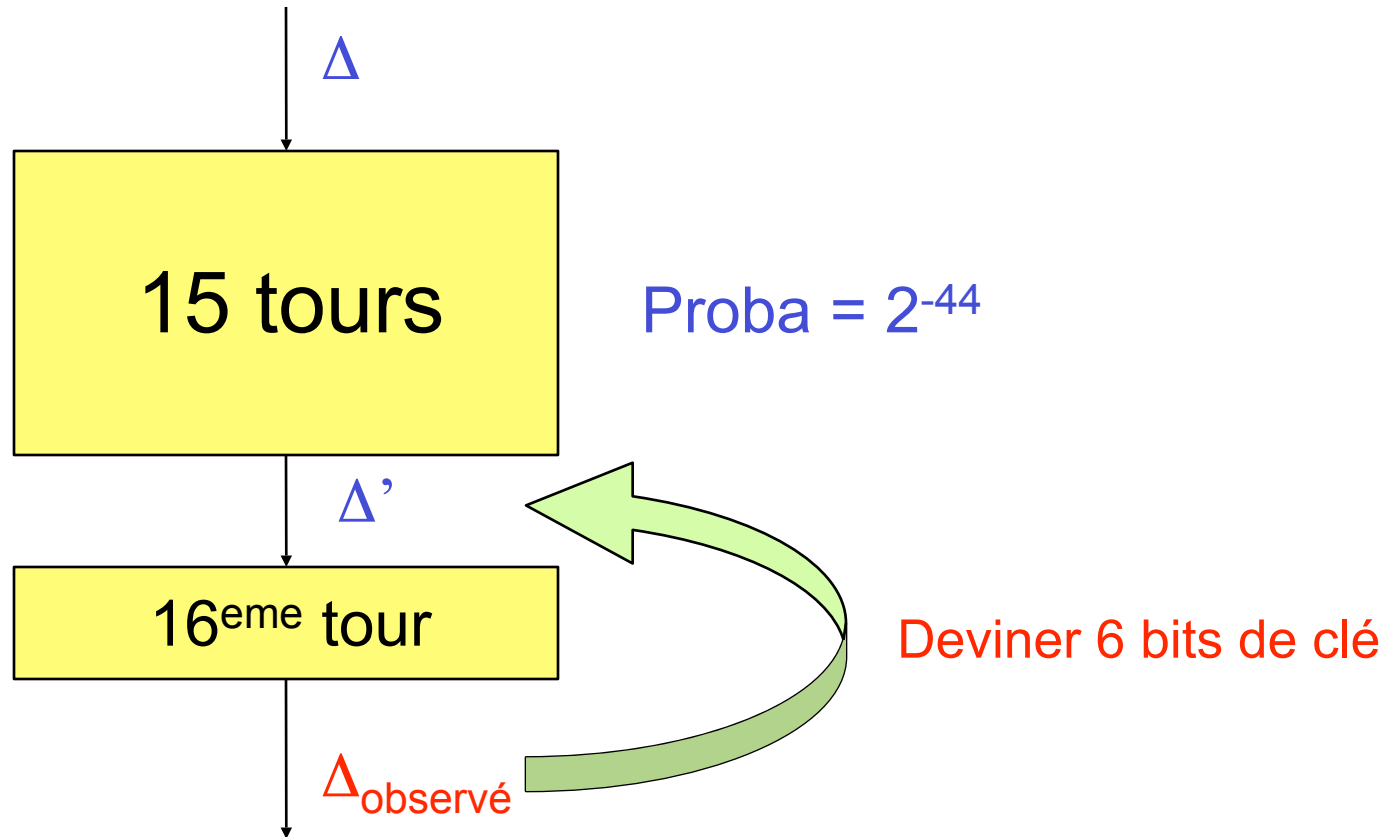




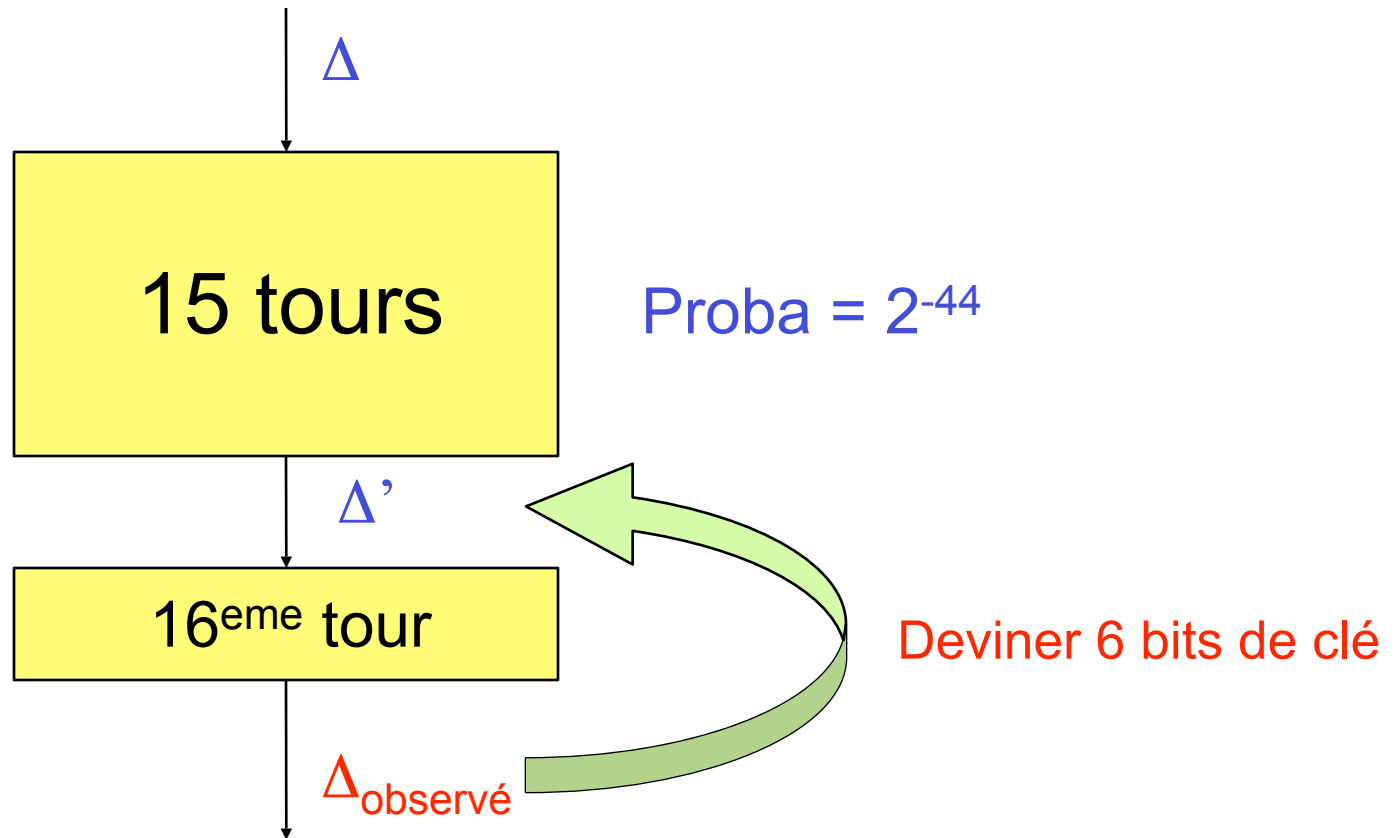
# Optimisations



# Optimisations



# Optimisations



L'attaque coûte  $2^6 \times 2^{44}$

# Historique

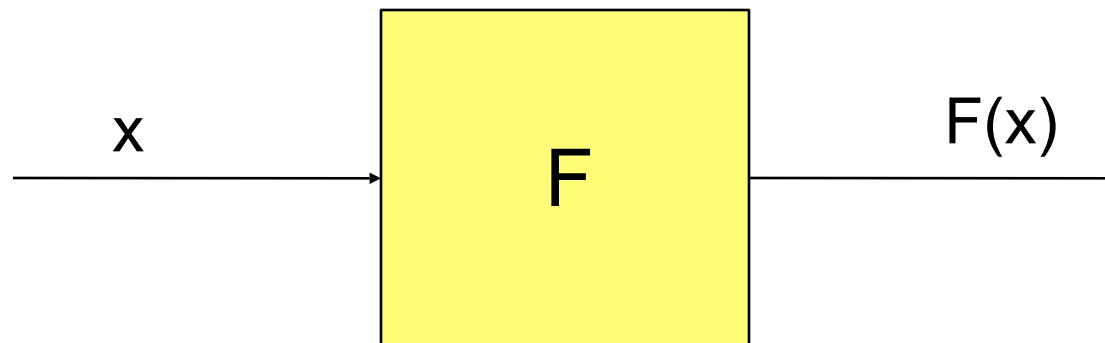
- Biham-Shamir 1990
  - Invention de la cryptanalyse différentielle
  - Observation : ne fonctionne pas pour le DES
- Biham-Shamir 1992
  - Optimisations (utiliser 13 tours seulement)
  - Attaque pour retrouver la clé
  - Complexité  $2^{47}$  messages choisis
  - Inutile en pratique

# Cryptanalyse linéaire

- Idée générale proche de la cryptanalyse différentielle
- On utilise des approximations linéaires des algorithmes de chiffrement par bloc

# Forme linéaire

- Soit  $F : \{0,1\}^n \rightarrow \{0,1\}^n$
- Une **forme linéaire**  $\lambda : \{0,1\}^n \rightarrow \{0,1\}$  est définie par un **masque**  $a = (a_1 \dots a_n)$
- $\lambda(x_1 \dots x_n) = a_1 \cdot x_1 \oplus \dots \oplus a_n \cdot x_n$

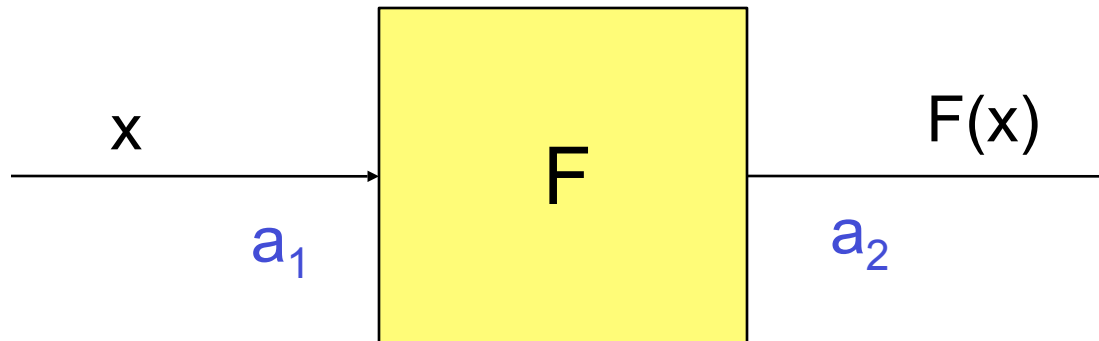


# Caractéristique linéaire

- Une **caractéristique linéaire** de  $F$  est un couple de **formes linéaires**  $(\lambda_1, \lambda_2)$  ayant pour **masques** associés  $a_1$  et  $a_2$  telles que

$$\lambda_1(x) = \lambda_2(F(x))$$

- avec probabilité  $p$  (prise sur tous les  $x$  possibles)



# Notation

- Par analogie avec la cryptanalyse différentielle, on note souvent

$$a_1 \rightarrow a_2 \text{ [proba} = p\text{]}$$

- On utilise une table des caractéristiques linéaires ( $\Leftrightarrow$  table des différences)



# Table des app. linéaires

$$F : \{0,1\}^2 \rightarrow \{0,1\}^2 : F(x,y) = (x.y, x)$$

$a_1 \backslash a_2$	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	0,5	0,75	0,75
(0,1)	0,5	0,5	0,75	0,25
(1,0)	0,5	1	0,75	0,75
(1,1)	0,5	0,5	0,25	0,75

(1,0)  $\rightarrow$  (0,1) [proba = 1]

# Table des app. linéaires

- En général, deux formes linéaires aléatoires sont égales avec probabilité 0,5
- On s'intéresse donc à l'écart avec  $p = 0,5$  aussi appelé **biais  $\varepsilon$**

$$a_1 \rightarrow a_2 \quad [\text{proba} = 0.5 * (1 + \varepsilon)]$$

$$a_1 \rightarrow a_2 \quad [\text{biais} = \varepsilon]$$

- Plus  **$|\varepsilon|$  est grand**, mieux c'est pour l'attaquant

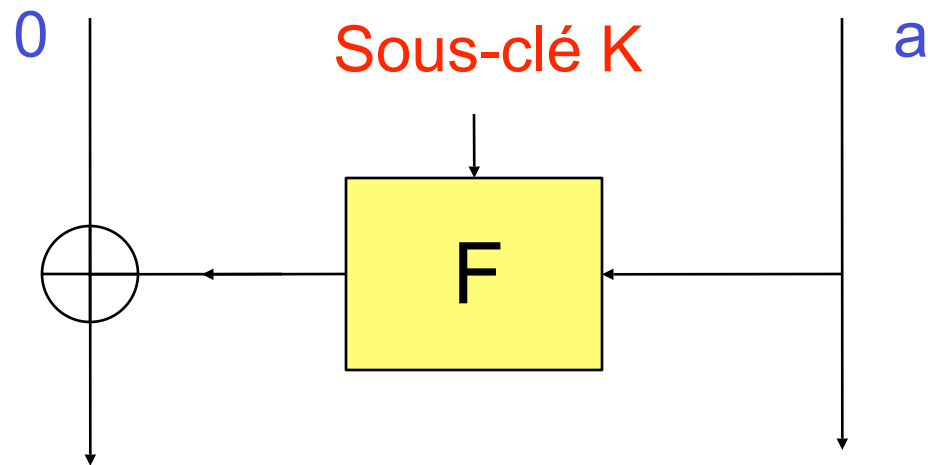
# Comment la calculer ?

- De façon naïve
  - Énumération de tous les masques et de toutes les entrées possibles de F
  - Complexité en  $2^{3n}$
- De façon plus subtile
  - Transformée de Walsh (= FFT dans le cas des fonctions  $\{0,1\}^n \rightarrow \{0,1\}^{n'}$ )
  - Complexité en  $2^{2n} * \log(n)$
  - Cela reste inapplicable pour le block cipher entier

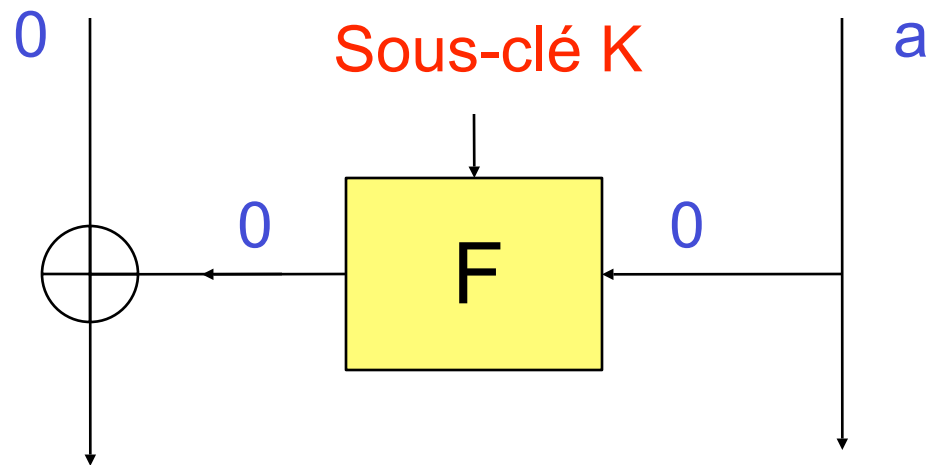
# Attaque contre le DES

- On cherche à combiner des caractéristiques linéaires sur plusieurs tours
- Analogie avec la cryptanalyse différentielle
- Les règles de combinaison des masques linéaires  $\neq$  combinaison des différentielles !

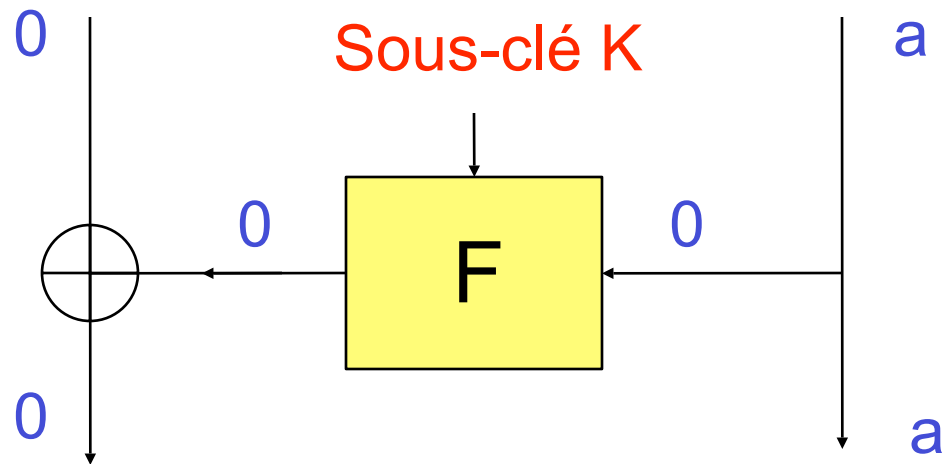
# 1 tour



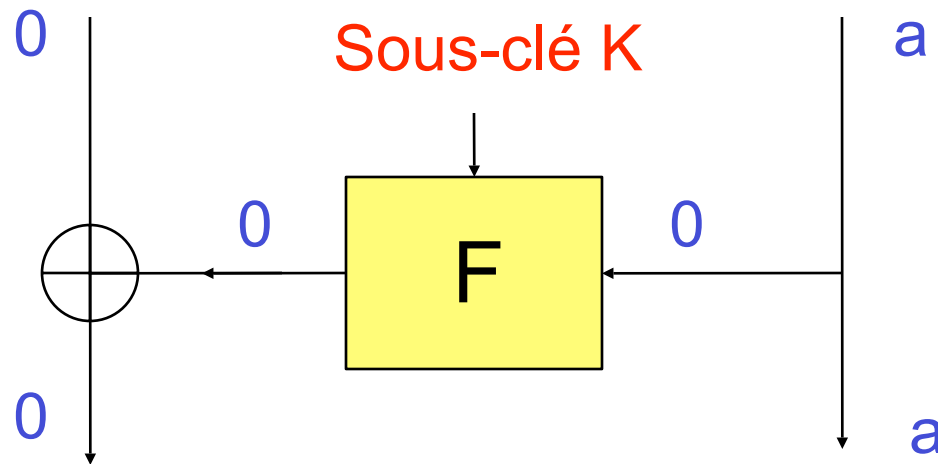
# 1 tour



# 1 tour



# 1 tour



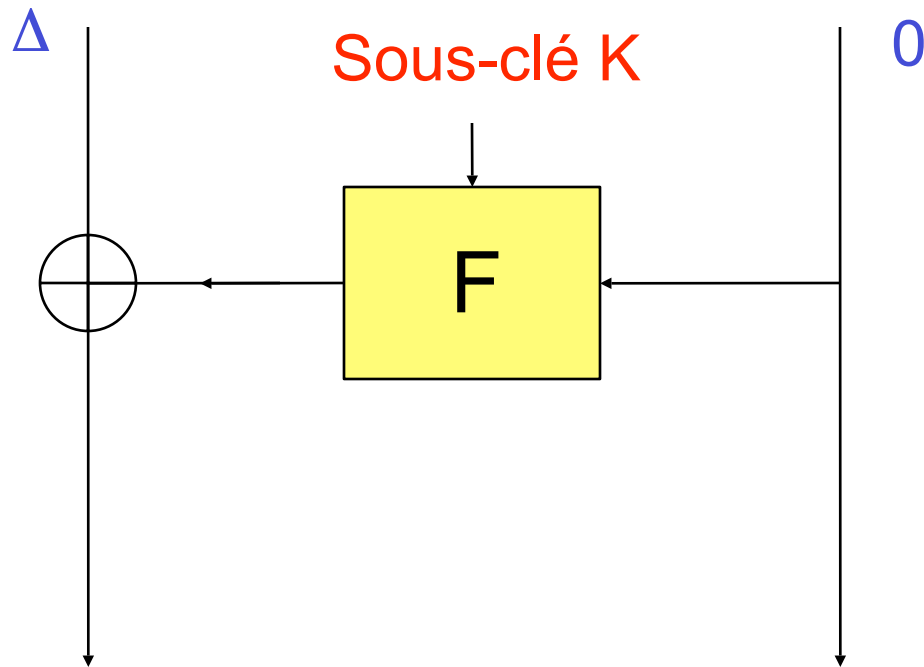
Donc  $\forall a$  et  $\forall K$ , on a :

$(0,a) \rightarrow (0,a)$  [proba = 1]

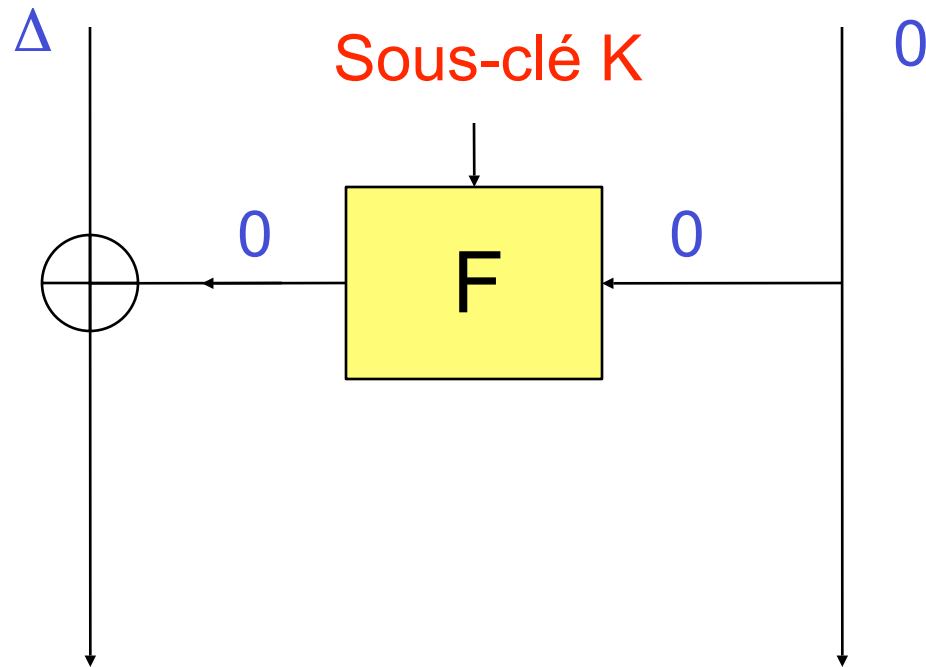
$(0,a) \rightarrow (0,a)$  [biais = 1]



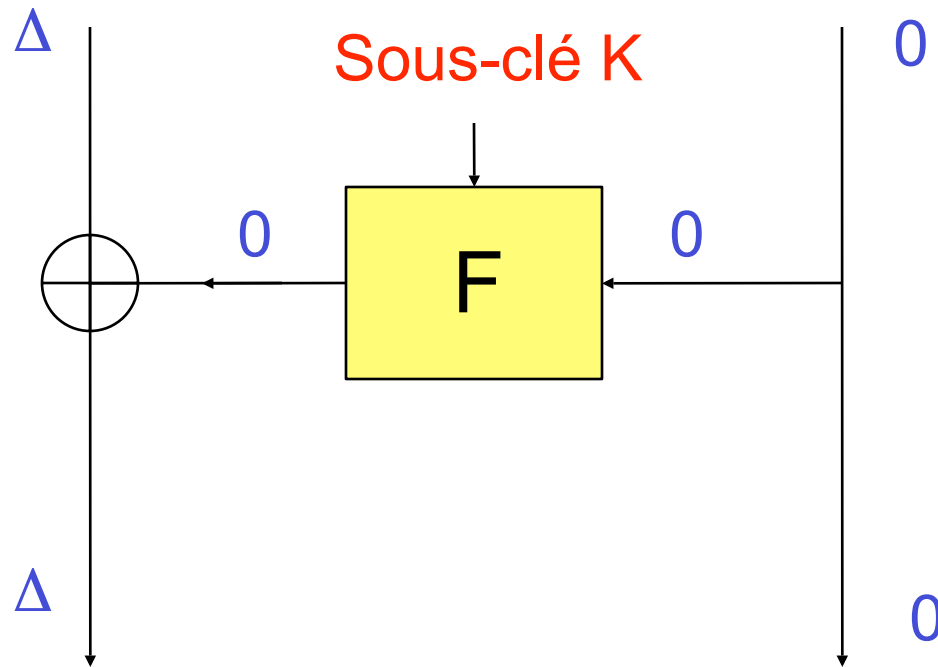
# Rappel : différentielle



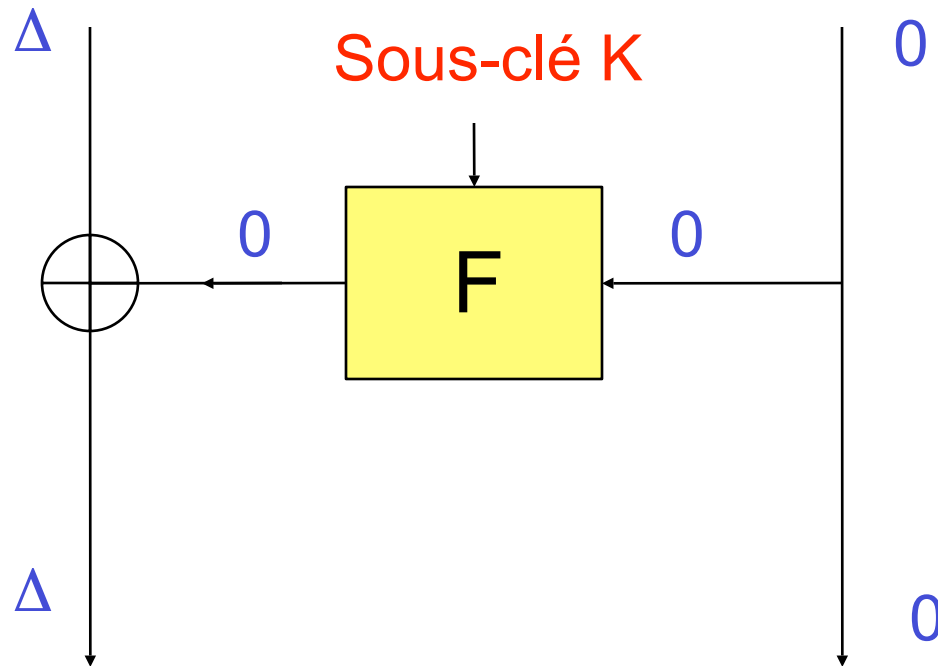
# Rappel : différentielle



# Rappel : différentielle



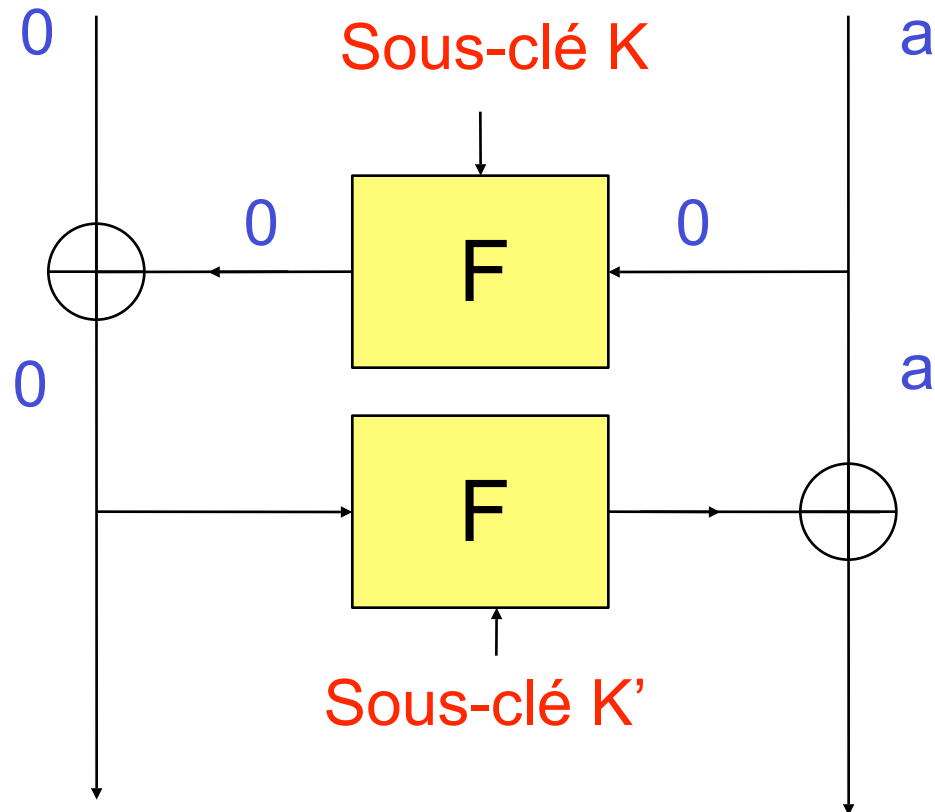
# Rappel : différentielle



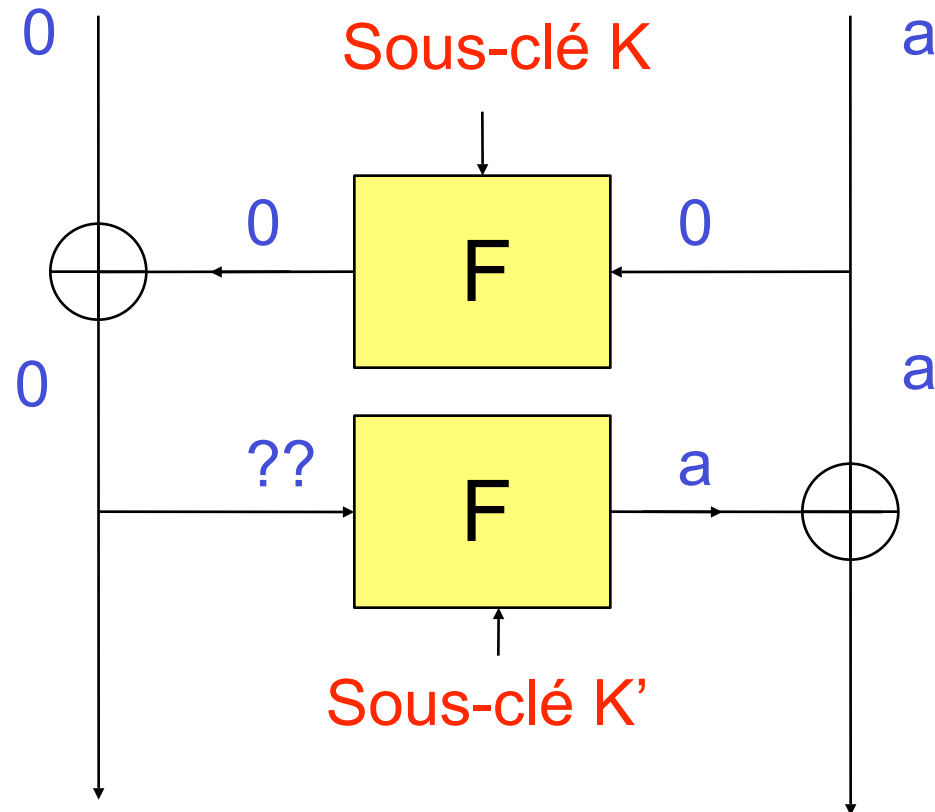
Donc  $\forall \Delta$  et  $\forall K$ , on a :

$$(\Delta, 0) \rightarrow (\Delta, 0) \quad [\text{proba} = 1]$$

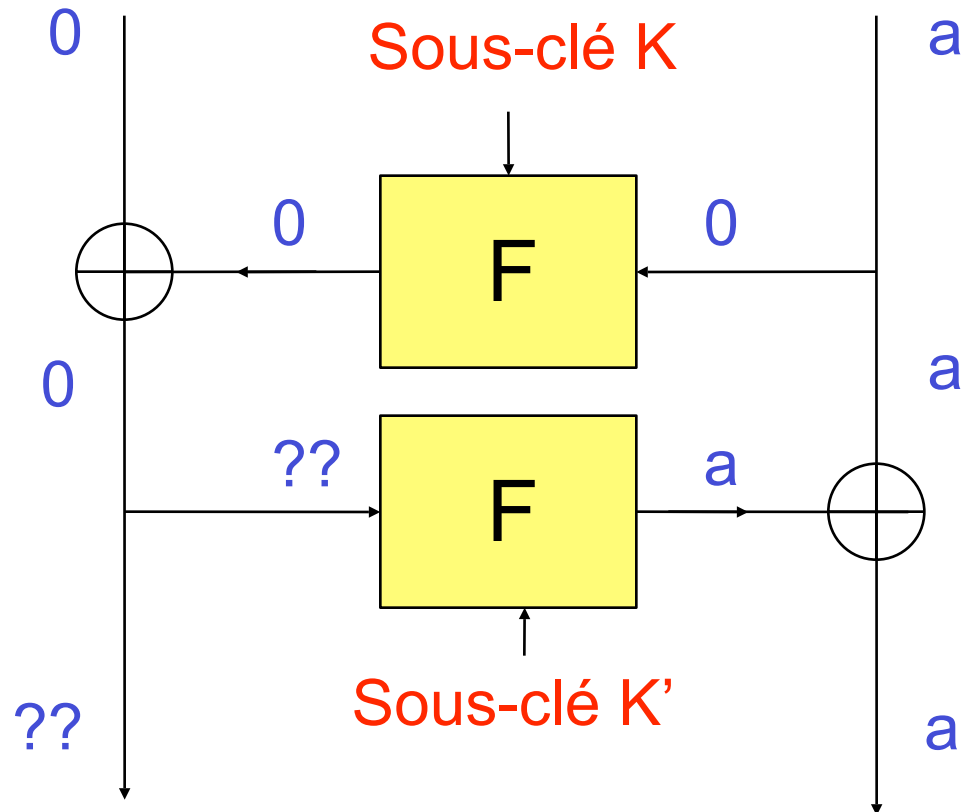
# 2 tours



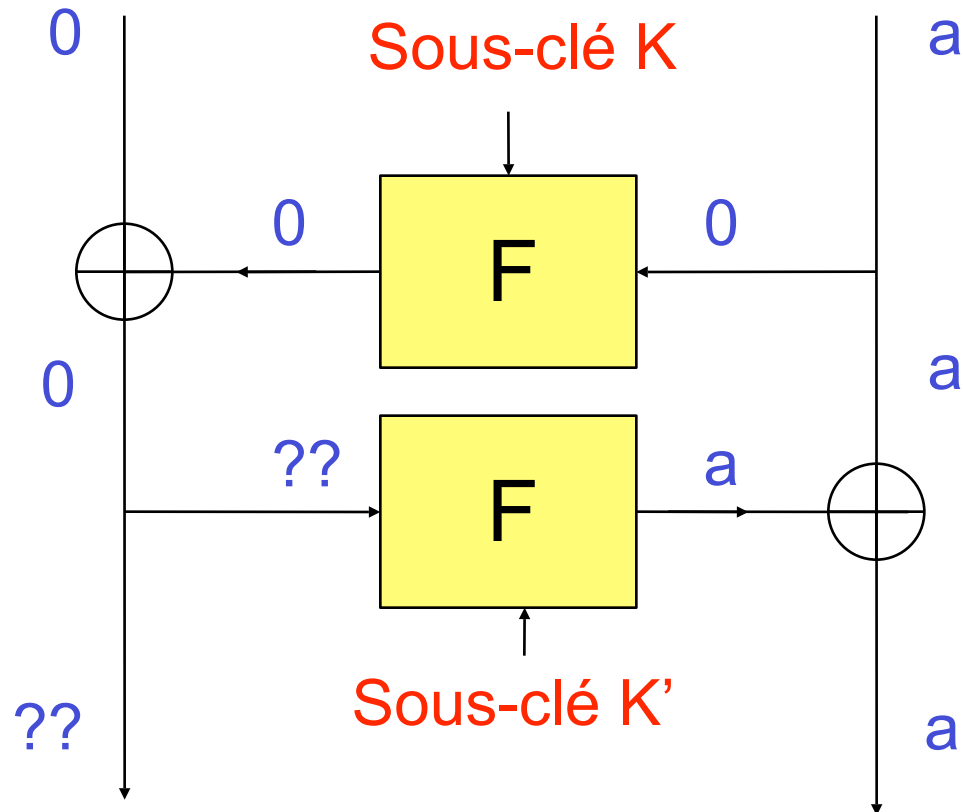
# 2 tours



# 2 tours



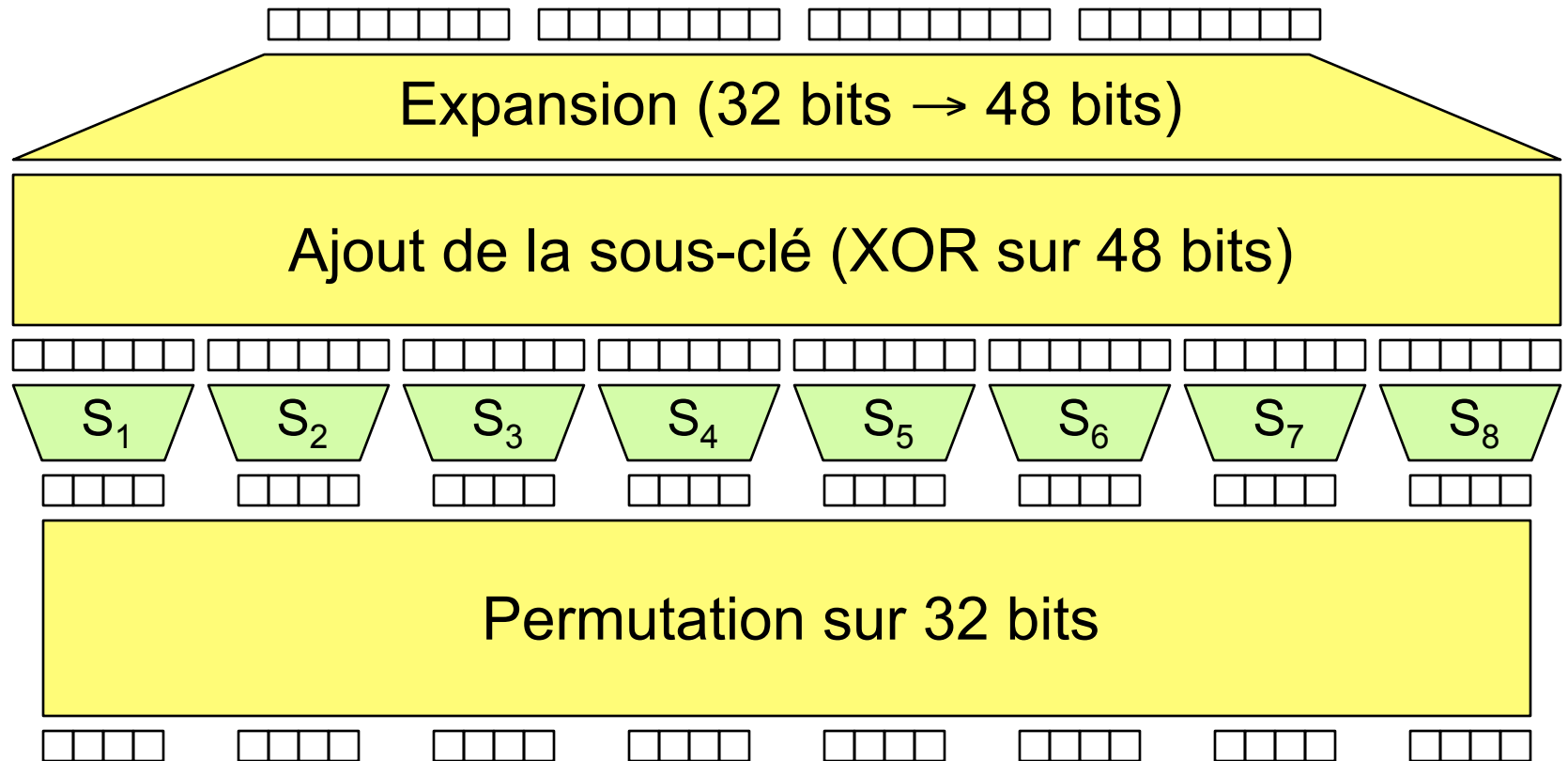
# 2 tours



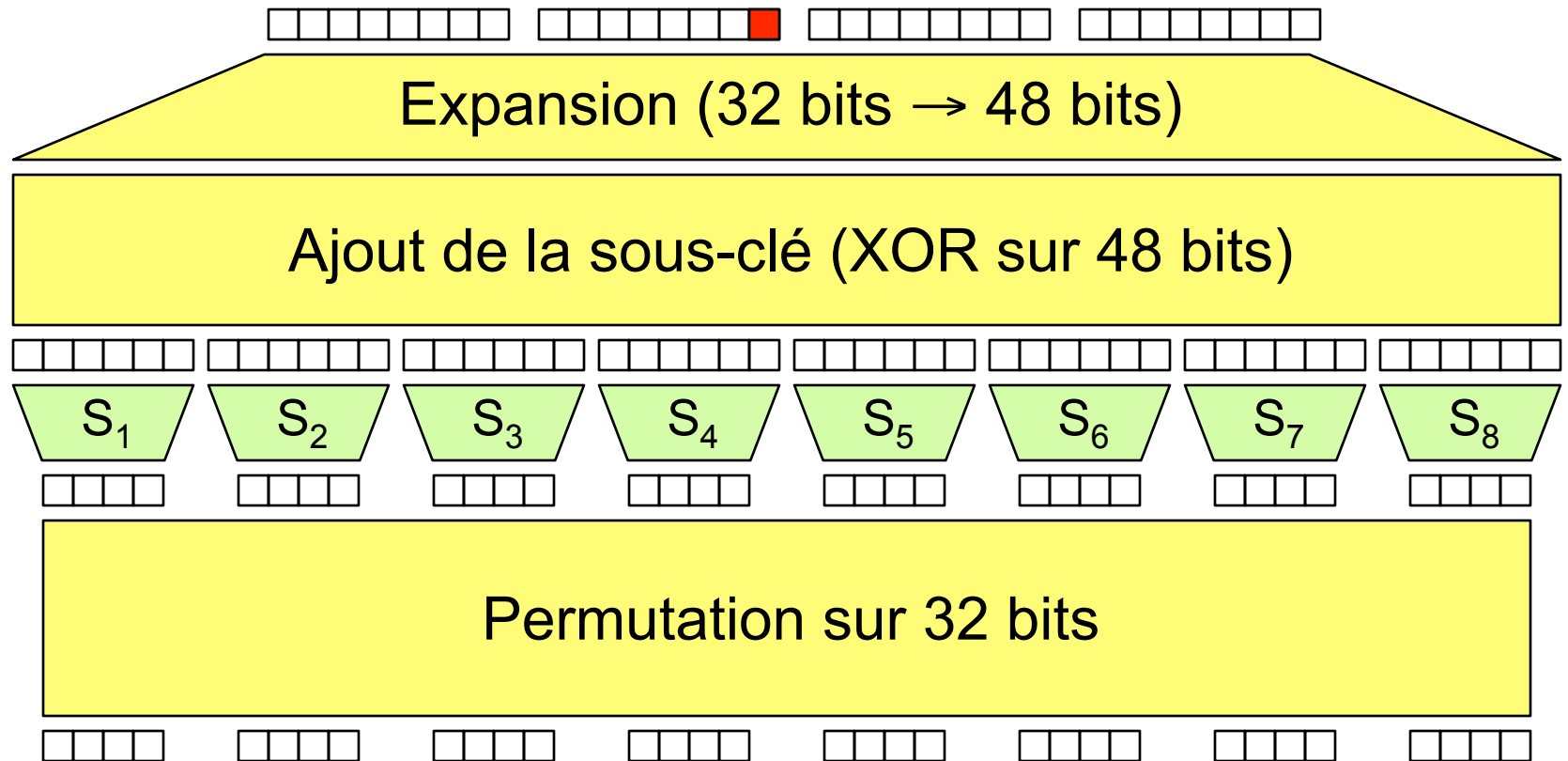
Nécessite d'étudier F plus en détails



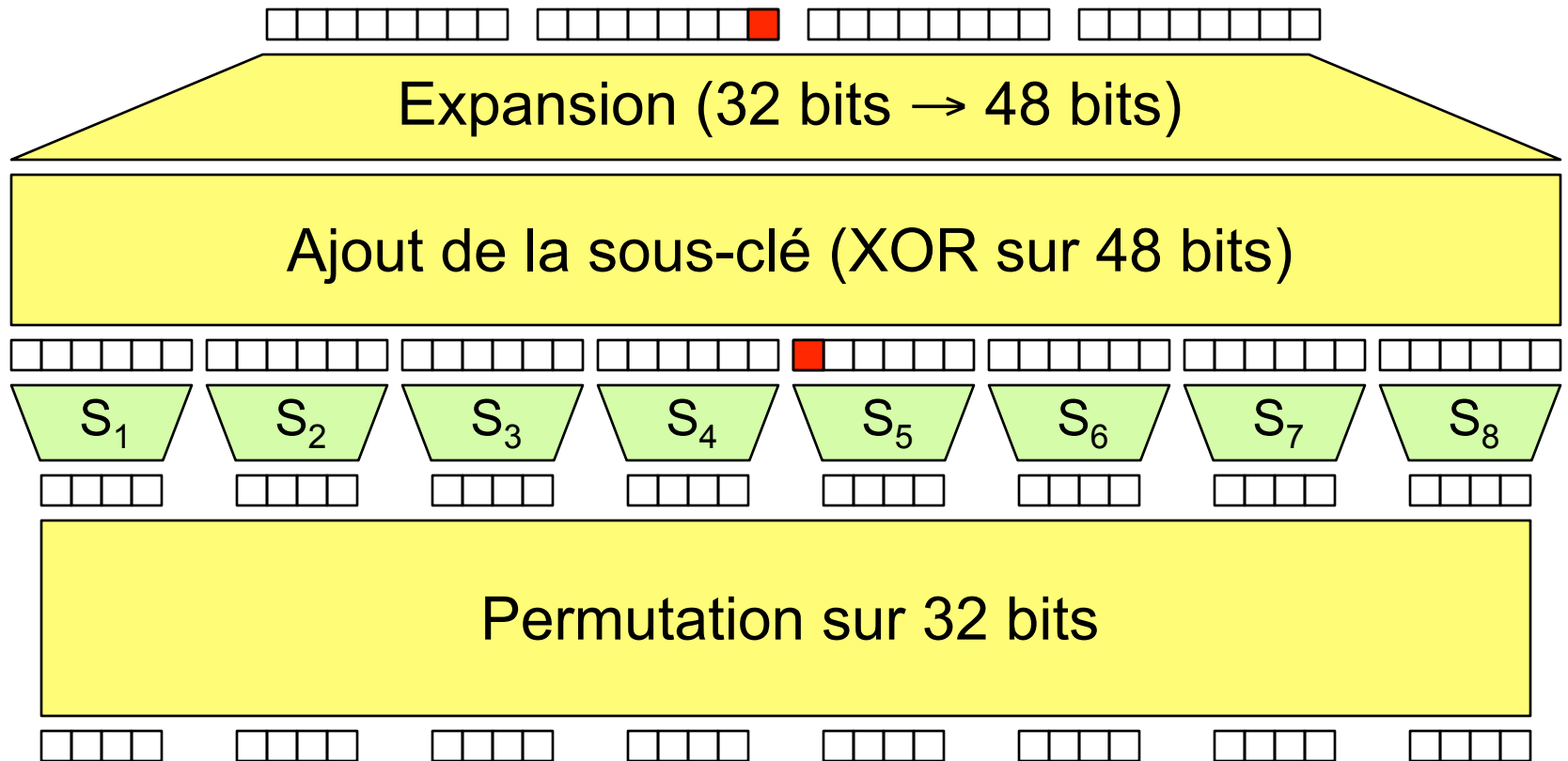
# App. Linéaire de F



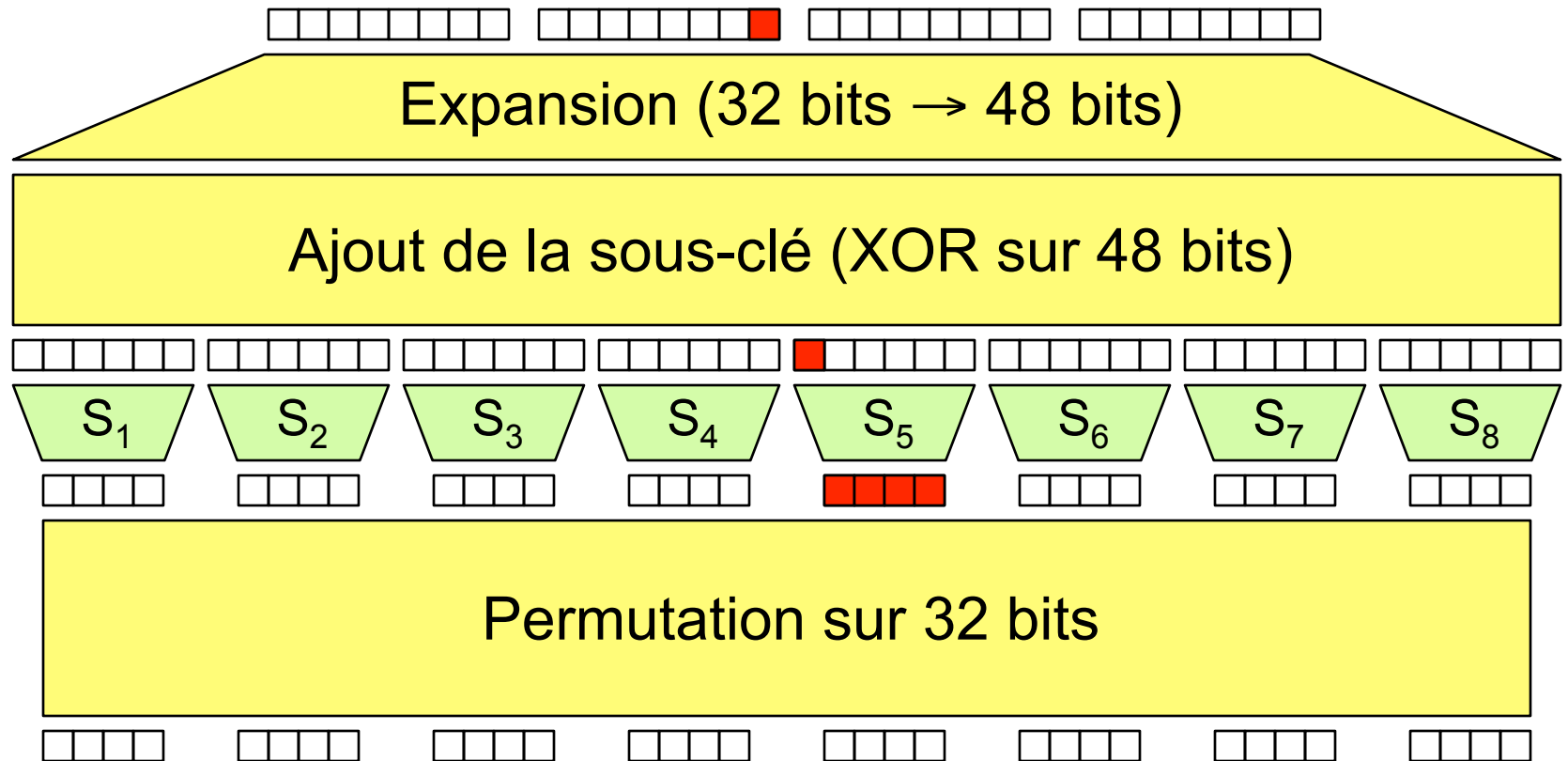
# App. Linéaire de F



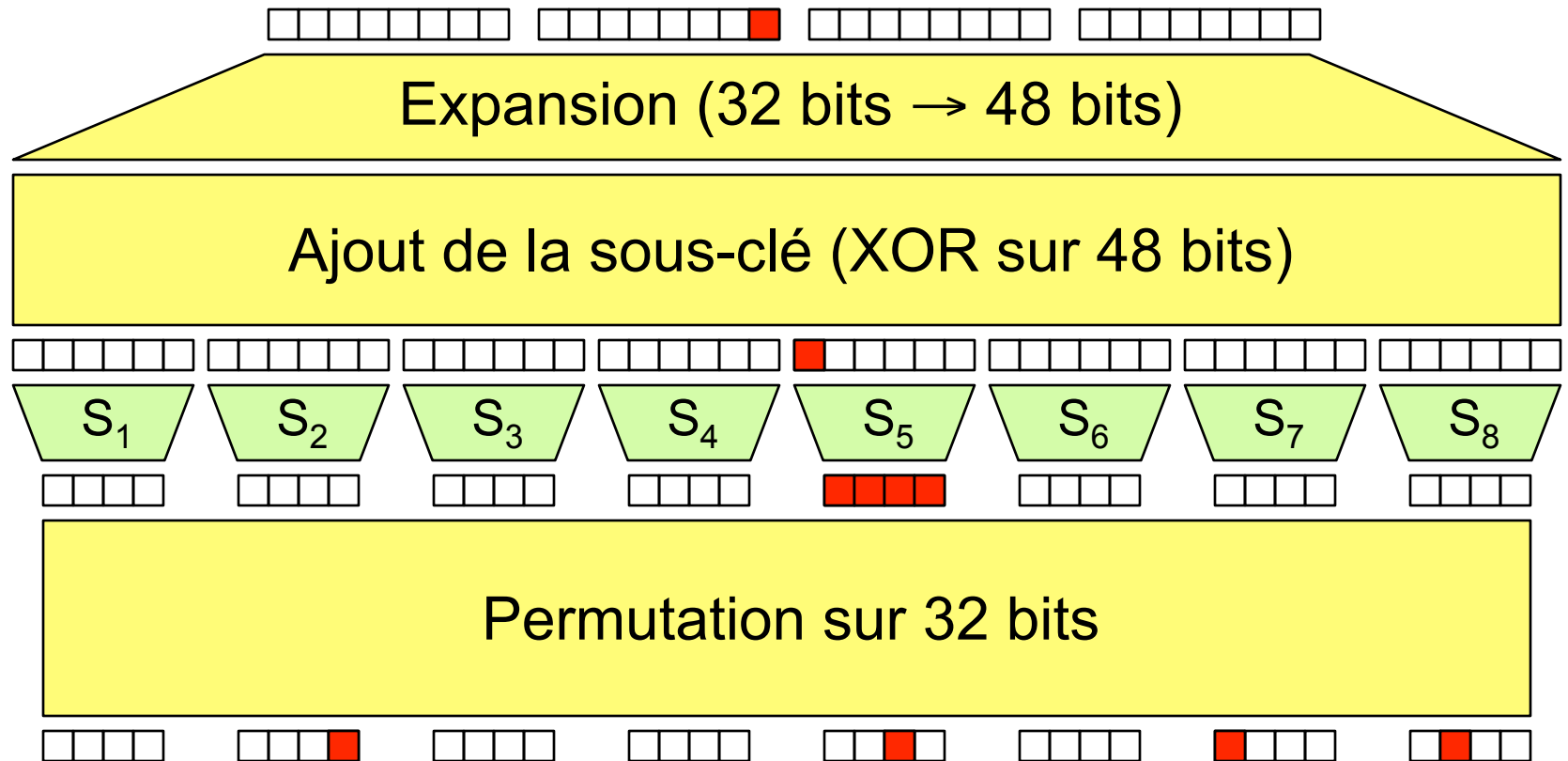
# App. Linéaire de F



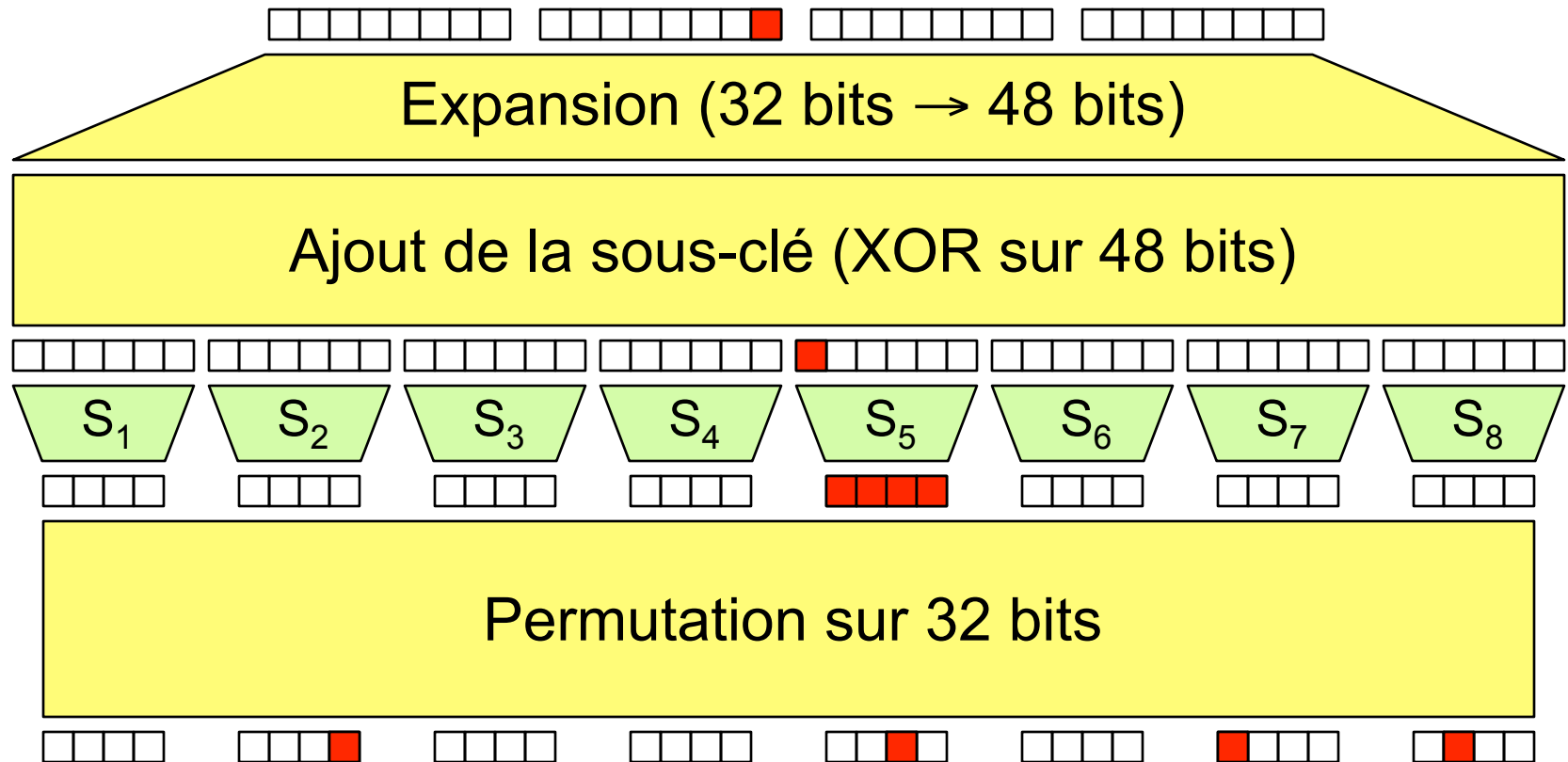
# App. Linéaire de F



# App. Linéaire de F



# App. Linéaire de F

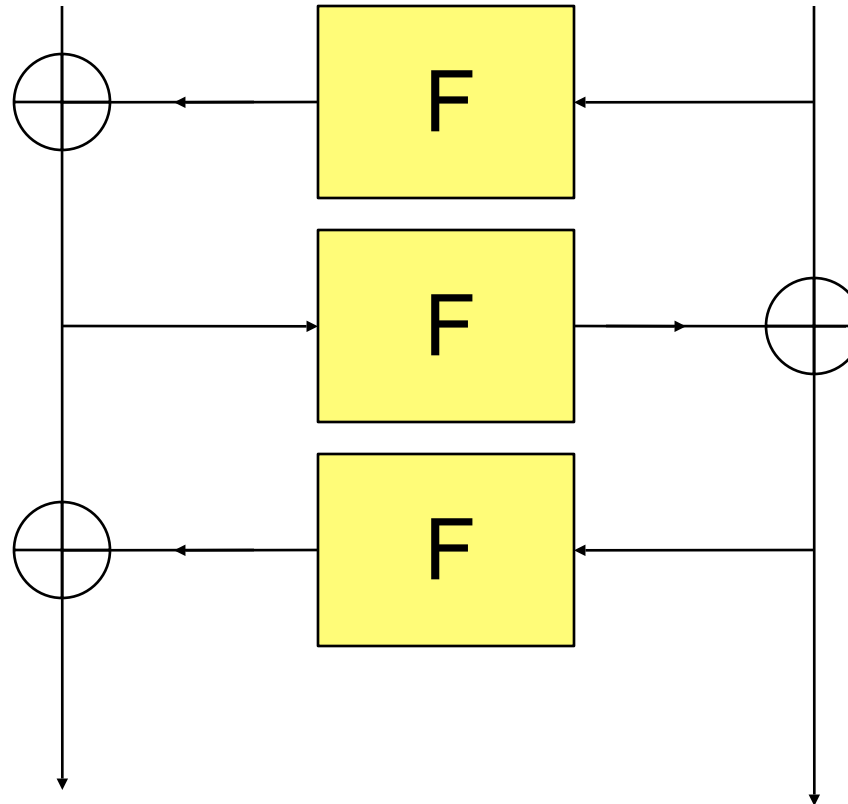


Donc : 00 08 00 00 → 21 04 00 80 avec biais -5/8

# 3 tours

21 04 00 80

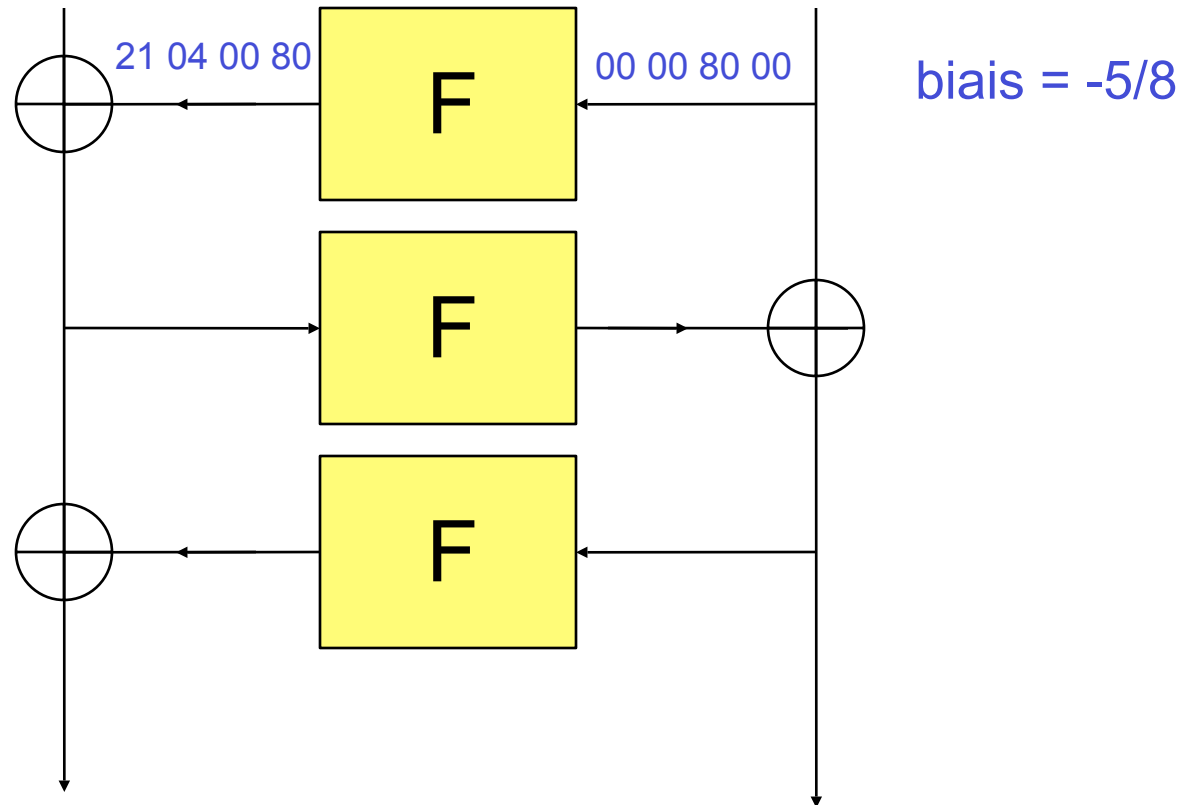
00 00 80 00



# 3 tours

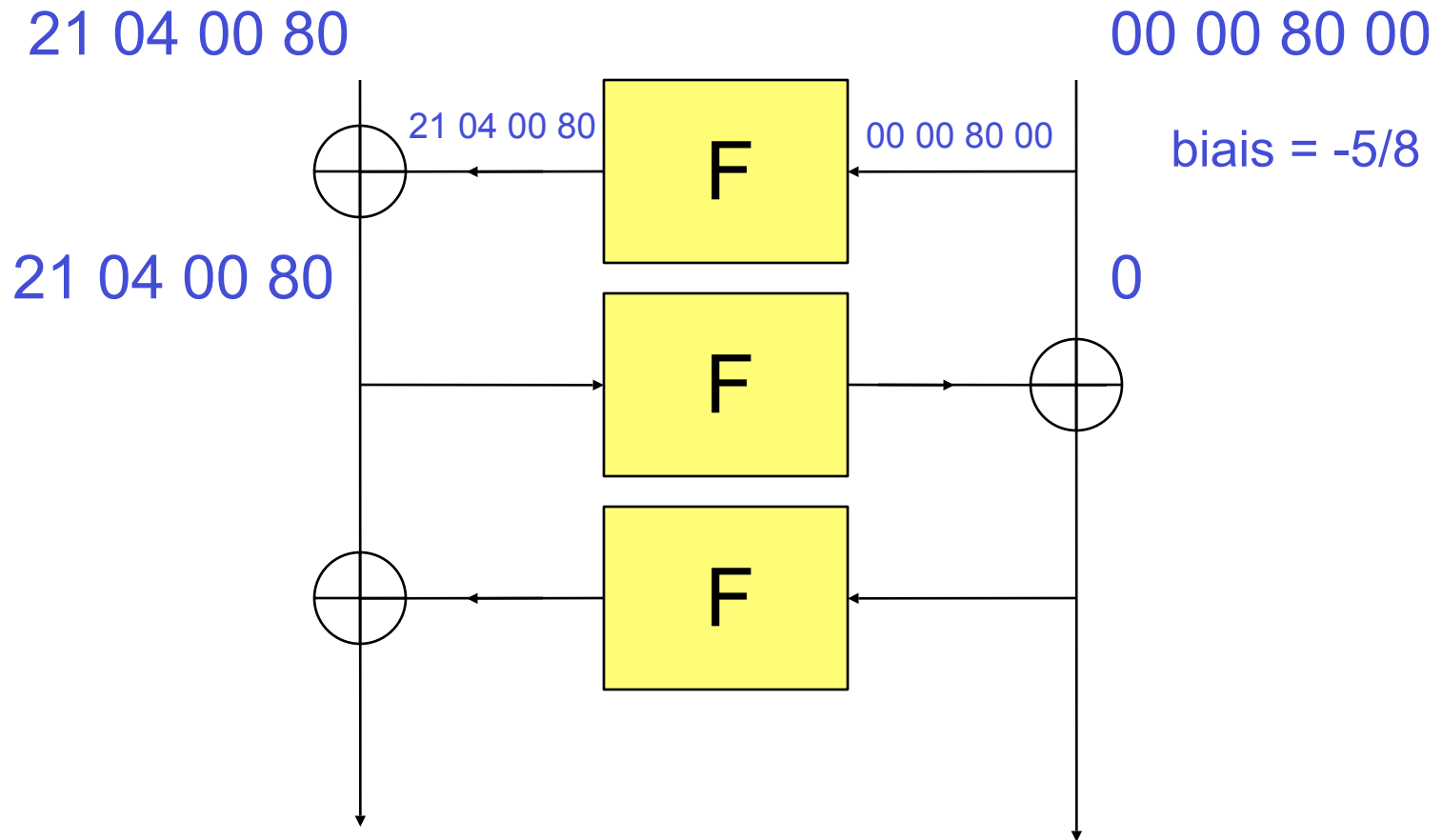
21 04 00 80

00 00 80 00

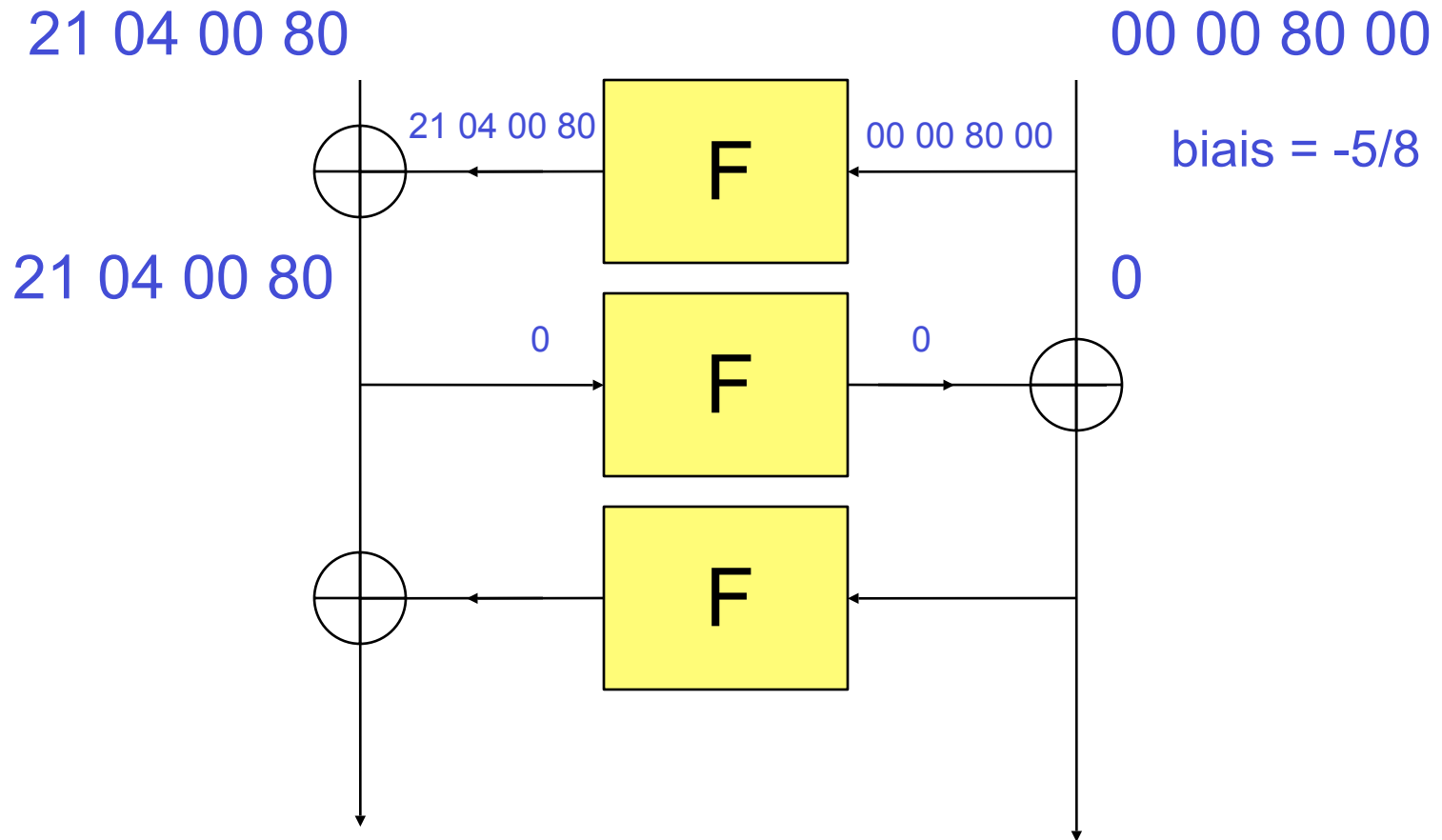




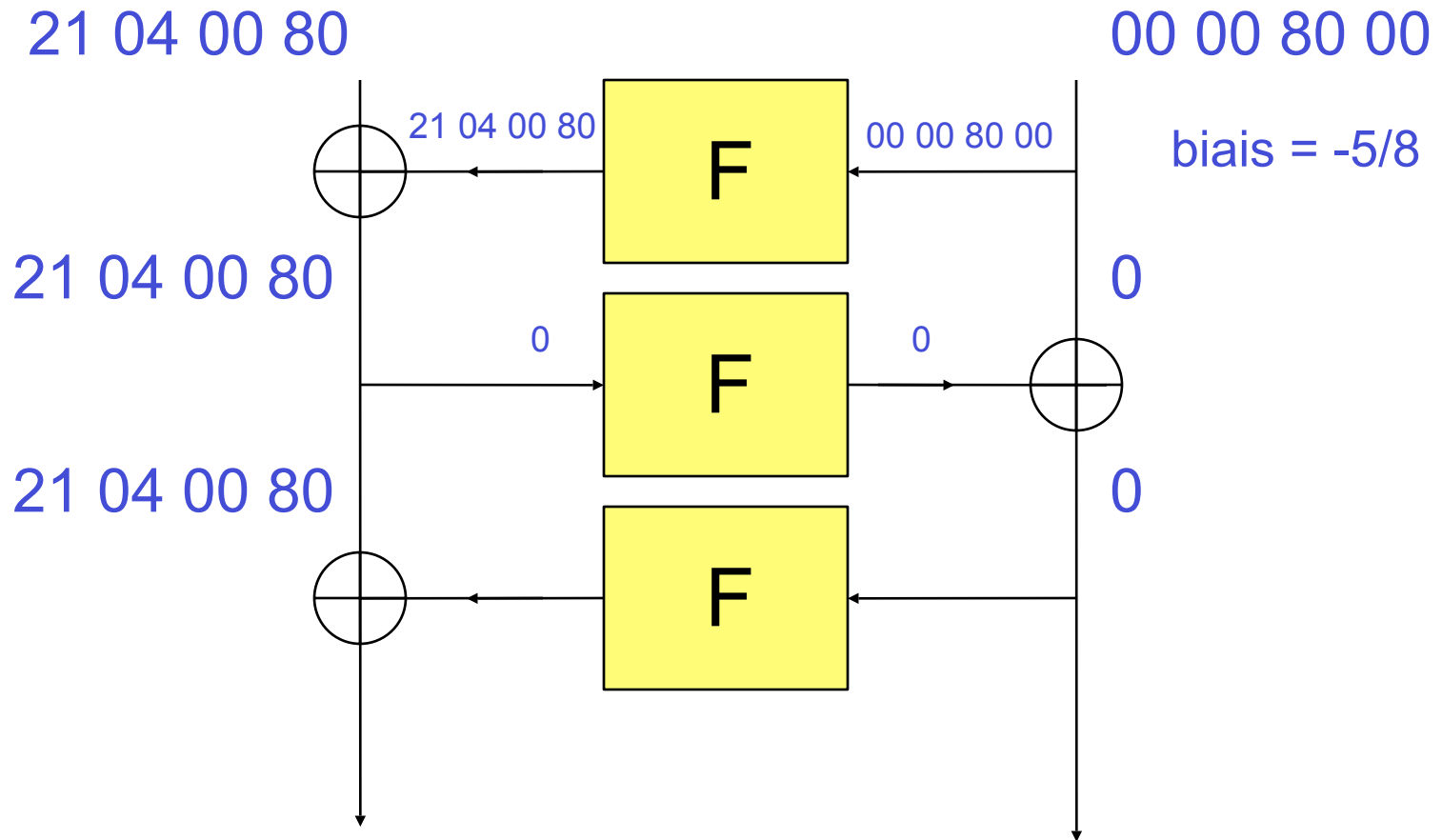
# 3 tours



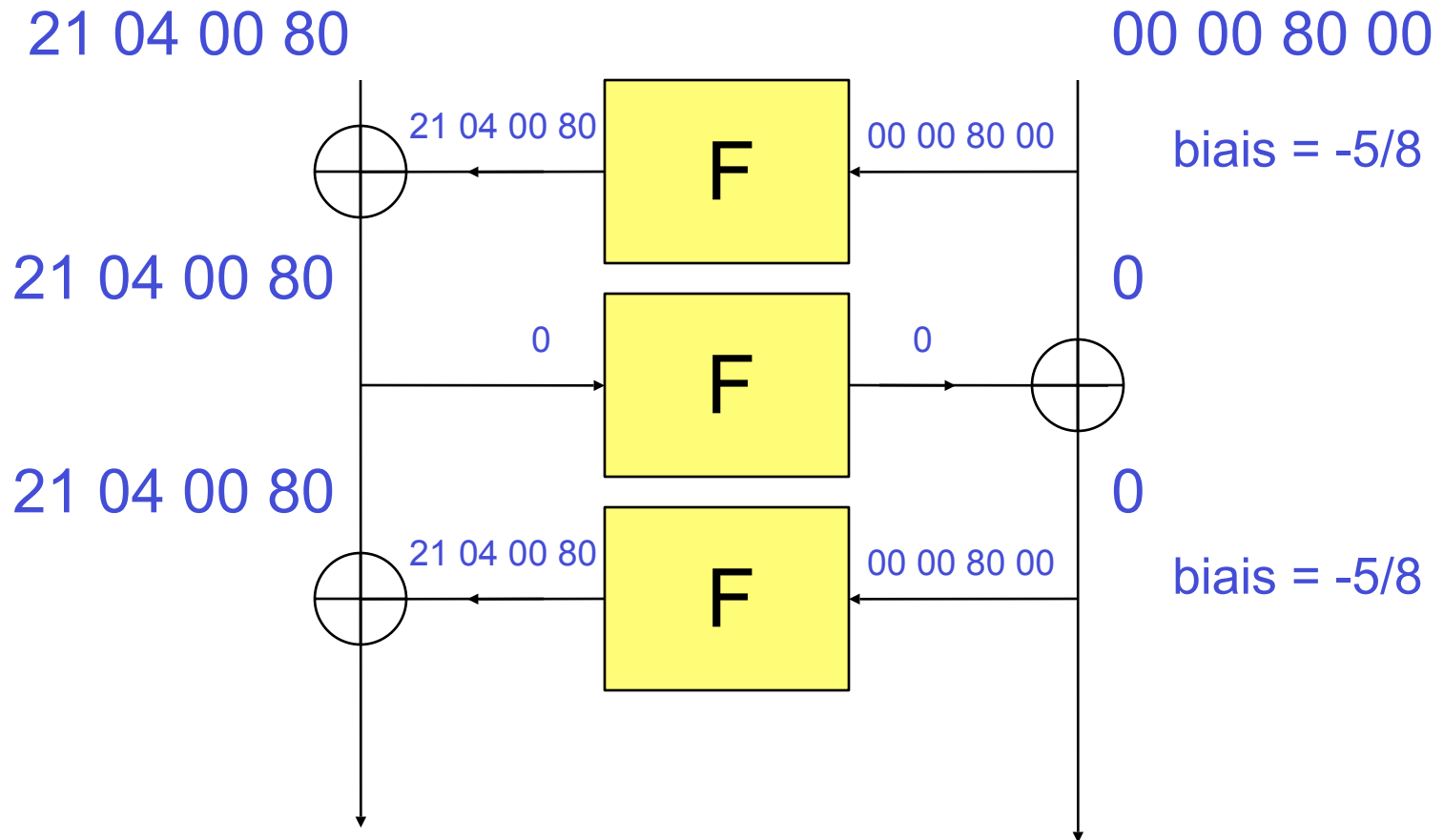
# 3 tours



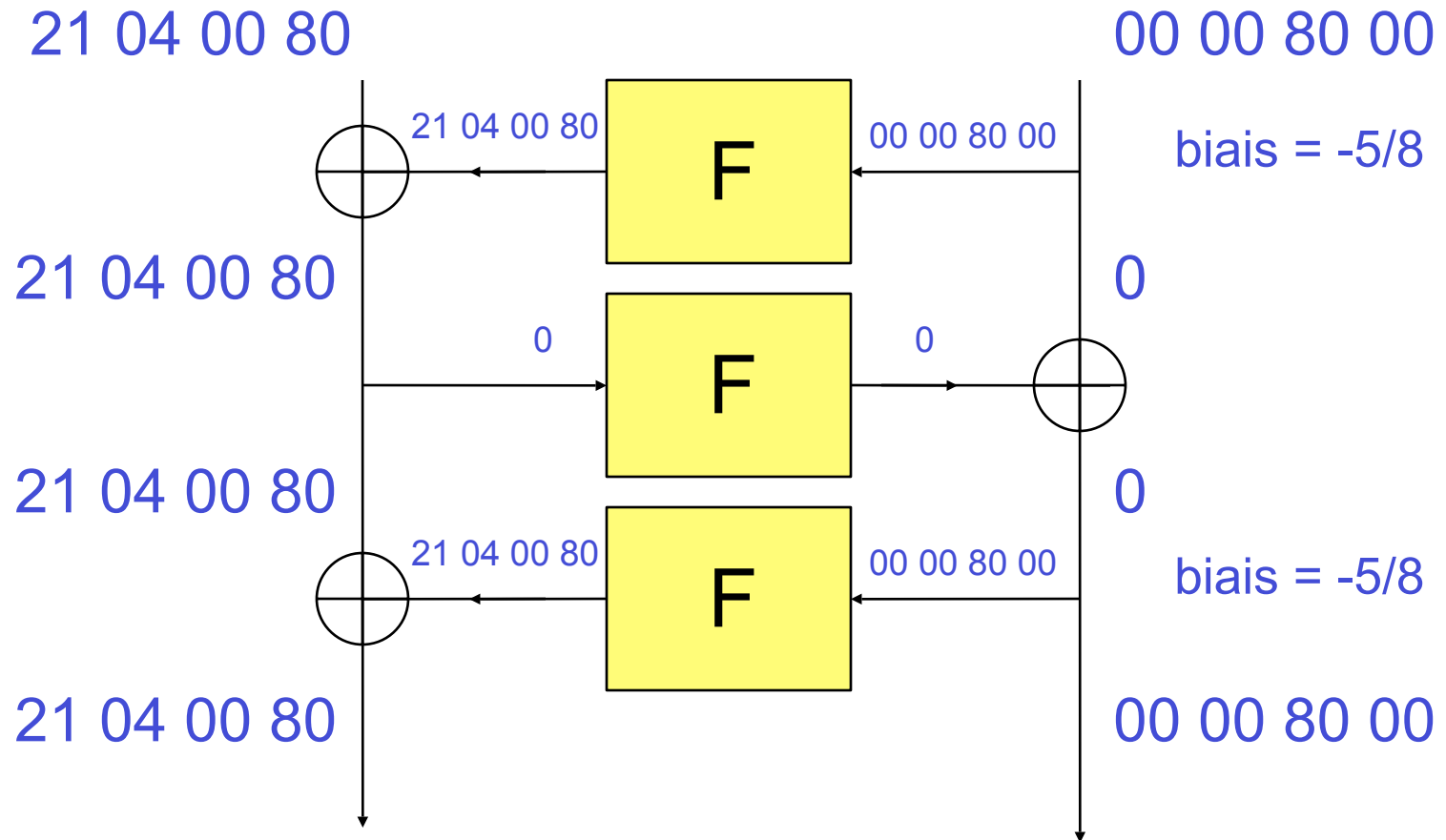
# 3 tours



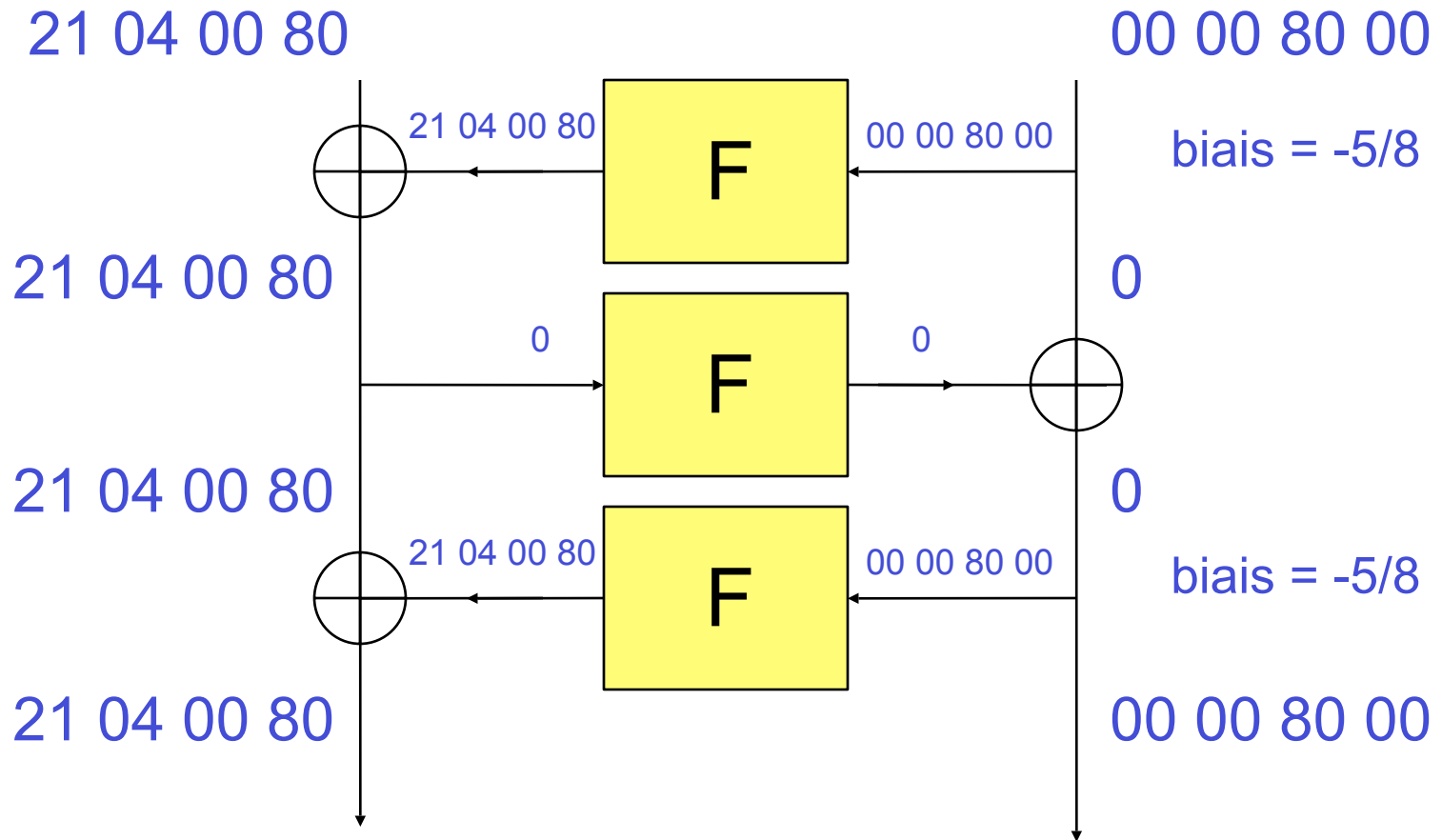
# 3 tours



# 3 tours



# 3 tours



$(21040080, 00008000) \rightarrow (21040080, 00008000)$  [biais = ??? ]

# Calcul du biais total

- Cryptanalyse différentielle
  - Les probabilités se multiplient
  - $P_{\text{total}} = p_1 * \dots * p_n$
- Cryptanalyse linéaire
  - Comment calculer le biais total ?
  - Si  $a_1 \rightarrow a_2$  [biais =  $\varepsilon$ ] et  $a_2 \rightarrow a_3$  [biais =  $\varepsilon'$ ]
  - Alors  $a_1 \rightarrow a_3$  [biais =  $\varepsilon \cdot \varepsilon'$ ]

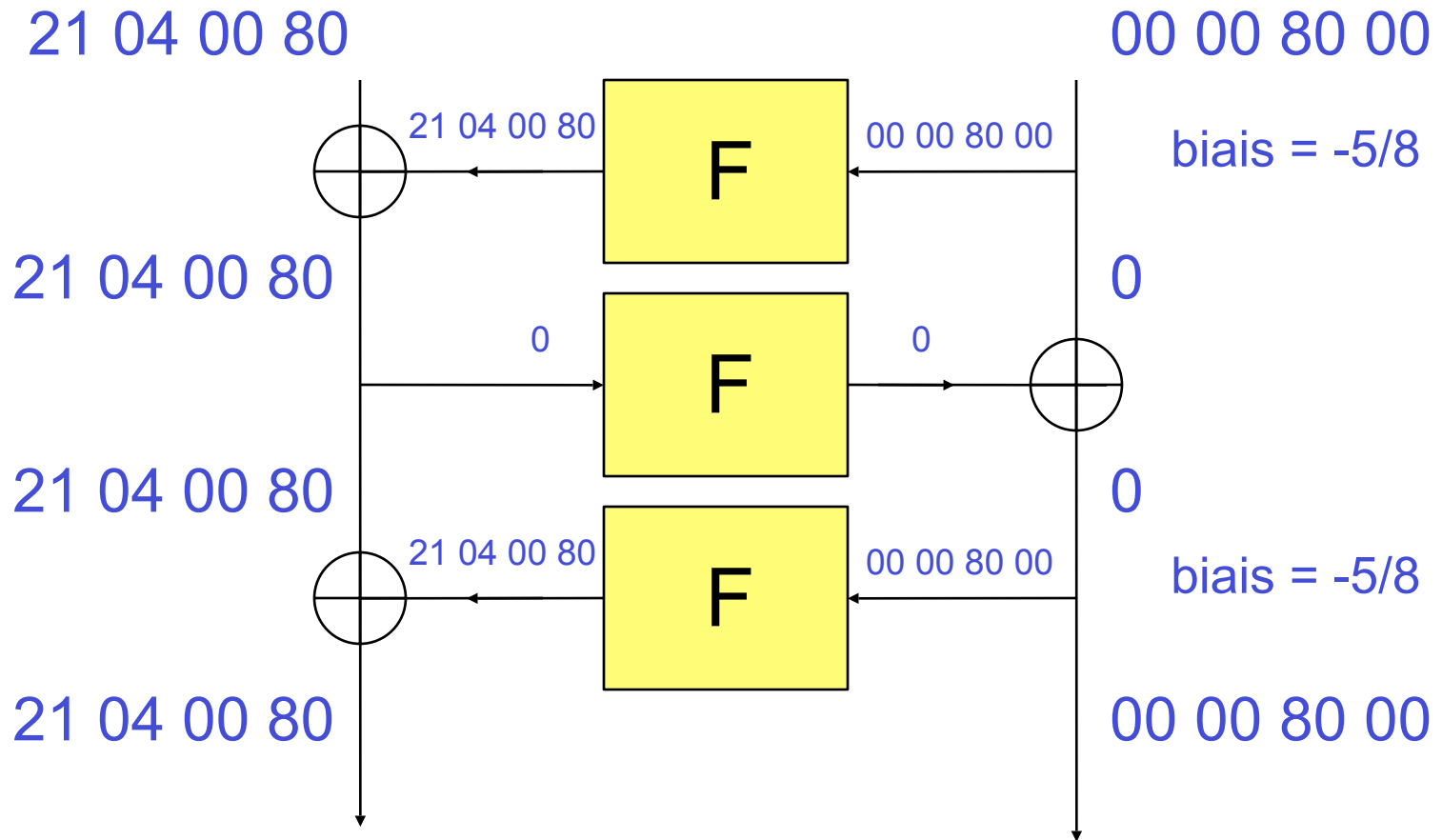
# Calcul du biais total

$$\begin{aligned} P[a_1 = a_3] &= P[a_1 = a_2] \cdot P[a_2 = a_3] \\ &\quad + P[a_1 \neq a_2] \cdot P[a_2 \neq a_3] \\ &= \frac{1}{2} \cdot (1+\varepsilon) \cdot \frac{1}{2} \cdot (1+\varepsilon') \\ &\quad + \frac{1}{2} \cdot (1-\varepsilon) \cdot \frac{1}{2} \cdot (1-\varepsilon') \\ &= \frac{1}{2} \cdot (1 + \varepsilon \cdot \varepsilon') \end{aligned}$$

Donc les biais se multiplient !



# 3 tours

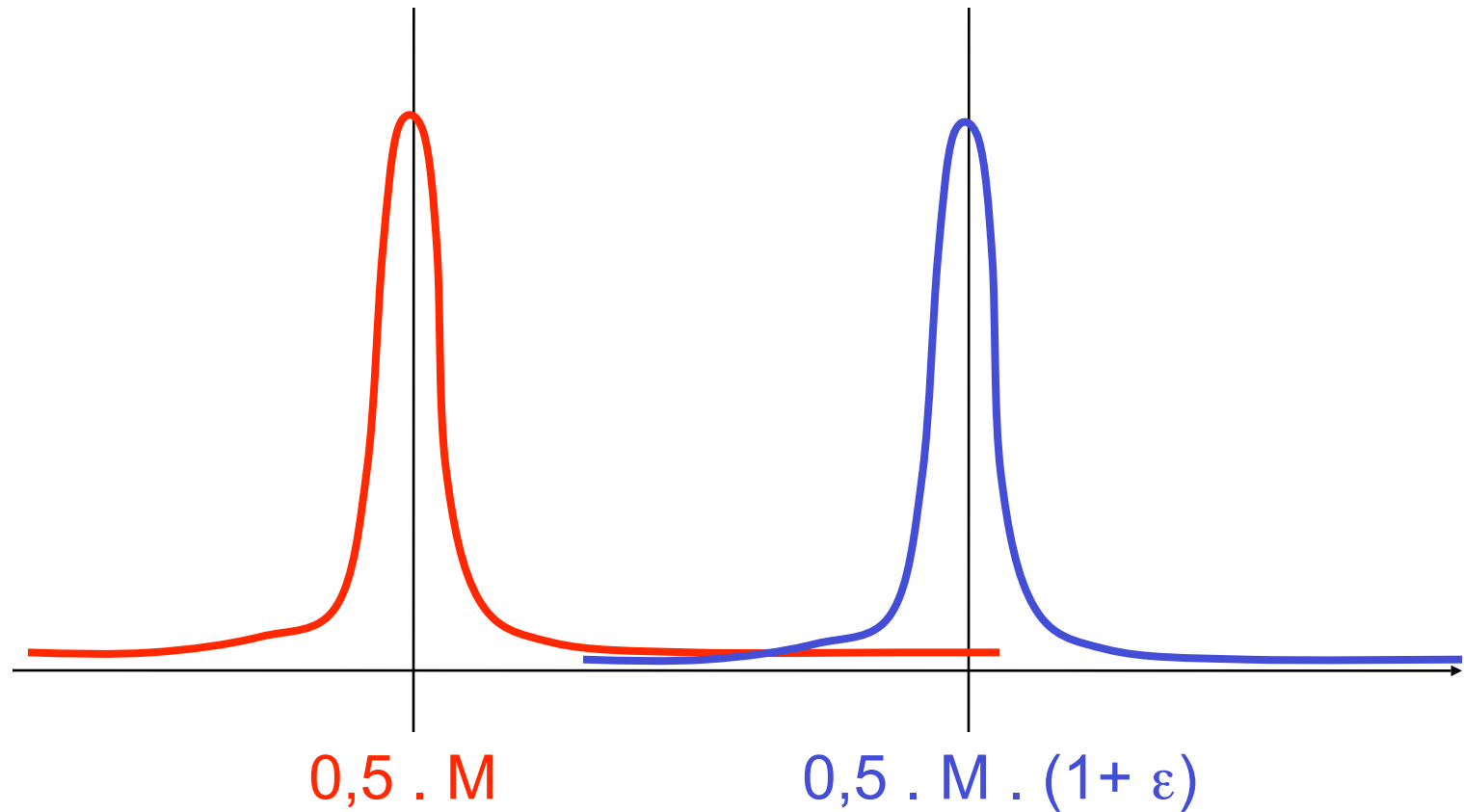


$(21040080, 00008000) \rightarrow (21040080, 00008000)$  [biais = 25/64]

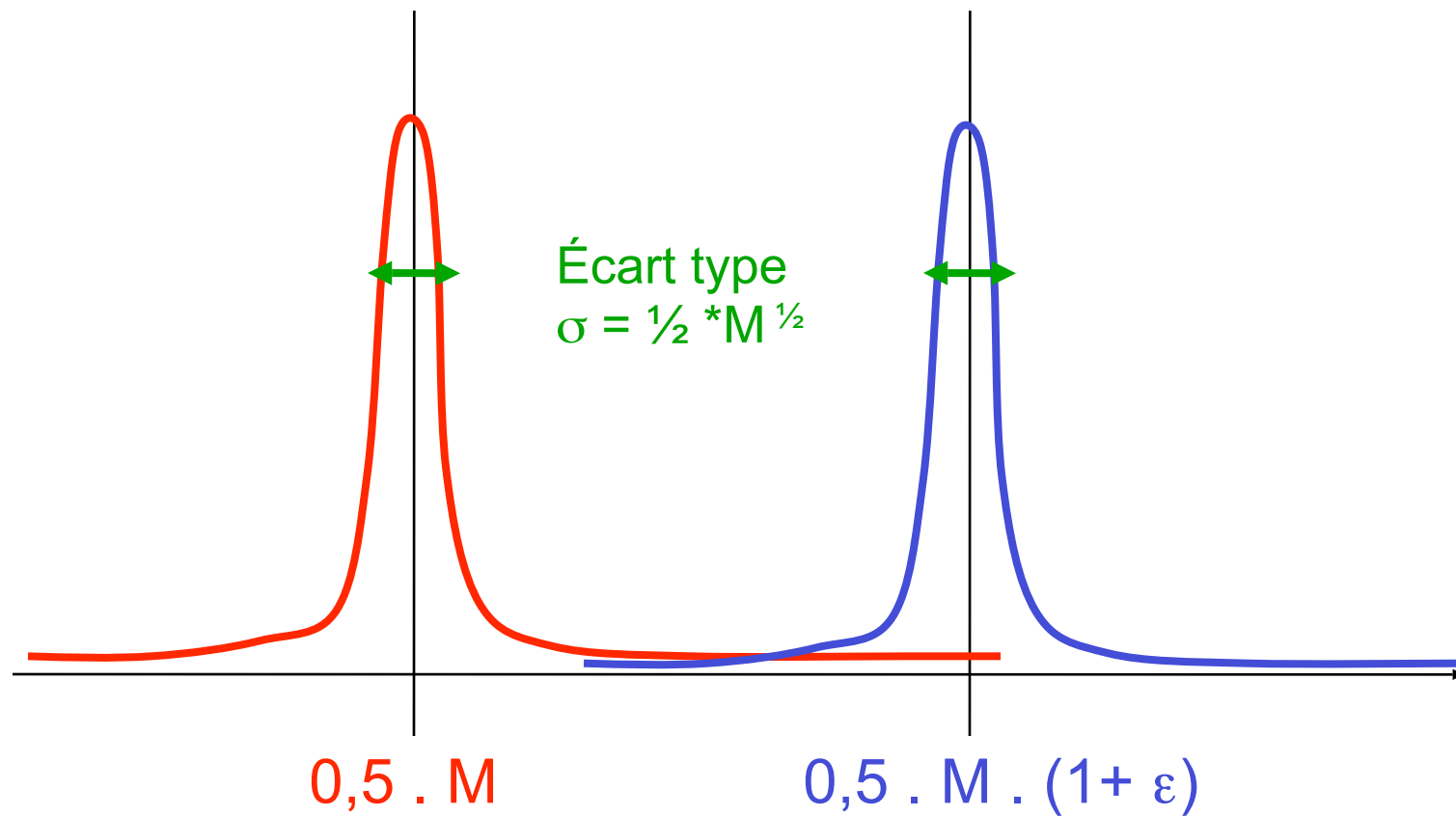
# Résultat contre le DES

- Matsui 1991
  - Caractéristique linéaire contre le DES (16 tours)
  - $a_1 \rightarrow a_2$  [biais =  $2^{-24}$ ]
- Attaque en distingueur
  - Chiffrer  $M = 2^{48}$  clairs
  - Évaluer les formes linéaires  $a_1$  (entrée) et  $a_2$  (sortie)
  - Observer le biais prédit

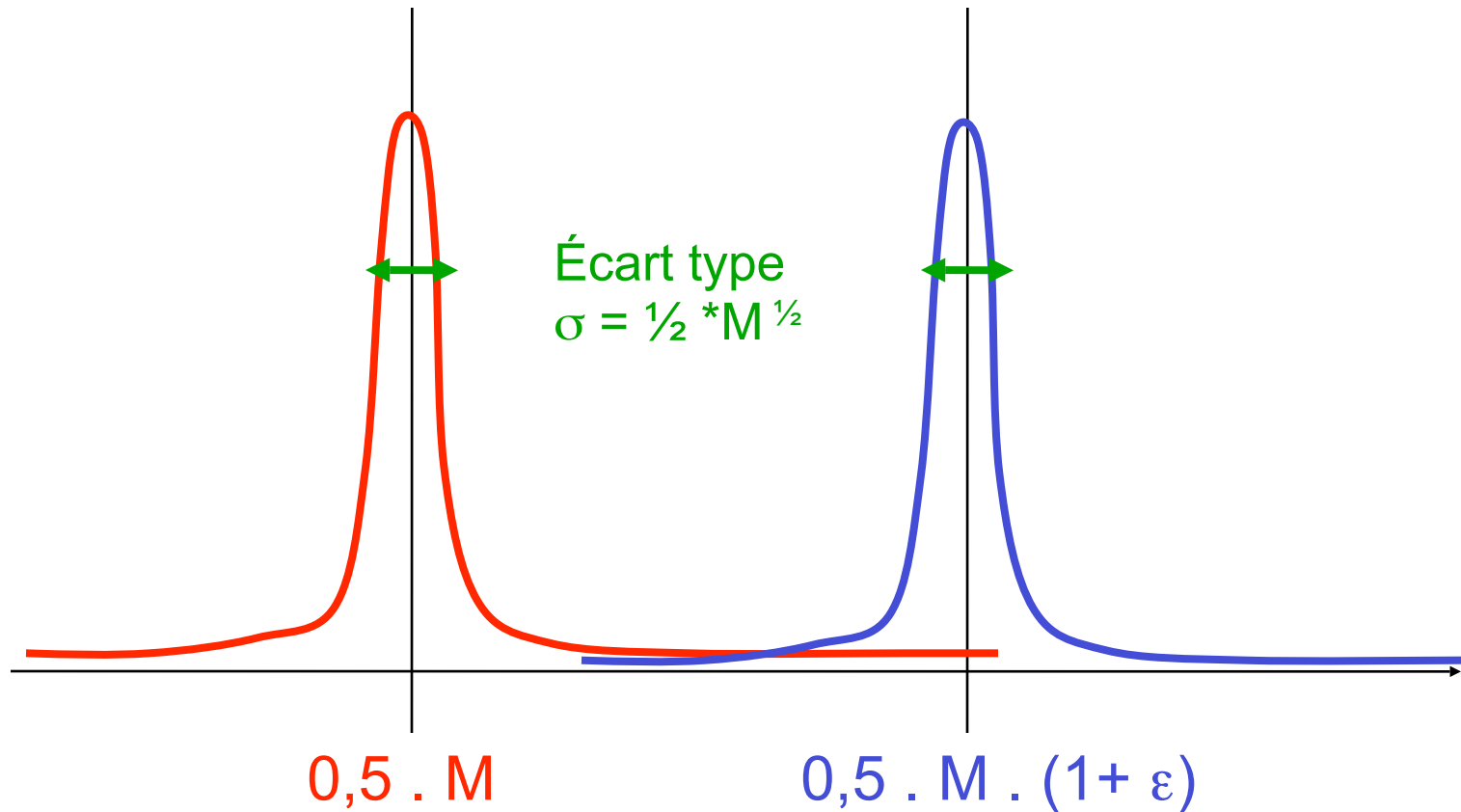
# M nécessaire



# M nécessaire



# M nécessaire



Condition pour distinguer :  $M \cdot \varepsilon = \sigma \rightarrow M = \varepsilon^{-2}$

# Synthèse DES

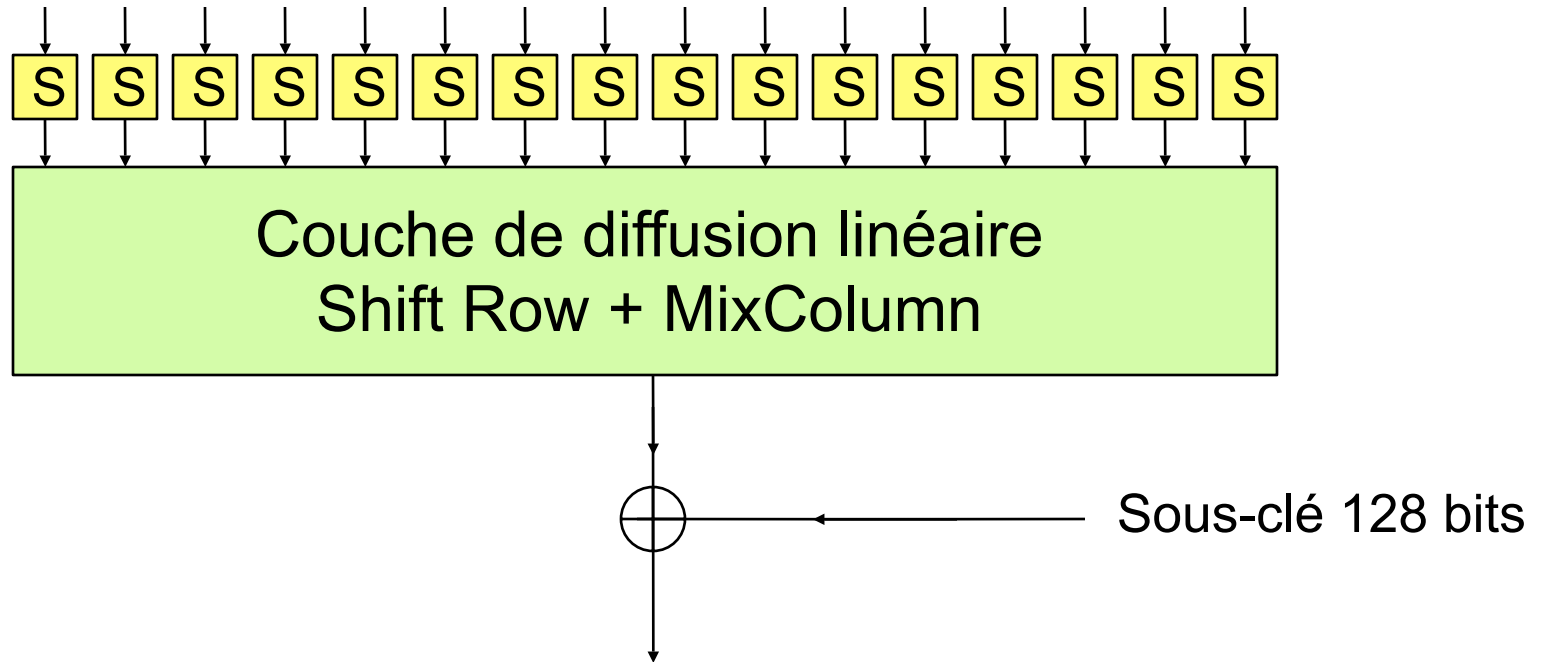
<i>Attaque</i>	<i>Auteur</i>	<i>Temps</i>	<i>Données</i>	<i>Type</i>
Exhaustive	-	$2^{56}$	1	connu
Linéaire	Matsui-91	$2^{43}$	$2^{43}$	connu
Différentielle	BiSha92	$2^{47}$	$2^{47}$	choisi
Davies-Murphy	DaMur95	$2^{45}$	$2^{45}$	choisi

Ces attaques ne marchent pas pour 3DES (nombre de tours !)

# Synthèse

- **Attaques statistiques**
  - Approximation du comportement du block cipher par une **fonction simple**
  - Cryptanalyse différentielle
  - Cryptanalyse linéaire
- Développées contre le DES
- Appliquées à de nombreux algorithmes (FEAL, LUCIFER, etc ...)
- Aujourd'hui, **critère de conception** pris en compte

# Critère de conception AES



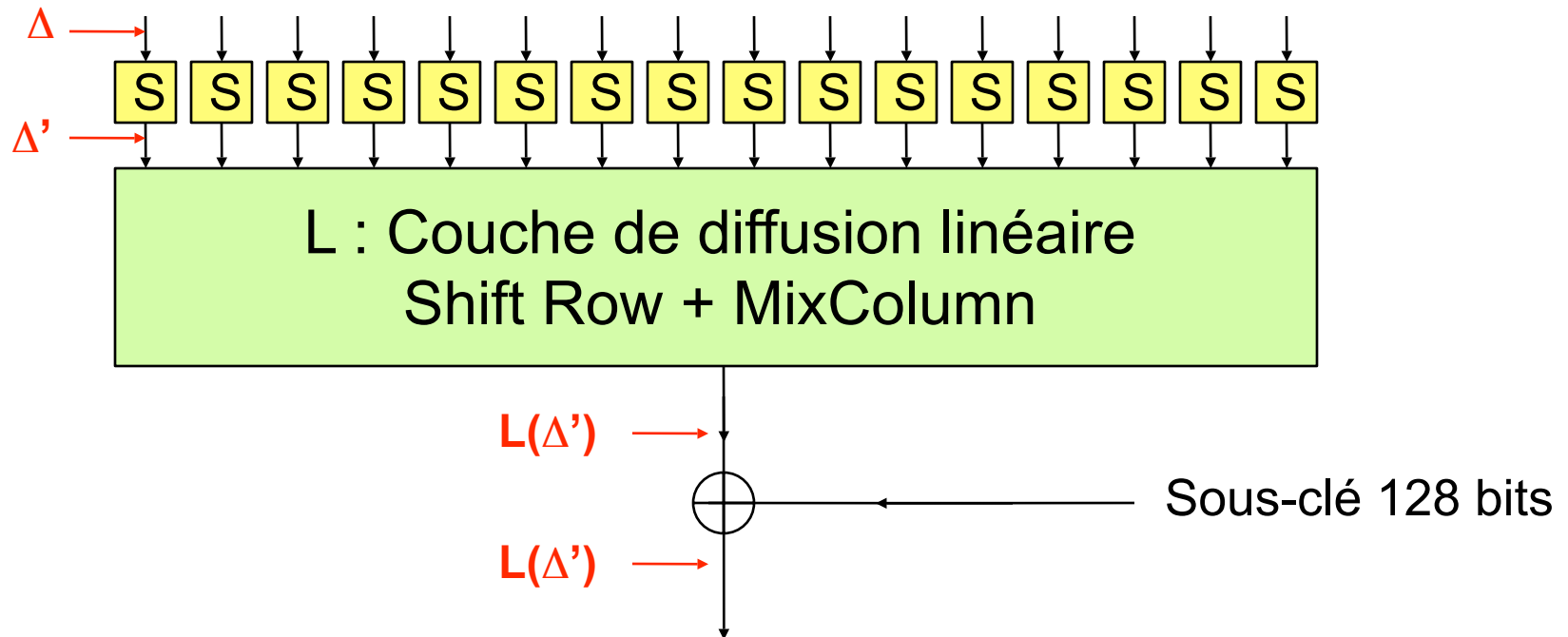
S est une fonction sur 8 bits : il existe forcément de « bonnes » caractéristiques différentielles et linéaires



# S-box de l'AES

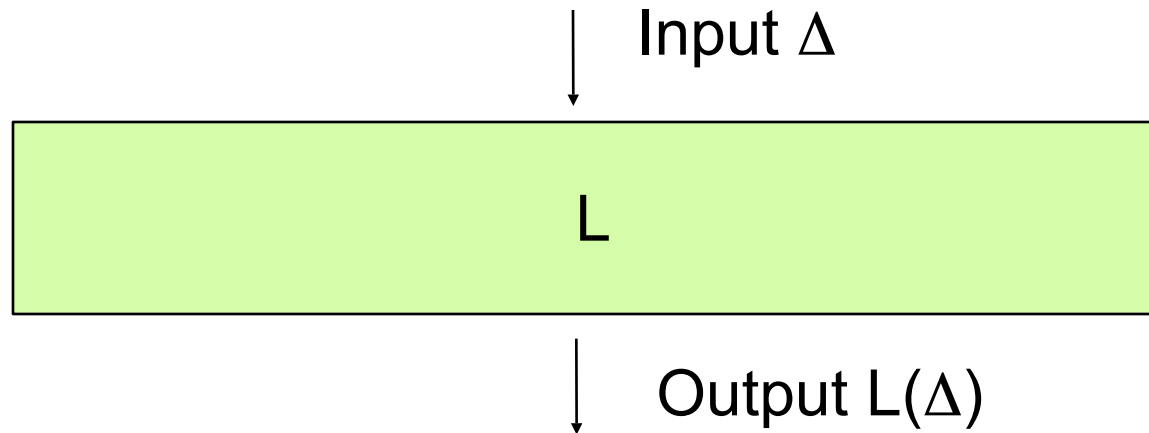
- Pour résister au mieux, S est basée sur l'inversion dans  $GF(256)$ 
  - Meilleure différentielle  $\Delta \rightarrow \Delta'$  [proba =  $2^{-6}$ ]
  - Meilleure linéaire  $a_1 \rightarrow a_2$  [biais =  $2^{-4}$ ]
- On ne peut pas faire mieux pour une fonction  $\{0,1\}^8 \rightarrow \{0,1\}^8$

# Différentielle/linéaire



Le rôle de L est de faire en sorte que  $L(\Delta')$  soit le moins «creux» possible pour tout  $\Delta'$

# Branch number



Pour un vecteur  $\Delta$  de 128 bits, son **poids**  
 **$w(\Delta)$  = Nombre d'octets  $\neq 0$  de  $\Delta$**

$$\text{Branch number}(\Delta) = w(\Delta) + w(L(\Delta))$$

# Cas de l'AES

- Shift Row + MixColumn :
  - Si  $w(\Delta) = 1$
  - Alors  $w(L(\Delta)) = 4$
- Plus généralement, on peut montrer que
$$\forall \Delta, w(\Delta) + w(L(\Delta)) \geq 5$$
- Donc on est obligé d'approximer au moins 2,5 S-box en moyenne par tour !

# Résistance de AES

- Cryptanalyse différentielle

par tour,  $p \geq (2^{-6})^{2,5} = 2^{-15}$

10 tours :  $p_{\text{total}} \geq 2^{-150}$

- Cryptanalyse linéaire

par tour,  $\varepsilon \geq (2^{-4})^{2,5} = 2^{-10}$

10 tours :  $\varepsilon_{\text{total}} \geq 2^{-100}$

# Résistance de l'AES

- L'utilisation de « bonnes » couches de diffusion permet de se protéger contre les attaques statistiques
- Le rôle des S-box est de minimiser les probabilités « élémentaires » d'approximation

# Cryptanalyse de AES

- Attaques prises en compte par les auteurs
  - Cryptanalyse différentielle
  - Cryptanalyse linéaire
- Autres pistes d'attaque
  - Attaque « SQUARE »
  - Attaque algébrique

# Autres attaques statistiques

- Cryptanalyse de Davies-Murphy (DES)
- Extensions de la différentielle
  - Attaque boomerang
  - Différentielle impossible
  - Différentielle « tronquée »
  - Meet(Miss) in the middle
- Extensions de la linéaire (bi-linéaire)



# Résultats académiques

- Souvent, lorsqu'on ne parvient pas à attaquer un block cipher, on **réduit son nombre de tours**
- **AES** avec 6 tours (au lieu de 10) est cassé
- **RC6** avec 17 tours (au lieu de 20) est cassé
- **SKIPJACK** avec 31 tours (au lieu de 32) est cassé
- **IDEA** avec 5 tours (au lieu de 8) est cassé

# Résultats académiques

- Toute attaque en distingueur ayant une complexité  $< 2^k$  est considérée comme « éliminatoire »
- Malgré ces « facilités », peu d'algorithmes « sérieux » sont attaqués
- Critères de sélections
  - Pas de clé faible/corrélée
  - Efficacité en software/hardware
  - Taille du bloc ! (3DES  $\rightarrow$  64 bits)

# Conclusions

Beaucoup de bons block ciphers

- Les 5 finalistes AES (RC6, Mars, Rijndael, Serpent, Twofish)
- candidats NESSIE (CAMELLIA, MISTY, ...)
- KASUMI (GSM)
- SEED (standard coréen)
- IDEA
- RC5
- Triple-DES

# Conclusions

- La plupart des attaques utilisent des approximations statistiques
  - Beaucoup étudiées
  - On sait s'en protéger !
- Nouvelles pistes d'attaques à l'étude :
  - Attaques algébriques
  - Attaques bi-linéaires