Modes de chiffrement symétrique

1

Modes = méthodes pour utiliser les chiffrements par blocs : les modes opératoires

Dans le cadre d'une implémentation pratique, l'algorithme 'pur' est combiné à une série d'opérations simples en vue d'améliorer la sécurité sans pour autant pénaliser l'efficacité de l'algorithme.

Cette combinaison est appelée un **mode cryptographique.**

Critères :

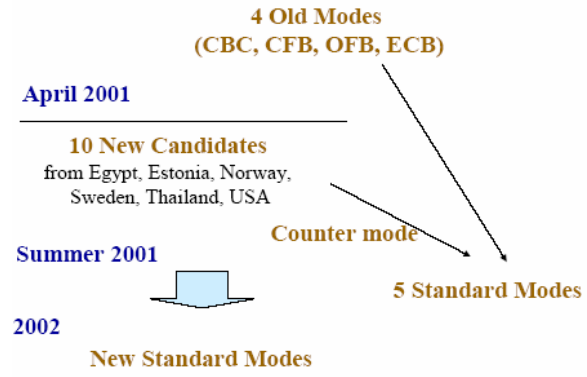
Sécurité:

- Effacement des formats standards (ex. l'introduction d'un texte).
- Protection contre la modification de C.
- Chiffrement de plusieurs messages avec la même clé.

Efficacité:

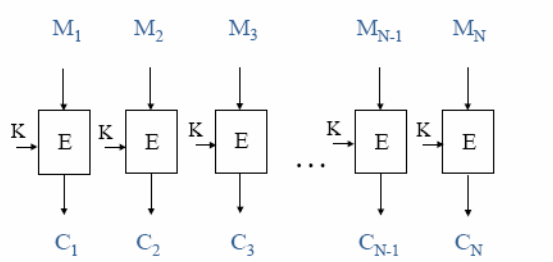
- L'utilisation d'un mode cryptographique ne doit pas pénaliser l'efficacité du cryptosystème.
- Limitation de la propagation des erreurs qui apparaissent dans M ou C.

Nouveaux modes



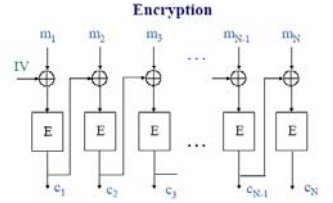
Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key.	•Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.	•General-purpose block-oriented transmission •Authentication
Cipher Feedback (CFB)	Input is processed J bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	•General-purpose stream-oriented transmission •Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding DES output.	•Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	•General-purpose block-oriented transmission •Useful for high-speed requirements

ECB – Electronic Code Book

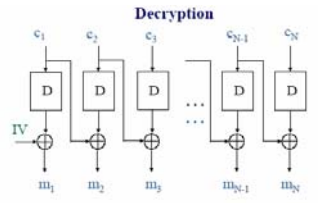


$C_i = E_K(M_i)$ for i=1..N

CBC – Cipher block chaining

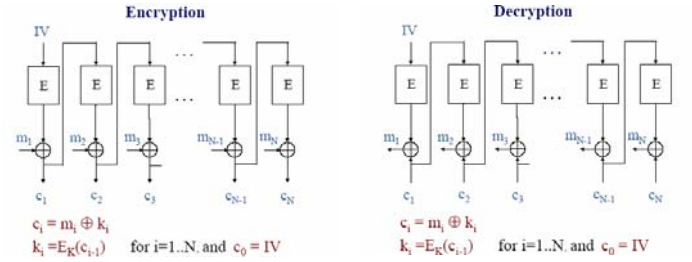


$c_i = E_K(m_i \oplus c_{i-1})$ for i=1..N c_0=IV



$m_i = D_K(c_i) \oplus c_{i-1}$ for i=1..N c_0=IV

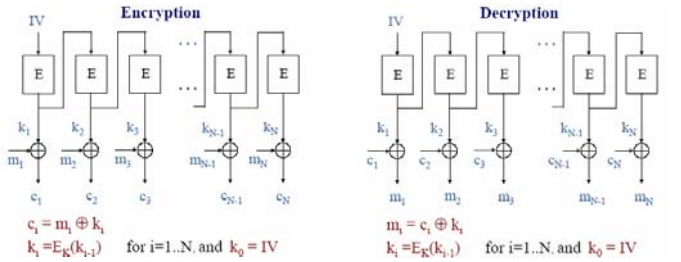
CFB – Cipher FeedBack



$c_i = m_i \oplus k_i$
$k_i = E_K(c_{i-1})$ for i=1..N, and $c_0 = IV$

$c_i = m_i \oplus k_i$
$k_i = E_K(c_{i-1})$ for i=1..N, and $c_0 = IV$

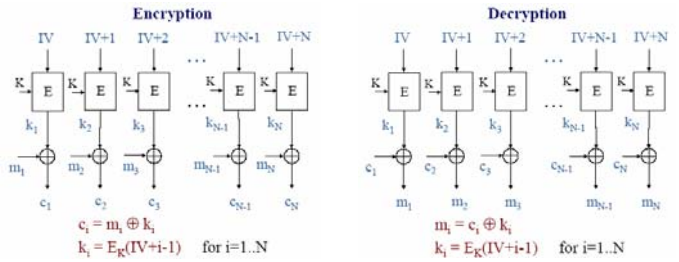
OFB – Output FeedBack



$c_i = m_i \oplus k_i$
$k_i = E_K(k_{i-1})$ for i=1..N, and $k_0 = IV$

$m_i = c_i \oplus k_i$
$k_i = E_K(k_{i-1})$ for i=1..N, and $k_0 = IV$

CTR



Comparaison des modes

	ECB	CBC	CFB	OFB	CTR
Sécurité	Faible	Forte	Forte	Forte	Forte
Vitesse de base	S_{ECB}	S_{ECB}	$j/L*S_{ECB}$	$j/L* S_{ECB}$	S_{ECB}
Paralléllisme - pipelining	E / D	E	E	Aucun	E / D
Opération de chiffrement	E / D	E	E	E	E / D
Préprocessing	Non	Non	Non	Oui	Oui
Accès aléatoire	R/W	R only	R only	non	R/W

E : chiffrement

D : déchiffrement

	ECB	CBC	CFB	OFB	CTR
Sécurité contre une attaque de type « exhaustive key search »					
Nbr min de bloc de C ou M nécessaires	1*M 1*C	1*M 2*C	1*M 2*C (pour j=L)	2*M 2*C (pour j=L)	2*M 2*C
Propagation d'erreur dans le message déchiffré					
Modification de j bits	L bits	L + j bits	L + j bits	j bits	j bits
Suppression de j bits	Courant + suite	Courant + suite	L bits	Courant + suite	Courant + suite
Intégrité	Non	Non	Non	Non	Oui

M : texte clair

C : texte chiffré

Algorithme de chiffrement conventionnels

Algorithme	Clé	Bloc	Ronde	Applications
DES	56	64	16	SET, Kerberos
3DES	168	64	48	Gestion clé financière, PGP, S/MIME
AES	128/192/256	128	10/12/14	Remplace DES
IDEA	128	64	8	PGP
Blowfish	→ 448	64	16	Divers
RC5	→ 448	64	→ 255	divers

"International Data Encryption Algorithm" (IDEA)

- proposé comme remplacement des DES
- chiffre symétrique par bloc
- blocs de 64 bits, clef de 128 bits
- Utilisé dans PGP

Blowfish

- chiffre symétrique par bloc, très rapide et économique en mémoire, mais à sécurité variable selon la taille de la clef

RC5

- Utilisable en solution matérielle ou logicielle
- Rapide et simple
- Nombre de round variable
- Longueur de clé variable
- Nécessite peu de mémoire
- Haute sécurité
- Rotation dépendantes des données

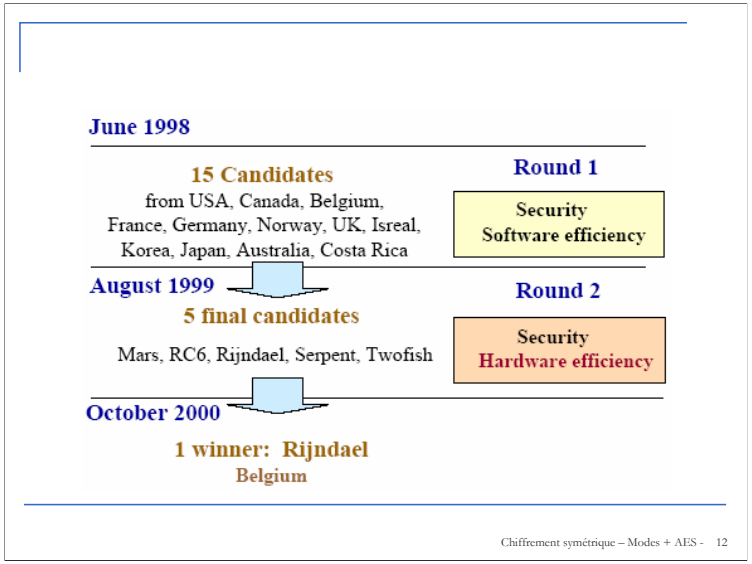
A.E.S

Introduction

11

•Le Triple DES demeure une norme acceptée pour les documents gouvernementaux U.S.A. pour le futur prévisible

•Pas de projet ou d'obligation de réencrypter les documents existants



Critères d'évaluation au 2e tour :

•La sécurité générale

•Le coût en terme de calculs

•Simplicité de l'algorithme et facilité d'implémentation

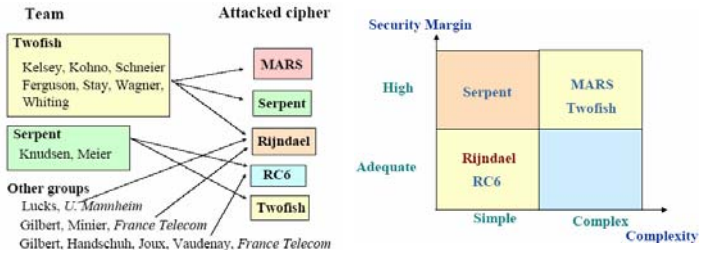
•Résistance aux attaques connues

•flexibilité

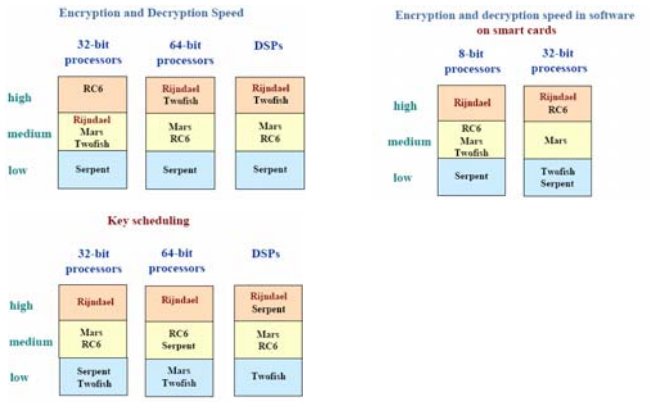
AES – cahier des charges

- Grande sécurité – résistance à toutes les attaques connues
- Large portabilité : carte à puces, processeurs dédiés, …
- Rapidité
- Lecture facile de l'algorithme
- Blocs de 128 bits et clés de 128/192/256 bits
- Durée de vie de 20 à 30 ans

Rapport du NIST : sécurité



Efficacité logicielle



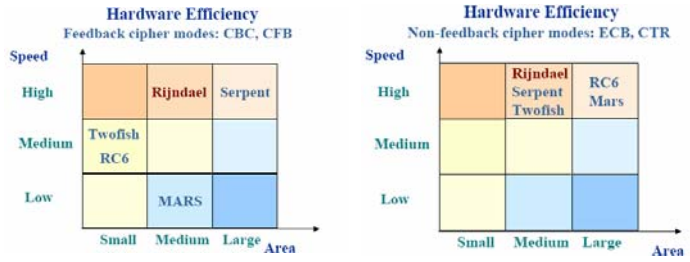
Chiffrement/déchiffrement

- forte variation des résultats
- Serpent est le moins bon pour la majorité des plate-formes

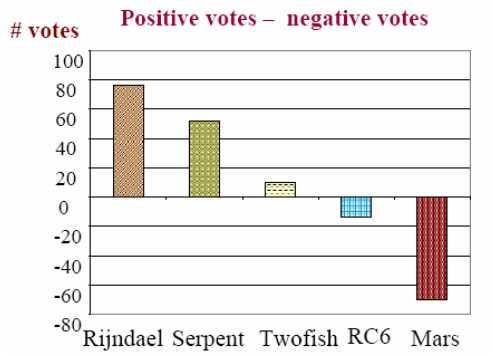
Configuration de la clé

- Variation modérée des résultats
- Rijndael et RC6 sont les meilleurs pour la majorité des plate-formes
- Twofish et Serpent sont les pires

Efficacité matérielle



Votes



Rijndael (Daemen, Rijmen)

- Avantages
  - Le plus rapide en implantation matérielle
  - Proche des plus rapides en implantation logicielle
  - Très grande flexibilité
- Inconvénients
  - Marge de sécurité

Serpent (Anderson, Biham, Knudsen)

- Avantages
  - Large marge de sécurité
  - Construction conservatrice
  - Très rapide en implantation matérielle
  - Réputation cryptanalytique de ses auteurs (Anderson, Biham, Knudsen)
- Inconvénients
  - Lent en implantation logicielle
  - Flexibilité modérée

Twofish (Schneier, Wagner, …)

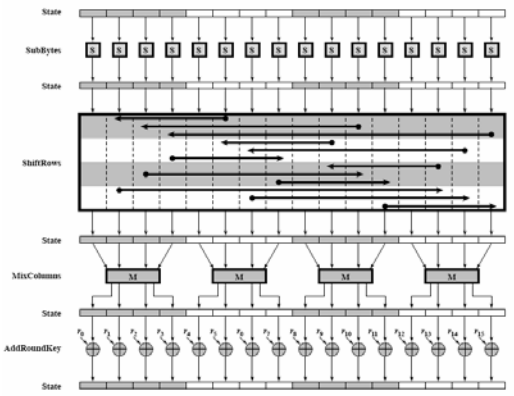
- Avantages
  - Bonne marge de sécurité
  - Rapide pour le chiffrement/déchiffrement en implantation logicielle
  - Américain
  - Beaucoup de publicité
- Inconvénients
  - Modérément rapide en implantation matérielle
  - Configuration lente de la clé en implantation logicielle
  - Flexibilité modérée

Rijndael

Rijndael - principe

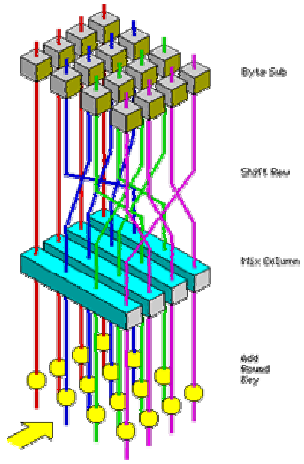
- Plusieurs longueurs de clef et de bloc: 128, 192, ou 256 bits
- chiffre itératif par bloc
- Nombre de cycles (ou "rondes) : de 10 à 14 selon la longueur des blocs et des clés
- Série de transformations/permutations/sélection
- Beaucoup plus performant que DES
- Adaptable à des processeurs de 8 ou de 32 bits
- Susceptible d'être parallélisé
- Pour le décryptage, les transformations sont apportées dans l'ordre inverse

Rijndael

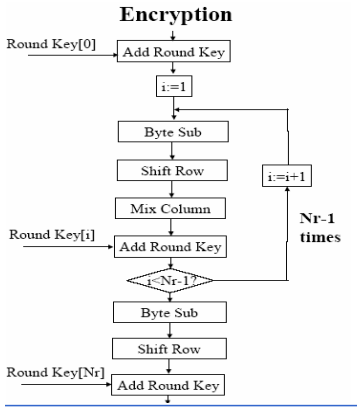


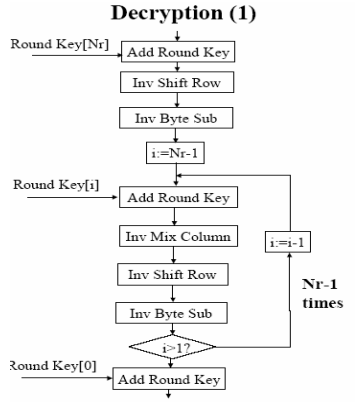
À chaque ronde, quatre transformations sont appliquées:

- substitution d'octets
- décalage de rangées
- déplacement de colonnes (sauf à la dernière ronde)
- addition d'une "clef de ronde" varie à chaque ronde

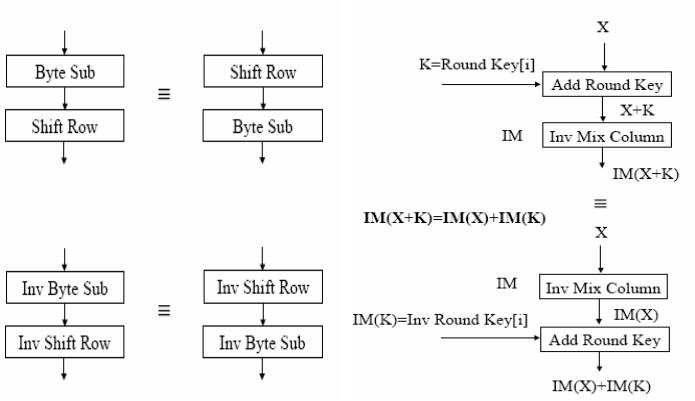


Chiffrement - déchiffrement

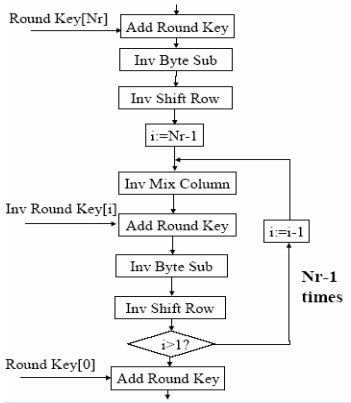




Propriété des transformations



Modification du déchiffrement



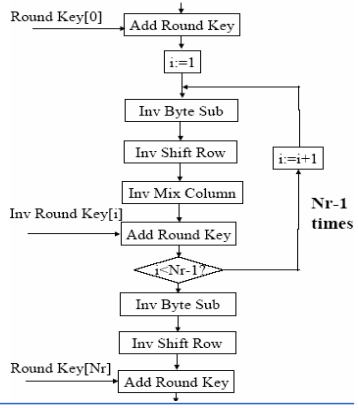
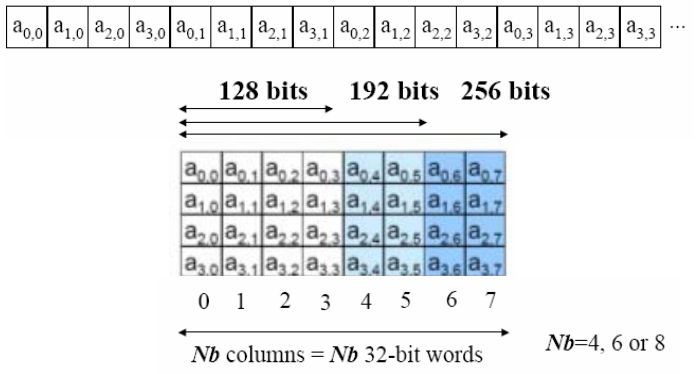


Table d'état du texte



Une table d'état est utilisée

- 4 rangées par N_b colonnes avec $N_b = L_{bloc}/32$

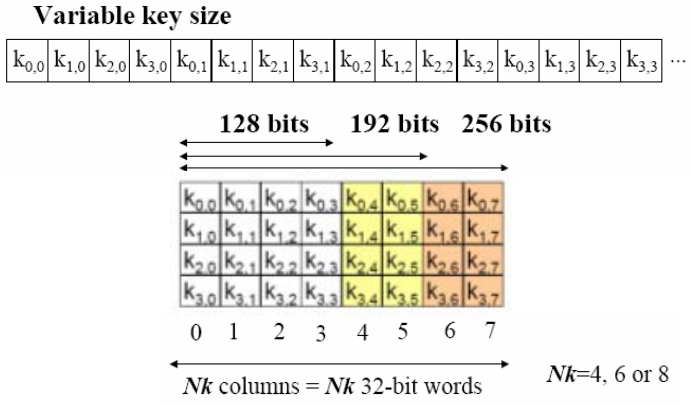
La clef est aussi représentée sous forme de tableau

- 4 rangées par N_k colonnes avec $N_k = L_{clef}/32$

L'input et l'output sont des séquences linéaires d'octets

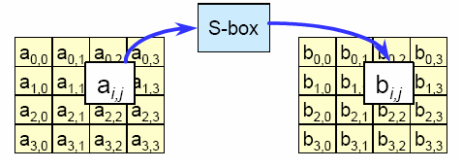
- longueurs possibles: 16, 24, ou 32 octets
- l'appariement de octets des blocs et de la clef se fait le long des colonnes des tableaux précédents
- donc une colonne du tableau correspond à un mot de 32 bits

Table d'état des clés

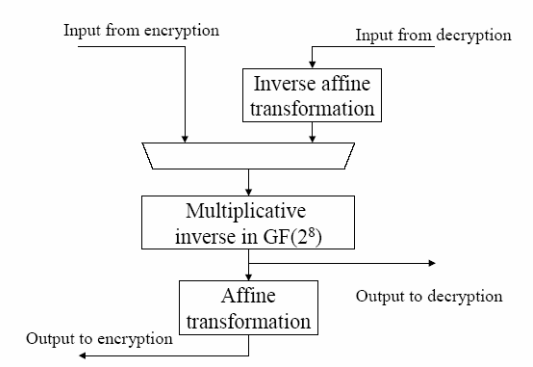


ByteSub

- Les bytes sont transformés en appliquant une S-Box inversible
- Une seule S-Box pour tous le chiffrement
- Augmente la non-linéarité



ByteSub fonctionnement



1 – Calcul d'un inverse multiplicatif

2 – Transformation affine

Transformation affine

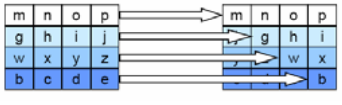
$$\begin{bmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \end{bmatrix} \begin{bmatrix} t_7 \\ t_6 \\ t_5 \\ t_4 \\ t_3 \\ t_2 \\ t_1 \\ t_0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

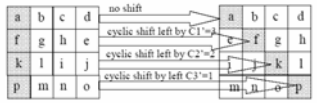
i.e.,

$$b_7 = t_7 + t_3 + t_2 + t_1 + t_0 + 1$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$b_0 = t_4 + t_3 + t_2 + t_1 + t_0 + 0$$

ShiftRow

Transformation inverse





	Block size		
	128 bits	192 bits	256 bits
C1	1	1	1
C2	2	2	2
C3	3	3	4

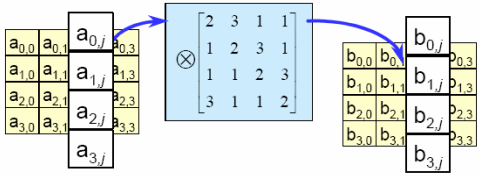
	Block size		
	128 bits	192 bits	256 bits
C1'	3	5	7
C2'	2	4	5
C3'	1	3	4

Décale les bits des sous-blocs

Augmente la diffusion des données dans le round

MixColumn

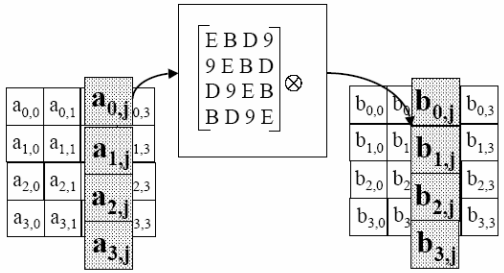
- Haute diffusion
  - Une différence sur 1 byte d'entrée se propage sur les 4 bytes de sortie
  - Une différence sur 2 bytes d'entrée se propage sur au moins 3 bytes de sortie



Multiplication d'une matrice aux sous-blocs de 16 bits.

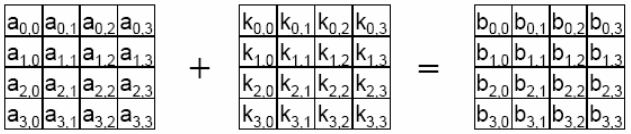
Diffusion des données entre les rounds

InvMixColumn



Add Round Key

- Simple XOR des clés



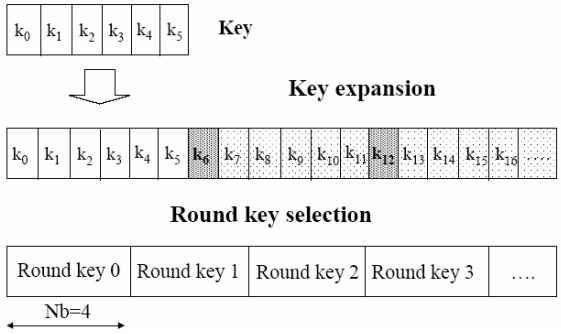
Additions des sous-clés aux sous-blocs correspondant

Nombre de ronde

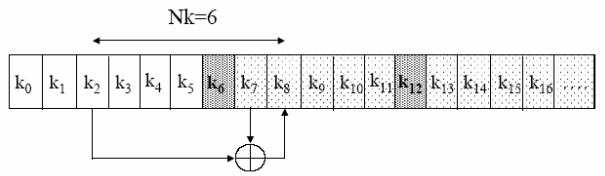
	Key length		
Block length	**128 bits Nk=4**	**192 bits Nk=6**	**256 bits Nk=8**
128 bits Nb=4	**10**	**12**	**14**
192 bits Nb=6	**12**	**12**	**14**
256 bits Nb=6	**14**	**14**	**14**

Calcul de la clé

Key size = 192 bits (Nk=6)
Block size = 128 bits (Nb=4)

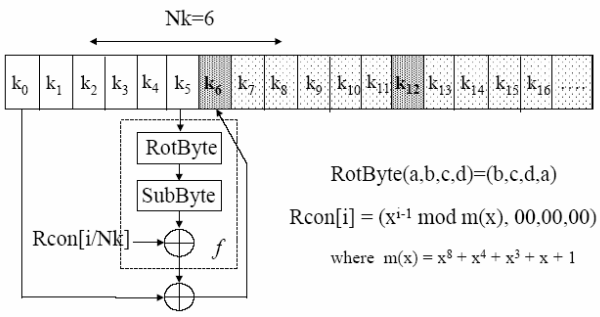


Expansion de la clé (1)



$$k_i = k_{i-Nk} \oplus k_{i-1}$$

Expansion de la clé (2)



RotByte(a,b,c,d)=(b,c,d,a)

Rcon[i] = (x^{i-1} mod m(x), 00,00,00)

where $m(x) = x^8 + x^4 + x^3 + x + 1$

$$k_i = k_{i-Nk} \oplus f(k_{i-1})$$

Rijndael

Avantages et limites

Avantages et limites

- Principaux avantages
  - performance très élevée
  - possibilité de réalisation en "Smart Card" avec peu de code
  - possibilité de parallélisme
  - pas d'opérations arithmétiques: décalages et XOR seulement
  - n'utilise pas de composants d'autres cryptosystèmes
  - n'est pas fondé sur d'obscures relations entres opérations
  - peu de possibilités d'insertion de trappes

Avantages et limites

- Principaux avantages
  - possibilité de l'utiliser comme fonction de hachage
  - le nombre de rondes peut facilement être augmenté si requis
  - Pas de clés faibles
  - Résistance à la cryptanalyse différentielle et linéaire
  - Faible propagation de patrons d'activité

Avantages et limites

- Limites
  - le décryptage est plus difficile à implanter en "Smart Card"
  - code et tables différents pour l'encryptage et le décryptage
  - dans une réalisation en matériel, il y a peu de réutilisation des circuits d'encryptage pour effectuer le décryptage

Chiffrements par flux

Il existe deux types de chiffrement à clé symétrique:

Le chiffrement par blocs:

L'opération de chiffrement s'effectue sur des blocs de texte clair (ex:le DES -blocs de 64 bits).

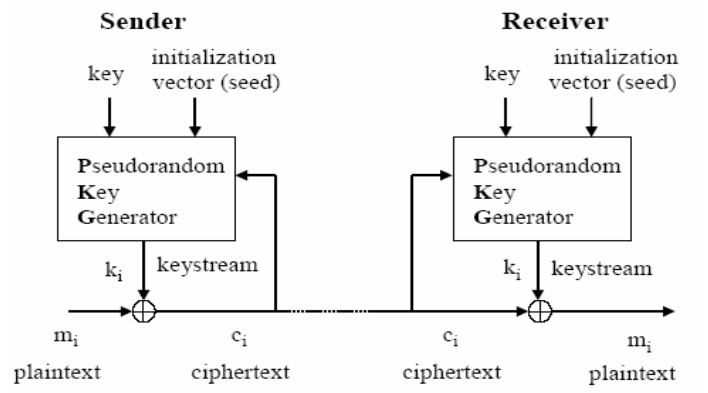
Le chiffrement par stream:

L'opération de chiffrement s'opère sur chaque élément du texte clair (caractère, bits).

Chiffrement d'un bit/caractère à la fois

La structure d'un chiffrement par stream repose sur un générateur de clé qui produit une séquence de clés k_1, k_2, …, k_i

Chiffrement par flux typique



La sécurité du chiffrement dépend de la qualité du générateur :

si k_i=0 ∀i, M=C

si la séquence des clés k_i est ∞ et complètement aléatoire, on obtient un One-Time-Pad.

En pratique, on se situe entre les deux, c 'est-à-dire une séquence pseudo aléatoire.

Propagation des erreurs:

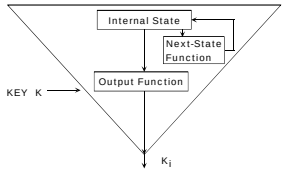
Une erreur dans C_i n'affecte qu'1 bit de M_i

La perte ou l'ajout d'un bit de C_i affecte tous les bits suivants de M après déchiffrement.

$$c_i = m_i \oplus k_i$$

$$m_i = c_i \oplus k_i$$

- Le générateur de clé peut être considéré comme une machine à états finis.
- Un exemple de ce type de générateur est le FSR (Feedback Shift Register)



Un FSR est constitué de

m sections contenant chacune 1 bit (état interne).

D'une fonction de retour (feedback).

A chaque impulsion de l'horloge, les éléments des sections sont décalés d'une position vers la droite.

L'élément de la section 0 constitue l'output du FSR.

L'élément de la section M-1 est le résultat de la fonction de retour.

Les coefficients c_i ont pour valeur 0 ou 1 et définissent les sections qui seront prises en compte par la fonction.

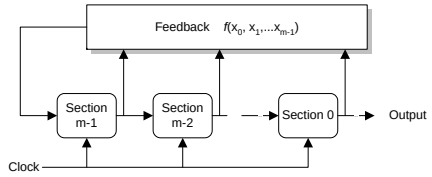
Les valeurs des coefficients ainsi que la valeur initiale du registre est fixée par la clé K symétrique.

La qualité du générateur est définie par sa période de retour, c'est à dire la période après laquelle la même série d'output est générée (=> générateur pseudo-aléatoire).

- Dans la cas où la fonction de retour est linéaire du type

$$f(x_0,...,x_{m-1}) = c_0 x_0 + c_1 x_1 + ... + c_{m-1} x_{m-1}$$
$$= \sum_{i=0}^{m-1} c_i x_i$$

- on parle de générateur LFSR (linear feedback shift register)



Questions

- Expliquer
  - Les modes du DES
  - AES
    - Choix de l'algorithme
    - Rijndael
    - Avantages et inconvénient du Rijndael

Références

- [Stallings] – ch 5
- http://csrc.nist.gov/CryptoToolkit/aes/
- http://www.rijndael.com/
- http://www.uqtr.ca/~delisle/Crypto/prives/blocs_modes.php
- http://home.ecn.ab.ca/~jsavard/crypto/co040401.htm