Chapitre 3 - Algorithmes de chiffrement par bloc

Alexandra Bruasse-Bac

Plan

- Introduction et généralités
- Les différents modes de chiffrement
- DES, AES
- Autres algorithmes :
  FEAL, IDEA, SAFER, RC5 ...

Introduction et généralités

Les bases

Chiffrement par blocs à n-bits

$$E : \Sigma^n \times \mathcal{K} \rightarrow \Sigma^n$$

Fonction de codage inversible

Les bases

Chiffrement par blocs à n-bits

$$E : \Sigma^n \times \mathcal{K} \rightarrow \Sigma^n$$

Fonction de codage inversible

$$C = E_K(P)$$

$$P = D_K(C)$$

Les bases

Chiffrement par blocs à n-bits

$$E : \Sigma^n \times \mathcal{K} \rightarrow \Sigma^n$$

Fonction de codage inversible

$$C = E_K(P)$$

$$P = D_K(C)$$

Chiffrement par substitution avec un alphabet de grande taille

Critères d'efficacité

Les critères d'efficacité à prendre en compte sont :

- Niveau de sécurité estimé : résistance à la cryptanalyse.

Critères d'efficacité

Les critères d'efficacité à prendre en compte sont :

- Niveau de sécurité estimé : résistance à la cryptanalyse.
- Longueur de la clé : sécurité vs. coût de génération, transmission, stockage

Critères d'efficacité

Les critères d'efficacité à prendre en compte sont :

- Niveau de sécurité estimé : résistance à la cryptanalyse.
- Longueur de la clé : sécurité vs. coût de génération, transmission, stockage
- Débit

Critères d'efficacité

Les critères d'efficacité à prendre en compte sont :

- Niveau de sécurité estimé : résistance à la cryptanalyse.
- Longueur de la clé : sécurité vs. coût de génération, transmission, stockage
- Débit
- Taille des blocs : sécurité vs. complexité (coût d'implémentation)

Critères d'efficacité

Les critères d'efficacité à prendre en compte sont :

- Niveau de sécurité estimé : résistance à la cryptanalyse.
- Longueur de la clé : sécurité vs. coût de génération, transmission, stockage
- Débit
- Taille des blocs : sécurité vs. complexité (coût d'implémentation)
- Complexité de la fonction de cryptage : sécurité vs. coût (développement et hardware)

Critères d'efficacité

Les critères d'efficacité à prendre en compte sont :

- Niveau de sécurité estimé : résistance à la cryptanalyse.
- Longueur de la clé : sécurité vs. coût de génération, transmission, stockage
- Débit
- Taille des blocs : sécurité vs. complexité (coût d'implémentation)
- Complexité de la fonction de cryptage : sécurité vs. coût (développement et hardware)
- Propagation des erreurs

Les différents modes de chiffrement

Pourquoi différents modes

Chiffrement par blocs :

bloc de n bits cryptés en blocs de n bits

Pourquoi différents modes

Chiffrement par blocs :

bloc de n bits cryptés en blocs de n bits

Comment crypter un message de longueur quelconque ?

Pourquoi différents modes

Chiffrement par blocs :

bloc de n bits cryptés en blocs de n bits

Comment crypter un message de longueur quelconque ?

$\Rightarrow$ Le couper en blocs de n bits

Pourquoi différents modes

Chiffrement par blocs :

bloc de n bits cryptés en blocs de n bits

Comment crypter un message de longueur quelconque ?

$\Rightarrow$ Le couper en blocs de n bits

Problèmes

Pas la seule solution

Pourquoi différents modes

Chiffrement par blocs :

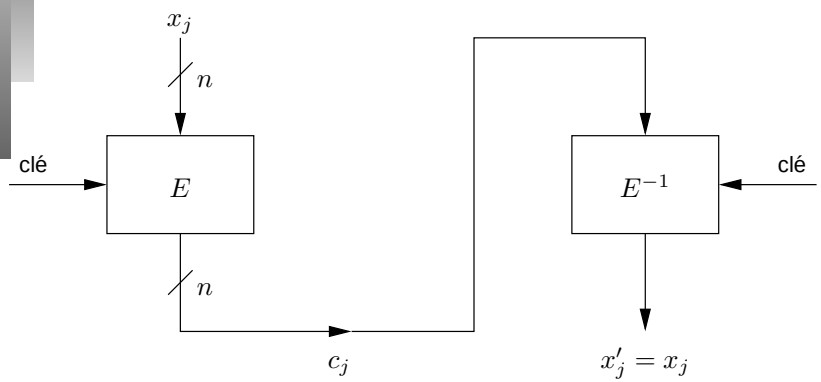
bloc de n bits cryptés en blocs de n bits

Comment crypter un message de longueur quelconque ?

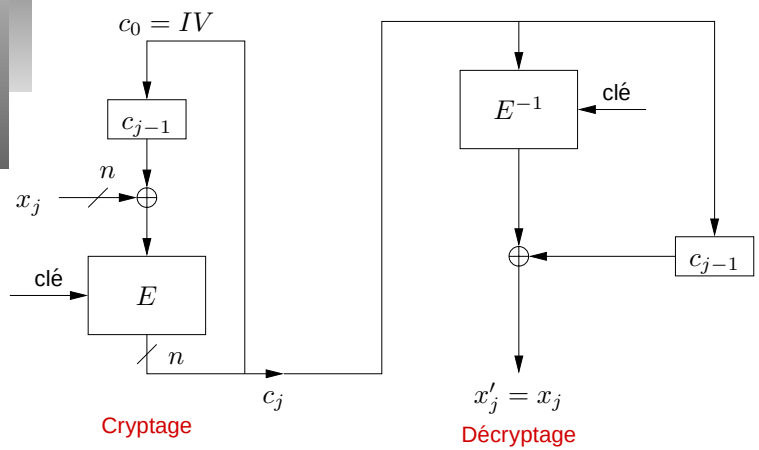
Modes de Chiffrement :

- ECB (Electronic CodeBook)
- CBC (Cipher-Block Chaining)
- CFB (Cipher FeedBack)
- OFB (Output FeedBack)

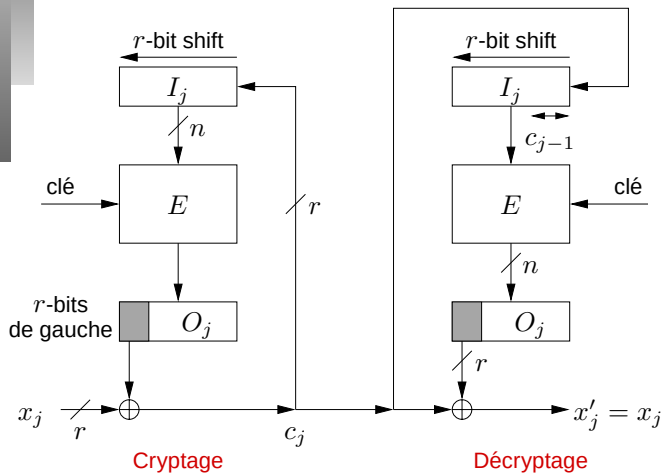
Mode ECB



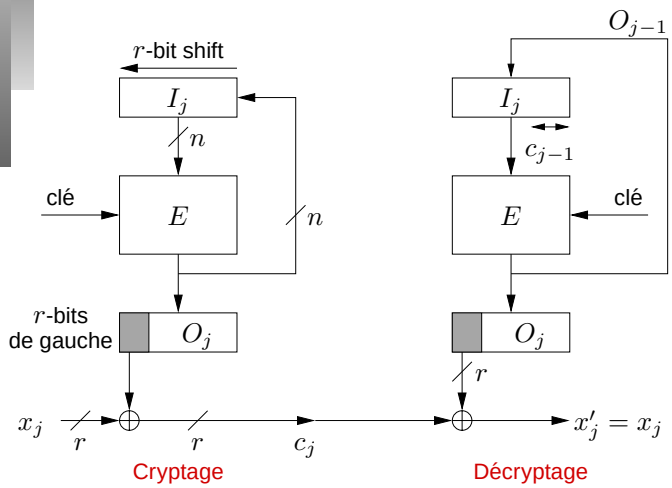
Mode CBC



Mode CFB



Mode OFB



DES

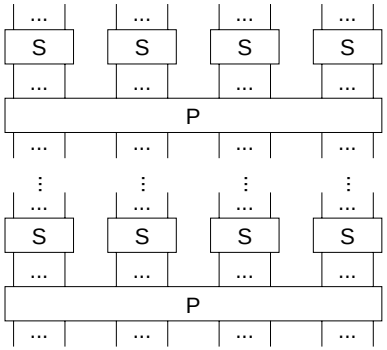
Principes de base

Systèmes cryptographiques produit

combine plusieurs transformations de sorte que la fonction de cryptage résultante soit plus sûre que ses composantes

Systèmes cryptographiques produit

Réseau SP



Systèmes cryptographiques produit

Réseau SP

Chiffrement de Feistel

$$(L_0, R_0) \longrightarrow (L_k, R_k)$$

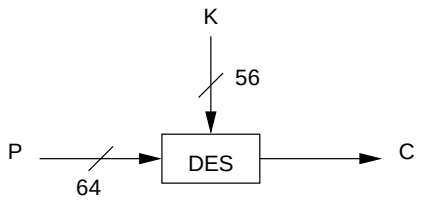
avec

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \end{cases}$$

K_i dérivée de K

Chiffrement Feistel

- Groupement du texte en blocs de 64 bits
- Clé 56 bits (notée K)



Chiffrement Feistel

- Groupement du texte en blocs de 64 bits
- Clé 56 bits (notée K)
- Proposé en 1975
- Adopté en 1977

Principe du DES

Chiffrement Feistel

- Groupement du texte en blocs de 64 bits
- Clé 56 bits (notée K)
- Proposé en 1975
- Adopté en 1977
- Doutes sur la taille de la clé (milieu 90')

Principe du DES

Chiffrement Feistel

- Groupement du texte en blocs de 64 bits
- Clé 56 bits (notée K)
- Proposé en 1975
- Adopté en 1977
- Doutes sur la taille de la clé (milieu 90')
- 1998 : message décrypté en 56 heures

Principe du DES

Chiffrement Feistel

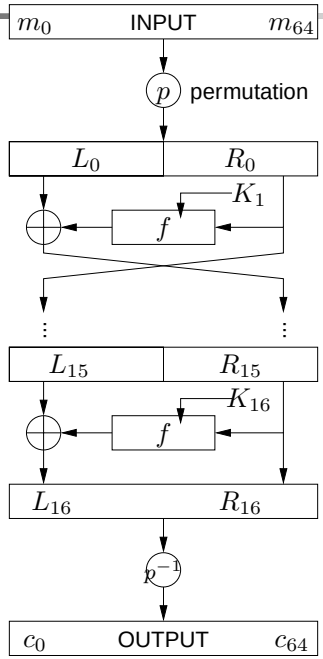
- Groupement du texte en blocs de 64 bits
- Clé 56 bits (notée K)
- Proposé en 1975
- Adopté en 1977
- Doutes sur la taille de la clé (milieu 90')
- 1998 : message décrypté en 56 heures
- 1999 : le temps tombe à 22 heures

Principe du DES

Chiffrement Feistel

- Groupement du texte en blocs de 64 bits
- Clé 56 bits (notée K)
- Proposé en 1975
- Adopté en 1977
- Doutes sur la taille de la clé (milieu 90')
- 1998 : message décrypté en 56 heures
- 1999 : le temps tombe à 22 heures
- AES : clé de 128, 192 ou 256 bits

Principe du DES



Principe du DES

Clé intermédiaire K_i

Transformation de K :

- on coupe K en deux fois 28 bits
- on shift de 1 ou 2 bits (un aux rounds 1, 2, 9 et 16)
- on recolle

On choisit 48 bits particuliers : K_i

Principe du DES

Chiffrement de Feistel : fonction f

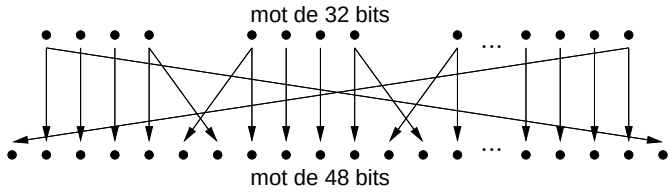
$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$$

Principe du DES

Chiffrement de Feistel : fonction f

$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$$

Fonction d'expansion E



Principe du DES

Chiffrement de Feistel : fonction f

$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$$

Boite S (complexité du DES)

8 boites S : 6 bits $\rightarrow$ 4 bits

mot $b_1b_2b_3b_4b_5b_6$:

- b_1b_6 : instruction
- $b_2b_3b_4b_5$: donnée

Principe du DES

Chiffrement de Feistel : fonction f

$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$$

Boite S (complexité du DES)

La boite S_5 :

$$\begin{pmatrix} 2 & 12 & 4 & 1 & 7 & 10 & 11 & 6 & 8 & 5 & 3 & 15 & 13 & 0 & 14 & 9 \\ 14 & 11 & 2 & 12 & 4 & 7 & 13 & 1 & 5 & 0 & 15 & 10 & 3 & 9 & 8 & 6 \\ 4 & 2 & 1 & 11 & 10 & 13 & 7 & 8 & 15 & 9 & 12 & 5 & 6 & 3 & 0 & 14 \\ 11 & 8 & 12 & 7 & 1 & 14 & 2 & 13 & 6 & 15 & 0 & 9 & 10 & 4 & 5 & 3 \end{pmatrix}$$

Principe du DES

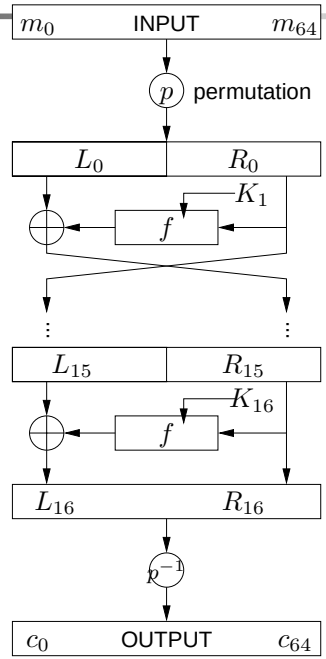
Chiffrement de Feistel : fonction f

$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$$

Permutation P

couplage des sorties des S-boites

Décryptage



Propriétés, sécurité

Propriétés de base :

- Chaque bit crypté dépend de tous les bits de la clé et du texte

Propriétés, sécurité

Propriétés de base :

- Chaque bit crypté dépend de tous les bits de la clé et du texte

- Pas de relation statistique entre le texte et son cryptage

Propriétés, sécurité

Propriétés de base :

- Chaque bit crypté dépend de tous les bits de la clé et du texte
- Pas de relation statistique entre le texte et son cryptage

- Changer 1 bit du texte modifie tous les bits cryptés avec proba. 1/2

Propriétés, sécurité

Propriété de complémentation

Soit $\bar{x}$ le complément bit à bit de x.

Si $y = E_K(x)$ alors $\bar{y} = E_{\bar{K}}(\bar{x})$

$\Rightarrow$ Réduit l'espace de recherche.

Propriétés, sécurité

Propriété de complémentation

Clés faibles, semi-faibles

Une *clé faible* est telle que :

$$E_K(E_K(x)) = x$$

Une paire de *clés semi-faibles* est telle que :

$$E_{K_1}(E_{K_2}(x)) = x$$

Propriétés, sécurité

Propriété de complémentation

Clés faibles, semi-faibles

DES a 4 clés faibles et 6 paires de clés semi-faibles

Propriétés, sécurité

Propriété de complémentation

Clés faibles, semi-faibles

DES n'est pas un groupe

$\Rightarrow$ Important pour le cryptage multiple

Cryptage multiple

Cryptage double

$$E(x) = E_{K_1}(E_{K_2}(x))$$

Cryptage triple

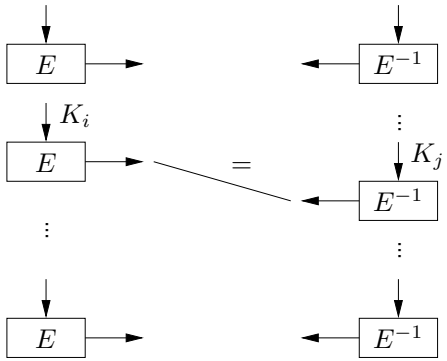
$$E(x) = \tilde{E}_{K_1}(\tilde{E}_{K_2}(\tilde{E}_{K_3}x))$$

avec $\tilde{E} = E$ ou E^{-1}

Le cas E-E^{-1}-E avec $K_1 = K_3$ est appelé *cryptage triple à clé double*.

Cryptage multiple

Attaque par le milieu (cryptage double)



Cryptage multiple

Cryptage triple :

Triple DES - 3DES

$\Rightarrow$ résoud le problème de la taille de la clé

$\Rightarrow$ lent

AES

Historique

Suite aux doutes sur DES :

- Janvier 1997 : projet AES annoncé

AES - Advanced Encryption Standard

Historique

Suite aux doutes sur DES :

- Janvier 1997 : projet AES annoncé

  AES - Advanced Encryption Standard

- Septembre 1997 : le “public” est invité à proposé des cryptosystèmes appropriés

Historique

Suite aux doutes sur DES :

- Janvier 1997 : projet AES annoncé

  AES - Advanced Encryption Standard

- Septembre 1997 : le “public” est invité à proposé des cryptosystèmes appropriés

- 15 candidats sérieux

  dont des variantes d’algorithmes populaires : RC5, SAFER-SK, CAST

Historique

Suite aux doutes sur DES :

- Janvier 1997 : projet AES annoncé

  AES - Advanced Encryption Standard

- Septembre 1997 : le “public” est invité à proposé des cryptosystèmes appropriés
- 15 candidats sérieux

  dont des variantes d’algorithmes populaires : RC5, SAFER-SK, CAST

- 5 finalistes :

  MARS, RC6, Rijndael (Joan Daemen, Vincent Rijmen - Belgique), Serpent, Twofish

Historique

Suite aux doutes sur DES :

- Janvier 1997 : projet AES annoncé

  AES - Advanced Encryption Standard

- Septembre 1997 : le “public” est invité à proposé des cryptosystèmes appropriés
- 15 candidats sérieux

  dont des variantes d’algorithmes populaires : RC5, SAFER-SK, CAST

- 5 finalistes :

  MARS, RC6, Rijndael (Joan Daemen, Vincent Rijmen - Belgique), Serpent, Twofish

  $\Rightarrow$ Octobre 2001

$\mathbb{F}_{2^8}$

Eléments :

- polynômes de degré au plus 7
- calcul modulo

$$m(X) = X^8 + X^4 + X^3 + X + 1$$

$\mathbb{F}_{2^8}$

Eléments :

- polynômes de degré au plus 7
- calcul modulo

$$m(X) = X^8 + X^4 + X^3 + X + 1$$

Représentation

suite de bits $b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0$:

$$b_7X^7 + b_6X^6 + b_5X^5 + b_4X^4 + b_3X^3 + b_2X^2 + b_1X + b_0$$

$\mathbb{F}_{2^8}$

Eléments :

- polynômes de degré au plus 7
- calcul modulo

$$m(X) = X^8 + X^4 + X^3 + X + 1$$

Exemple

La suite `01010111` (`'57'` en hexadécimal) :

$$X^6 + X^4 + X^2 + X + 1$$

Algorithme

$\Rightarrow$ longueur de bloc variable (128, 192 ou 256 bits)

$\Rightarrow$ longueur de clé variable (128, 192 ou 256 bits)

Algorithme

⇒ longueur de bloc variable (128, 192 ou 256 bits)
⇒ longueur de clé variable (128, 192 ou 256 bits)
⇒ Transformations portent sur un état
tableau rectangulaire d'octets de :

- *4 lignes*
- $Nb = (lg\_bloc/32)$ *colonnes*

Algorithme

⇒ longueur de bloc variable (128, 192 ou 256 bits)
⇒ longueur de clé variable (128, 192 ou 256 bits)
⇒ Transformations portent sur un état (N_b)

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,6}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,6}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,6}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,6}$

- Exemple d'état pour $Nb = 6$: 192 bits -

Algorithme

⇒ longueur de bloc variable (128, 192 ou 256 bits)
⇒ longueur de clé variable (128, 192 ou 256 bits)
⇒ Transformations portent sur un état (N_b)
⇒ Même représentation pour la clé (N_k)

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

- Exemple de clé pour $Nk = 4$: 128 bits -

Algorithme

Ordre de remplissage des octets (à partir d'un bloc) :

$m_{0,0}$	$m_{0,1}$		...
$m_{1,0}$	$m_{1,1}$		...
$m_{2,0}$	$m_{2,1}$		...
$m_{3,0}$	$m_{3,1}$		...

Algorithme

Ordre de remplissage des octets (à partir d'un bloc) :

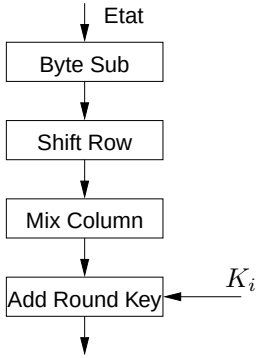
$m_{0,0}$	$m_{0,1}$		...
$m_{1,0}$	$m_{1,1}$		...
$m_{2,0}$	$m_{2,1}$		...
$m_{3,0}$	$m_{3,1}$		...

Nombre de rounds N_r dépend de N_b et N_k :

N_r	$N_b = 4$	$N_b = 6$	$N_b = 8$
$N_k = 4$	10	12	14
$N_k = 6$	12	12	14
$N_k = 8$	14	14	14

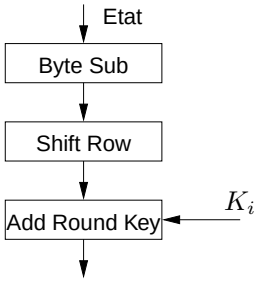
Algorithme

Structure d'un round :



Algorithme

Round final :



Transformation "Byte Sub"

On considère chaque octet de l'état comme un élément de $\mathbb{F}_{2^8}$.

On appelle S-box la suite de transformation :

- prendre l'inverse multiplicatif dans $\mathbb{F}_{2^8}$

Transformation "Byte Sub"

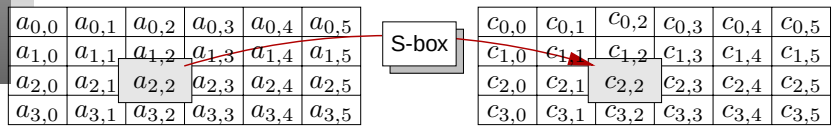
On considère chaque octet de l'état comme un élément de $\mathbb{F}_{2^8}$.

On appelle S-box la suite de transformation :

- prendre l'inverse multiplicatif dans $\mathbb{F}_{2^8}$
- appliquer la transformation affine suivante sur $\mathbb{F}_2$:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Transformation "Byte Sub"



Transformation "Shift Row"

On applique à chaque ligne une permutation cyclique

ligne 0 $\rightarrow$ pas shiftée
ligne 1 $\rightarrow$ vers offset C_1
ligne 2 $\rightarrow$ vers offset C_2
ligne 3 $\rightarrow$ vers offset C_3

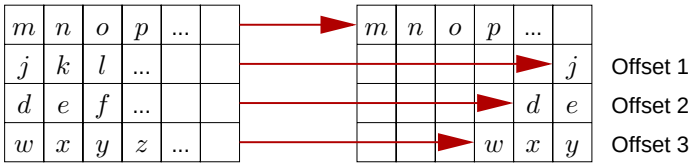
Transformation "Shift Row"

On applique à chaque ligne une permutation cyclique

N_b	C_1	C_2	C_3
4	1	2	3
6	1	2	3
8	1	3	4

Transformation "Shift Row"

On applique à chaque ligne une permutation cyclique



Transformation "Mix Column"

On considère les colonnes de l'état comme des polynômes de $\mathbb{F}_{2^8}[X]$.

Transformation Mix Column

$$a(X) \rightarrow a(X) \cdot c(X) \ [X^4 + 1]$$

avec

$$c(X) = \texttt{'03'}X^3 + \texttt{'02'}X^2 + \texttt{'01'}X + \texttt{'02'}$$

Transformation "Mix Column"

On considère les colonnes de l'état comme des polynômes de $\mathbb{F}_{2^8}[X]$.

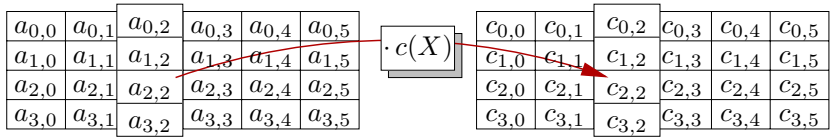
Transformation Mix Column

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Transformation "Mix Column"

On considère les colonnes de l'état comme des polynômes de $\mathbb{F}_{2^8}[X]$.

Transformation Mix Column

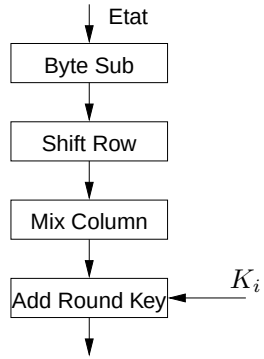


Transformation "Add Round Key"

La clé du round est ajoutée (XOR) à l'état.

$\Rightarrow$ la clé du round est de même taille que l'état

Résumé



Clé de round obtenue par :

- Expansion de la clé
- Sélection d'un certain nombre de bits

Sécurité, remarques

- Pas de symétries
- Pas de clés faibles ou semi-faibles

Sécurité, remarques

- Pas de symétries
- Pas de clés faibles ou semi-faibles
- Preuve de non sensibilité à la cryptanalyse différentielle

Sécurité, remarques

- Pas de symétries
- Pas de clés faibles ou semi-faibles
- Preuve de non sensibilité à la cryptanalyse différentielle
- Preuve de non sensibilité à la cryptanalyse linéaire

Sécurité, remarques

- Pas de symétries
- Pas de clés faibles ou semi-faibles
- Preuve de non sensibilité à la cryptanalyse différentielle
- Preuve de non sensibilité à la cryptanalyse linéaire
- Sensibilité à l'attaque "Square" (mais coûteux)

Sécurité, remarques

- Pas de symétries
- Pas de clés faibles ou semi-faibles
- Preuve de non sensibilité à la cryptanalyse différentielle
- Preuve de non sensibilité à la cryptanalyse linéaire
- Sensibilité à l'attaque "Square" (mais coûteux)

Le décryptage est moins efficace que le cryptage (implémentation soft et hard).

Autres cryptosystèmes

IDEA

IDEA (International Data Encryption Algorithm)

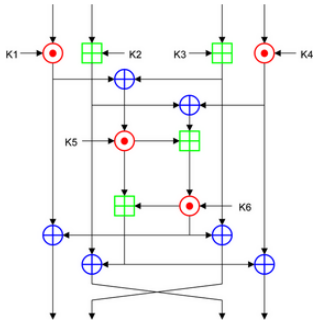
- blocs de 64 bits
- clé de 128 bits
- 8 round et un round de sortie

- utilisé comme algorithme de cryptage symétrique dans les versions initiales de pgp

IDEA

IDEA (International Data Encryption Algorithm)

- blocs de 64 bits
- clé de 128 bits
- 8 round et un round de sortie



- $\oplus$ XOR
- $\odot$ multiplication modulo $2^{16}+1$
- $\boxplus$ addition modulo 2^{16}

IDEA

IDEA (International Data Encryption Algorithm)

- blocs de 64 bits
- clé de 128 bits
- 8 round et un round de sortie

$\Rightarrow$ jugé très sûr jusqu'en 1999

$\Rightarrow$ problèmes dus aux progrès de la cryptanalyse

$\Rightarrow$ algorithmes plus rapides

Blowfish

- blocs de 64 bits
- clé de 32 à 448 bits
- 16 round Feistel

- très similaire à CAST-128
- non propriétaire (utilisé dans ssh)