# Cryptographie symétrique AES : *"Advanced Encryption Standard"* GS15

Rémi Cogranne

Automne 2015-2016

# Sommaire

# Sommaire

# Historique

## Un petit d'histoire (les début de AES)

- 1993 : Premier travaux sur $\mathrm{G}\mathbb{F}_{256}$ à Louvain (Belgique).
- 1996 : Evaluation de la sécurité de DES conluant à la nécessité de trouver remplacant !
- été 1996 : L'université de Louvain couclut que son algorithme fonctionne avec des clés et des blocs de 128 bits au moins.
- fin 1996 : Implémentation de l'algorithme à l'UCL.
- Été 1997 : Appel à candidature international pour proposer un remplaçant à DES (blocks de tailles quelconque, clés de 128/196/256 bits).
- 01/1997 : Le NIST souhaite trouver un remplaçant DES.
- 09/1997 : Le NIST publie officiellement un appel à candidature international avec spécifications techniques (blocks de tailles quelconque, clés de 128/196/256 bits).

# Historique

## Un petit d'histoire (apogée de AES)

- 08/1998 : Première conférence sur les candidats de AES.
- Cinq réponse séléctionnées, classées ainsi :
  1. Rijndael (Daemen,Rijmen - BE), 10/12/14 rondes, Bloc : 128 bits : 86 pour / 10 contre ;
  2. Serpent (Anderson,Biham,Knudsen - UK) 32 rondes Bloc : 128 bits ; (Clé :de $k = 8x \in [0, 2048]$) : 59 pour / 7 contre ;
  3. Twofish (Schneier et al - US) 16 rondes Bloc : 128 bits : 31 pour / 21 contre ;
  4. RC6 (RSA Security Inc. - US) 20 rondes Bloc : 128 bits ; (Clé :de $k = 8x \in [0, 2048]$) : 23 pour / 37 contre ;
  5. MARS (Coppersmith/IBM - US) 16 rondes Bloc : 128 bits ; (Clé :de $k = 128 + 32k \in [128, 448]]$) : 13 pour / 84 contre ;
- 2000 : Le NIST choisi pour standard : AES-Rijndael.
- Complexité des meilleurs cryptanalyse de AES connues : $2^{128-1.9}$/128 bits, $2^{192-2.3}$/192 bits et $2^{256-1.6}$/256 bits.

# Conventions dans l'AES

## Une version simplifiée de Rijndael

- L'algorithme approuvé dans l'AES utilise des blocs de $128$ bits alors que le schéma de Rijndael prévoit l'utilisation de blocs et de clés de $128 + 32k \in [128, 256]$ bits.
- Dans la suite on appelle $\mathrm{Octet}$ un mot binaire de longueur 8 et $\mathrm{Mot}$ un mot binaire de 32 bits (4 $\mathrm{Octets}$).
- Une clé est constituée de $N_k = \{4, 6, 8\}$ $\mathrm{Mots}$ ($128$ , $196$ ou $256$ bits).
- AES utilise des blocs pour le message clair $N_b = 4$ $\mathrm{Mots}$ ($32$ bits).

# Représentation des blocs à chiffrer

## Exemple d'un bloc à chiffrer

Le message à chiffrer est découpé en blocs de $128$ bits qui sont représentés sous la forme forme d'une matrice de $4 \times 4$ octets comme suit :

| | | | |
|---|---|---|---|
| $a_{0,0}$ | $a_{0,1}$ | $a_{0,2}$ | $a_{0,3}$ |
| $a_{1,0}$ | $a_{1,1}$ | $a_{1,2}$ | $a_{1,3}$ |
| $a_{2,0}$ | $a_{2,1}$ | $a_{2,2}$ | $a_{2,3}$ |
| $a_{3,0}$ | $a_{3,1}$ | $a_{3,2}$ | $a_{3,3}$ |

Représentation matricielle des blocs du message à chiffrer.

Chaque élément $a_{i,j}$ représente un $\mathrm{Octet}$.

Chaque ligne ou colonne représente un $\mathrm{Mot}$.

# Conventions dans l'AES

## Notations et conventions

- On appelle $\mathrm{State}$ l'écriture matricielle du flux d'entrée sortie sur lequel opère chaque étape de l'algorithme.
- L'algorithme de l'AES consiste en une itération de $N_r$ tournées fonction taille de la clé :

$$N_r = \begin{cases} 10 & \text{quand} \quad \mathsf{N}_k = 4 \\ 12 & \text{quand} \quad \mathsf{N}_k = 6 \\ 14 & \text{quand} \quad \mathsf{N}_k = 8 \end{cases}$$

.

# Corps fini à $2^8 = 256$ élément $\mathbb{GF}_{256}$

## Rappel sur le corps fini de Rijndael $\mathbb{GF}_{256}$.

- AES est basé sur le corps fini $\mathbb{GF}_{256}$, défini par $\mathbb{Z}/2\mathbb{Z}[X]/ < M >$ avec $M = X^8 + X^4 + X^3 + X + 1$ est irréductible dans $\mathbb{Z}/2\mathbb{Z}[X]$ (huit racines distinctes $\in \mathbb{C}$).
- On a $\mathbb{GF}_{256} = \{a_7X^7 + a_6X^6 + a_5X^5 + a_4X^4 + a_3X^3 + a_2X^2 + a_1X + a_0\ ,\ a_i \in \mathbb{Z}/2\mathbb{Z}\}$.
- Dans AES les éléments $P \in \mathbb{GF}_{256}$ sont des $\mathrm{Octets}$ représentés par le vecteur $(a_7a_6a_5a_4a_3a_2a_1a_0)$.
- Cet ensemble à 255 élément non-nul et 1 élément nul $(00000000)$.
- L'élément générateur de ce corps est $X + 1$, d'où $\alpha = (00000011)$.
- À partir de l'élément générateur il est facile de calculer les inverses de chaque élément.

# $\alpha = (00000011)$ générateur de $\mathbb{GF}_{256}$

| $i$ | $\alpha^i$ | vecteur |
|---|---|---|
| x | $0$ | 0000 0000 |
| 0 | $x+1$ | 0000 0011 |
| 1 | $x^2+1$ | 0000 0101 |
| 2 | $x^3+x^2+x+1$ | 0000 1111 |
| 3 | $x^4+1$ | 0001 0001 |
| 4 | $x^5+x^4+x+1$ | 0011 0011 |
| 5 | $x^6+x^4+x^2+1$ | 0101 0101 |
| 6 | $x^7+x^6+x^5+x^4+x^3+x^2+x+1$ | 1111 0011 |
| 7 | $x^4+x^3+x^2+x$ | 0001 1110 |
| 8 | $x^5+x$ | 0010 0010 |
| 9 | $x^6+x^5+x^2+x$ | 0110 0110 |
| 10 | $x^7+x^5+x^3+x$ | 1010 1010 |
| 11 | $x^7+x^6+x^5+1$ | 1110 0001 |
| 12 | $x^5+x^4+x^3+x^2$ | 0011 1100 |
| 13 | $x^6+x^2$ | 0100 0100 |
| ... | ... | ... |
| ... | ... | ... |

# Principe Général de AES

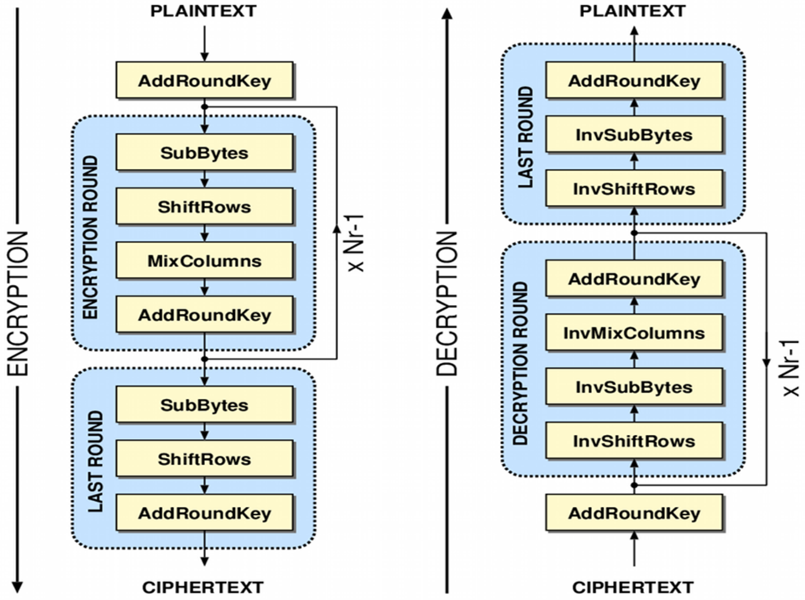
## Fonctionnement global de AES

L'algorithme de l'AES utilise les 5 fonctions élémentaire suivantes :

1. La fonction SubBytes qui est une fonction de substitution hautement non-linéaire (comme les S-box de DES) appliquée à chaque octet (8bits) du bloc $M$.
2. La fonction ShiftRows permet une diffusion en opérant un décalage circulaire différent sur chaque ligne.
3. La fonction MixColumnspermet une diffusion entre colonnes par opération linéaire sur les colonnes.
4. La fonction AddRoundKey qui combine par $\oplus$ ("xor") le bloc avec une sous-clés.

- La fonction KeyExpansion dont l'utilisation permet de créer suffisamment de sous-clés à partir d'une unique clé de 128/196/256 bits.

# Schéma de fonctionnement de AES



# Représentation algorithmique de AES

## Pseudo-code de l'algorithme

**procedure** Encryption AES($\text{State}, K$)
    $\text{RoundKeys} \leftarrow \text{KeyExpansion}(K)$
    $\text{State} \leftarrow \text{AddRoundKey}(\text{State}, \text{RoundKeys}[0])$
    **while** $(r < N_r)$ **do** ▷ Les $N_r$ rondes
        $\text{State} \leftarrow \text{SubBytes}(\text{State})$
        $\text{State} \leftarrow \text{ShiftRows}(\text{State})$
        $\text{State} \leftarrow \text{MixColumns}(\text{State})$
        $\text{State} \leftarrow \text{AddRoundKey}(\text{State}, \text{RoundKeys}[r])$
        $r \leftarrow r + 1$
    **end while**
    $\text{State} \leftarrow \text{SubBytes}(\text{State})$ ▷ Ronde finale
    $\text{State} \leftarrow \text{ShiftRows}(\text{State})$
    $\text{State} \leftarrow \text{AddRoundKey}(\text{State}, \text{RoundKeys}[r])$
    **return** State
**end procedure**

# Sommaire

# Fonction SubBytes

## Définition de la fonction SubBytes

$b_{i,j} = \mathrm{SubBytes}(\mathrm{m_{i,j}}) = \mathbf{A} \times m_{i,j}^{-1} \oplus c$ :

$$b_{i,j} = \begin{pmatrix} b_{i,j}[7] \\ b_{i,j}[6] \\ b_{i,j}[5] \\ b_{i,j}[4] \\ b_{i,j}[3] \\ b_{i,j}[2] \\ b_{i,j}[1] \\ b_{i,j}[0] \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} m_{i,j}^{-1}[0] \\ m_{i,j}^{-1}[1] \\ m_{i,j}^{-1}[2] \\ m_{i,j}^{-1}[3] \\ m_{i,j}^{-1}[4] \\ m_{i,j}^{-1}[5] \\ m_{i,j}^{-1}[6] \\ m_{i,j}^{-1}[7] \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

## Exercice / Question

Soit $m_{i,j} = (00010001)$ calculer $\mathrm{SubBytes}(\mathrm{m_{i,j}})$. Idem avec $m_{i,j}(00000011)$ et $m_{i,j}(01010011)$.

# Fonction SubBytes

## Réponse (1/2, calcul d'inverse)

L'inverse de $m_{i,j} = (00010001)$ est $(10110100)$ car :

$$(X^4+1)\times(X^7+X^5+X^4+X^2)$$
$$=X^{11}+X^9+X^8+X^7+X^6+X^5+X^4+X^2$$
$$=(X^8+X^4+X^3+X+1)\times(X^3+X+1)+1$$

L'inverse de $m_{i,j} = (00000011)$ est $(11110110)$ car :

$$(X^2+1)\times(X^7+X^6+X^5+X^4+X^2+X) = X^8+X^4+X^3+X$$

Or, $X^8+X^4+X^3+X = (X^8+X^4+X^3+X+1)+1$

L'inverse de $m_{i,j} = (01010011)$ est $(11001010)$ car :

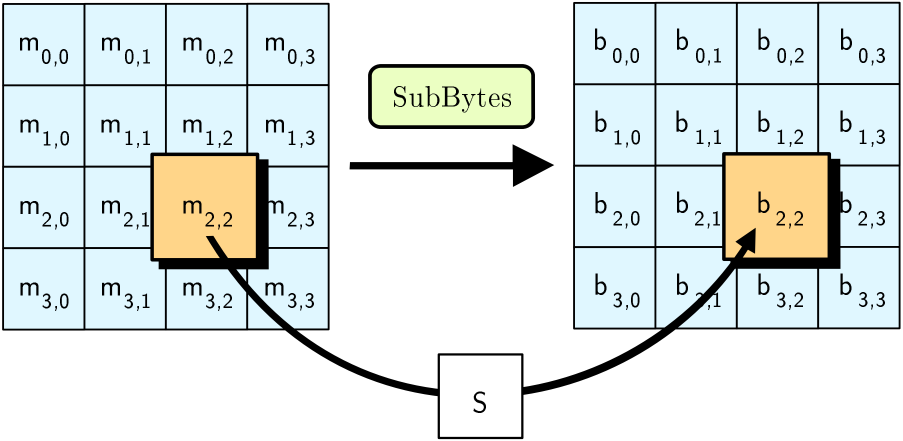
$$(X^6+X^4+X+1)\times(X^7+X^6+X^3+X)$$
$$=X^{13}+X^{12}+X^{11}+X^{10}+X^9+X^8+X^6+X^5+X^4+X^3+X^2+X$$
$$=(X^8+X^4+X^3+X+1)\times(X^5+X^4+X^3+X^2+1)+1$$

# Fonction SubBytes

## Réponse (2/2, transformation affine)

Pour $m_{i,j} = (00010001)$ on a donc
$\mathrm{SubBytes(m_{i,j})} = \mathrm{A} \times (10110100)^T + c = (11011000)$.
Pour $m_{i,j} = (00000011)$ on a donc
$\mathrm{SubBytes(m_{i,j})} = \mathrm{A} \times (11110110)^T + c = (01000111)$.
Pour $m_{i,j} = (01010011)$ on a donc
$\mathrm{SubBytes(m_{i,j})} = \mathrm{A} \times (11001010)^T + c = (00101110)$

# Représentation graphique de la fonction SubBytes




Application de la S-box SubBytes sur un octet.

# Fonction InvSubBytes

## Définition de la fonction InvSubBytes

$m_{i,j} = \mathrm{InvSubBytes}(\mathrm{b_{i,j}}) = (\mathbf{A}' \times b_{i,j} \oplus c')^{-1}$ avec :

$$\mathbf{A}' = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \quad \text{et} \quad c' = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

## Exercice / Question

Soit $b_{i,j} = (11011000)$ calculer $\mathrm{InvSubBytes}(\mathrm{m_{i,j}})$. Idem avec $b_{i,j} = (01000111)$ et $b_{i,j} = (00101110)$.

# Fonction SubBytes

## Réponse

On calcule d'abord $\mathbf{A}' \times b_{i,j} \oplus c'$. On calcule ensuite l'inverse du résultat trouvé.
Dans notre cas on a $\mathbf{A}' \times (11011000)^T \oplus c' = (10110100)^T$
L'inverse de $(10110100)$ est $(00010001)$ car :

$$(X^4+1) \times (X^7+X^5+X^4+X^2)$$
$$= (X^8+X^4+X^3+X+1) \times (X^3+X+1) + 1$$

Deuxième exemple : $\mathbf{A}' \times (01000111)^T \oplus c' = (11110110)^T$.
L'inverse de $(11110110)$ est $(00000011)$ car :

$$(X^2+1) \times (X^7+X^6+X^5+X^4+X^2+X)$$
$$= (X^8+X^4+X^3+X+1) + 1$$

Dernier exemple : $\mathbf{A}' \times (01000111)^T \oplus c' = (11001010)^T$.
L'inverse de $(11001010)$ est $(01010011)$ car :

$$(X^6+X^4+X+1) \times (X^7+X^6+X^3+X)$$
$$= (X^8+X^4+X^3+X+1) \times (X^5+X^4+X^3+X^2+1) + 1$$

# "Pré-caculls" des fonctions SubBytes et InvSubBytes

## Tabulation de la fonction SubBytes

| | $x0$ | $x1$ | $x2$ | $x3$ | $x4$ | $x5$ | $x6$ | $x7$ | $x8$ | $x9$ | $xa$ | $xb$ | $xc$ | $xd$ | $xe$ | $xf$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $0x$ | 63 | 7c | 77 | 7b | f2 | 6b | 6f | c5 | 30 | 01 | 67 | 2b | fe | d7 | ab | 76 |
| $1x$ | ca | 82 | c9 | 7d | fa | 59 | 47 | f0 | ad | d4 | a2 | af | 9c | a4 | 72 | c0 |
| $2x$ | b7 | fd | 93 | 26 | 36 | 3f | f7 | cc | 34 | a5 | e5 | f1 | 71 | d8 | 31 | 15 |
| $3x$ | 04 | c7 | 23 | c3 | 18 | 96 | 05 | 9a | 07 | 12 | 80 | e2 | eb | 27 | b2 | 75 |
| $4x$ | 09 | 83 | 2c | 1a | 1b | 6e | 5a | a0 | 52 | 3b | d6 | b3 | 29 | e3 | 2f | 84 |
| $5x$ | 53 | d1 | 00 | ed | 20 | fc | b1 | 5b | 6a | cb | be | 39 | 4a | 4c | 58 | cf |
| $6x$ | d0 | ef | aa | fb | 43 | 4d | 33 | 85 | 45 | f9 | 02 | 7f | 50 | 3c | 9f | a8 |
| $7x$ | 51 | a3 | 40 | 8f | 92 | 9d | 38 | f5 | bc | b6 | da | 21 | 10 | ff | f3 | d2 |
| $8x$ | cd | 0c | 13 | ec | 5f | 97 | 44 | 17 | c4 | a7 | 7e | 3d | 64 | 5d | 19 | 73 |
| $9x$ | 60 | 81 | 4f | dc | 22 | 2a | 90 | 88 | 46 | ee | b8 | 14 | de | 5e | 0b | db |
| $ax$ | e0 | 32 | 3a | 0a | 49 | 06 | 24 | 5c | c2 | d3 | ac | 62 | 91 | 95 | e4 | 79 |
| $bx$ | e7 | c8 | 37 | 6d | 8d | d5 | 4e | a9 | 6c | 56 | f4 | ea | 65 | 7a | ae | 08 |
| $cx$ | ba | 78 | 25 | 2e | 1c | a6 | b4 | c6 | e8 | dd | 74 | 1f | 4b | bd | 8b | 8a |
| $dx$ | 70 | 3e | b5 | 66 | 48 | 03 | f6 | 0e | 61 | 35 | 57 | b9 | 86 | c1 | 1d | 9e |
| $ex$ | e1 | f8 | 98 | 11 | 69 | d9 | 8e | 94 | 9b | 1e | 87 | e9 | ce | 55 | 28 | df |
| $fx$ | 8c | a1 | 89 | 0d | bf | e6 | 42 | 68 | 41 | 99 | 2d | 0f | b0 | 54 | bb | 16 |

# “Pré-caculls” des fonctions SubBytes et InvSubBytes

## Tabulation de la fonction InvSubBytes

| | $x0$ | $x1$ | $x2$ | $x3$ | $x4$ | $x5$ | $x6$ | $x7$ | $x8$ | $x9$ | $xa$ | $xb$ | $xc$ | $xd$ | $xe$ | $xf$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $0x$ | 52 | 09 | 6a | d5 | 30 | 36 | a5 | 38 | bf | 40 | a3 | 9e | 81 | f3 | d7 | fb |
| $1x$ | 7c | e3 | 39 | 82 | 9b | 2f | ff | 87 | 34 | 8e | 43 | 44 | c4 | de | e9 | cb |
| $2x$ | 54 | 7b | 94 | 32 | a6 | c2 | 23 | 3d | ee | 4c | 95 | 0b | 42 | fa | c3 | 4e |
| $3x$ | 08 | 2e | a1 | 66 | 28 | d9 | 24 | b2 | 76 | 5b | a2 | 49 | 6d | 8b | d1 | 25 |
| $4x$ | 72 | f8 | f6 | 64 | 86 | 68 | 98 | 16 | d4 | a4 | 5c | cc | 5d | 65 | b6 | 92 |
| $5x$ | 6c | 70 | 48 | 50 | fd | ed | b9 | da | 5e | 15 | 46 | 57 | a7 | 8d | 9d | 84 |
| $6x$ | 90 | d8 | ab | 00 | 8c | bc | d3 | 0a | f7 | e4 | 58 | 05 | b8 | b3 | 45 | 06 |
| $7x$ | d0 | 2c | 1e | 8f | ca | 3f | 0f | 02 | c1 | af | bd | 03 | 01 | 13 | 8a | 6b |
| $8x$ | 3a | 91 | 11 | 41 | 4f | 67 | dc | ea | 97 | f2 | cf | ce | f0 | b4 | e6 | 73 |
| $9x$ | 96 | ac | 74 | 22 | e7 | ad | 35 | 85 | e2 | f9 | 37 | e8 | 1c | 75 | df | 6e |
| $ax$ | 47 | f1 | 1a | 71 | 1d | 29 | c5 | 89 | 6f | b7 | 62 | 0e | aa | 18 | be | 1b |
| $bx$ | fc | 56 | 3e | 4b | c6 | d2 | 79 | 20 | 9a | db | c0 | fe | 78 | cd | 5a | f4 |
| $cx$ | 1f | dd | a8 | 33 | 88 | 07 | c7 | 31 | b1 | 12 | 10 | 59 | 27 | 80 | ec | 5f |
| $dx$ | 60 | 51 | 7f | a9 | 19 | b5 | 4a | 0d | 2d | e5 | 7a | 9f | 93 | c9 | 9c | ef |
| $ex$ | a0 | e0 | 3b | 4d | ae | 2a | f5 | b0 | c8 | eb | bb | 3c | 83 | 53 | 99 | 61 |
| $fx$ | 17 | 2b | 04 | 7e | ba | 77 | d6 | 26 | e1 | 69 | 14 | 63 | 55 | 21 | 0c | 7d |

# Fonction ShiftRows

## Définition de la fonction ShiftRows

Soit un tableau state représenté sous la forme :

| | | | |
|---|---|---|---|
| $a_{0,0}$ | $b_{0,1}$ | $b_{0,2}$ | $b_{0,3}$ |
| $b_{1,0}$ | $b_{1,1}$ | $b_{1,2}$ | $b_{1,3}$ |
| $b_{2,0}$ | $b_{2,1}$ | $b_{2,2}$ | $b_{2,3}$ |
| $b_{3,0}$ | $b_{3,1}$ | $b_{3,2}$ | $b_{3,3}$ |

D'une façon générale la fonction ShiftRowsintroduit un décalage circulaire de $c_i$ position de la $i$-ième ligne avec $c_i$ donné par :

| Nb | $c_0$ | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|---|
| 4 | 0 | 1 | 2 | 3 |
| 6 | 0 | 1 | 2 | 3 |
| 8 | 0 | 1 | 3 | 4 |

# Fonction ShiftRows

## Définition de la fonction ShiftRows

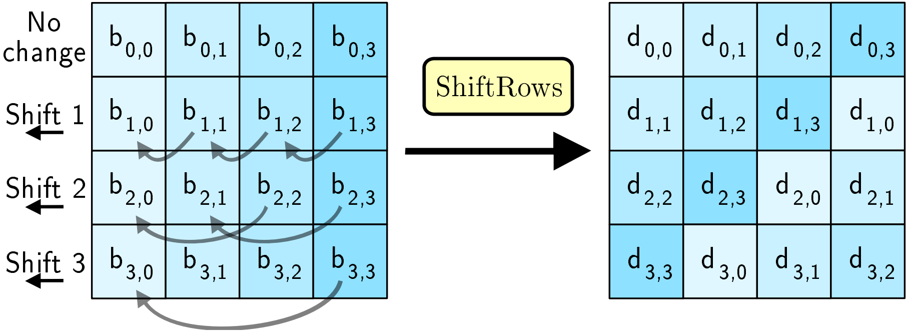
Par exemple pour le schéma AES avec des clés des messages de 128 la fonction ShiftRows peut être représenté sous la forme :

$$\mathrm{ShiftRows} \ : \begin{pmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} \rightarrow \begin{pmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,1} & b_{1,2} & b_{1,3} & b_{1,0} \\ b_{2,2} & b_{2,3} & b_{2,0} & b_{2,1} \\ b_{3,3} & b_{3,0} & b_{3,1} & b_{3,2} \end{pmatrix}$$

## Remarque

L'effet de cette transformation, quand on l'applique dans plusieurs tournées, est une diffusion du chiffrement au sein des lignes (dissipations des redondances statistique du message après application de la fonction).

# Représentation graphique de la fonction ShiftRows




Application de la fonction ShiftRows sur un State (16 Octets).

# Fonction InvShiftRows

**Question / Exercice (hyper-facile)**

Donner la définition de la fonction $\mathrm{InvShiftRows}$, telle que $\mathrm{InvShiftRows} \circ \mathrm{ShiftRows} = Id$ ?

# Fonction InvShiftRows

## Question / Exercice (hyper-facile)

Donner la définition de la fonction InvShiftRows, telle que $\mathrm{InvShiftRows} \circ \mathrm{ShiftRows} = Id$ ?

## Réponse : définition de la fonction InvShiftRows

La réponse est évidémment un décalage circulaire "vers la droite de $c_i$ positions" ou "vers la gauche de $\mathrm{Nb} - c_i$ positions". Par exemple, pour le schéma AES avec des clés des messages de 128 la fonction InvShiftRows peut être représenté sous la forme :

$$\mathrm{ShiftRows} \; : \begin{pmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} \rightarrow \begin{pmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,3} & b_{1,0} & b_{1,1} & b_{1,2} \\ b_{2,2} & b_{2,3} & b_{2,0} & b_{2,1} \\ b_{3,1} & b_{3,2} & b_{3,3} & b_{3,0} \end{pmatrix}$$

# Fonction MixColumns

## Définition de la fonction MixColumns

Soit $\mathbf{b}_j = (b_{0,j}, b_{1,j}, b_{2,j}, b_{3,j})^T$ la $j$-ième colone de l'état state. On identifie la colonne $\mathbf{b_j}$ avec le polynôme :

$$b_{0,j} + b_{1,j}X + b_{2,j}X^2 + b_{3,j}X^3 \in \mathrm{GF}_{2^8}$$

La transformation MixColumns est définie par :

$$\mathrm{MixColumns} \ : \mathbf{b_j} \rightarrow \mathbf{p(X)b_j} \bmod \mathbf{X^4+1}$$

avec

$$p(X) = 3X^3 + X^2 + X + 2.$$

Ce qui correspond à une transformation linéaire dans $\mathrm{GF}_{2^8}^4$ :

$$\mathbf{b_j} = \begin{pmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{pmatrix} \leftarrow \begin{pmatrix} \{02\} & \{03\} & \{01\} & \{01\} \\ \{01\} & \{02\} & \{03\} & \{01\} \\ \{01\} & \{01\} & \{02\} & \{03\} \\ \{03\} & \{01\} & \{01\} & \{02\} \end{pmatrix} \times \begin{pmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{pmatrix}$$

# Fonction MixColumns

## Remarque

L'effet de cette transformation, quand on l'applique dans plusieurs tournées, est une diffusion du chiffrement au sein des colonnes.

## Exercice / Question

Soit les Mots :

$m_1 = 0x(d4, bf, 5d, 30) =$
$(1101\,0100, 1011\,1111, 0101\,1101, 0011\,0000)$

$m_2 = 0x(db, 13, 53, 45) =$
$(1101\,1011, 0001\,0011, 0101\,0011, 0100\,0101)$

Appliquer la fonction MixColumns à ces deux mots.

# Calculs de la fonction MixColumns

## Réponse / Solution (1/5)

En utilisant la forme matricielle (plus simple) on a :

$$d4 \times \{02\} = 1101\,0100 \times 0000\,0010(<<1) = 1\,1010\,1000$$

Or puisque $X^8 = M(= X^8 + X^4 + X^3 + X + 1) + X^4 + X^3 + X + 1$ on a $d4 \times \{02\} = 1010\,1000 \oplus 0001\,1011 = 1011\,0011 = 0x\{b3\}$.

De la même façon on a :

$$bf \times \{03\} = 1011\,1111 \times 0000\,0011 = 1\,0111\,1110 \oplus 1011\,1111$$

Or puisque $X^8 = M + X^4 + X^3 + X + 1$ on a $bf \times \{03\} = 0111\,1110 \oplus 1011\,1111 \oplus 0001\,1011 = 1101\,1010 = 0x\{da\}$.

Enfin $5d \times \{01\} = 5d$ et $30 \times \{01\} = 30$.

En faisant la somme obtient alors :

$b3+da+5d+30 = 1011\,0011 \oplus 1101\,1010 \oplus 0101\,1101 \oplus 0011\,0000 = 0000\,0100 = \{04\}$.

# Calculs de la fonction MixColumns

## Réponse / Solution (2/5)

Pour la “seconde ligne” on a :

$$bf \times \{02\} = 1011\,1111 \times 0000\,0010 (<< 1) = 1\,0111\,1110$$
$$= 0111\,1110 \oplus 0001\,1011 = 0110\,0101 = \{65\}$$

De la même façon on a :

$$5d \times \{03\} = 0101\,1101 \times 0000\,0011 = 1011\,1010 \oplus 0101\,1101$$
$$= 1110\,0111 = 0x\{e7\}.$$

Enfin $d4 \times \{01\} = d4$ et $30 \times \{01\} = 30$.
En faisant la somme obtient alors :
$d4 + 65 + e7 + 30 = 1101\,0100 \oplus 0110\,0101 \oplus 1110\,0111 \oplus 0011\,0000 = 0110\,0110 = \{66\}$.

# Calculs de la fonction MixColumns

## Réponse / Solution (3/5)

Pour la “troisième ligne” on a :

$5d \times \{02\} = 0101\,1101 \times 0000\,0010 (<< 1) = 1011\,1010 = 0x\{ba\}$.

De la même façon on a :

$$\begin{aligned} 30 \times \{03\} &= 0011\,0000 \times 0000\,0011 = 0110\,0000 \oplus 0011\,0000 \\ &= 0101\,0000 = 0x\{50\} \end{aligned}$$

Enfin $d4 \times \{01\} = d4$ et $bf \times \{01\} = bf$.

En faisant la somme obtient alors :

$d4 + bf + ba + 50 = 1101\,0100 \oplus 1011\,1111 \oplus 1011\,1010 \oplus 0101\,0000 = 1000\,0001 = \{81\}$.

# Calculs de la fonction MixColumns

## Réponse / Solution (4/5)

Pour la “dernière ligne” on a :
$30 \times \{02\} = 0011\,0000 \times 0000\,0010 (<< 1) = 0110\,0000 = 0x\{60\}$.
De la même façon on a :
$d4 \times \{03\} = 1101\,0100 \times 0000\,0011 = 1\,1010\,1000 \oplus 1101\,0100$
Or puisque $X^8 = M + X^4 + X^3 + X + 1$ on a
$d4 \times \{03\} = 1010\,1000 \oplus 1101\,0100 \oplus 0001\,1011 = 0110\,0111 = \{67\}$.
Enfin $bf \times \{01\} = bf$ et $5d \times \{01\} = 5d$.
En faisant la somme obtient alors :
$67 + bf + 5d + 60 = 0110\,0111 \oplus 1011\,1111 \oplus 0101\,1101 \oplus 0110\,0000 = 1110\,0101 = \{e5\}$.

## Résultat

La colonne “mixée” est donc le mot $\mathbf{b}_j = 0x(04, 66, 81, e5)^T$.

# Calculs de la fonction MixColumns

## Réponse / Solution (5/5)

Pour la “première ligne” du second mot on a :

$$db \times \{02\} = 1101\,1011 \times 0000\,0010 (<< 1) = 1\,1011\,0110$$
$$= 1011\,0110 \oplus 0001\,1011 = 1010\,1101 = 0x\{ad\}$$

De la même façon on a :

$$13 \times \{03\} = 0001\,0011 \times 0000\,0011 = 0011\,0101 = 0x\{35\}$$

Enfin $53 \times \{01\} = 53$ et $45 \times \{01\} = 45$.

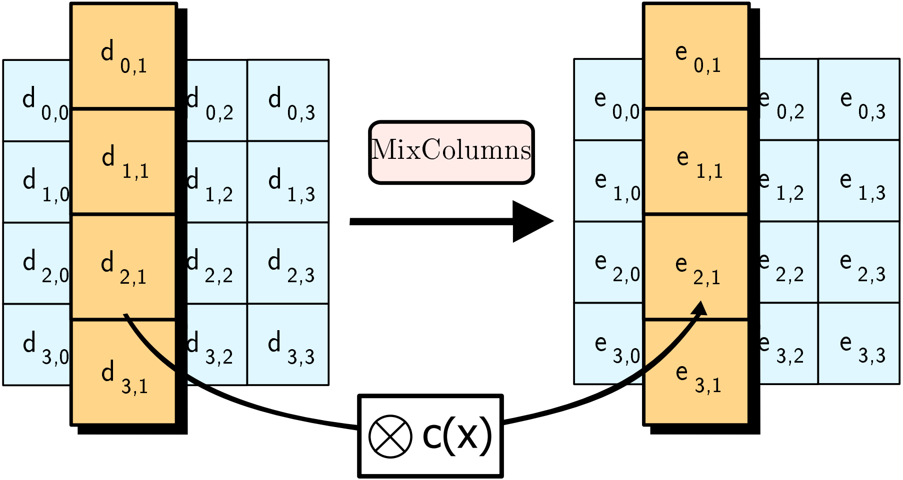
En faisant la somme obtient alors :

$ad + 35 + 53 + 45 = 1010\,1101 \oplus 0011\,0101 \oplus 0101\,0011 \oplus 0100\,0101 = 1000\,1110 = \{8e\}$.

## Résultat

La colonne “mixée” est ... le mot $\mathbf{b}_j = 0x(8e, 4d, a1, bc)^T$.

# Représentation graphique de la fonction MixColumns




Application de la fonction MixColumns sur un Mot (4 Octets).

# Fonction InvMixColumns

## Définition de la fonction InvMixColumns

Soit $\mathbf{b}_j = (b_{0,j}, b_{1,j}, b_{2,j}, b_{3,j})^T$ la $j$-ième colone de l'état state.
La fonction InvMixColumns est la fonction telle que :

$$\text{InvMixColumns} \circ \text{MixColumns} = Id$$

Cette fonction est définie comme la transformation linéaire dans $\mathrm{GF}_{2^8}^4$ :

$$\mathbf{b_j} = \begin{pmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{pmatrix} \leftarrow \begin{pmatrix} \{14\} & \{11\} & \{13\} & \{09\} \\ \{09\} & \{14\} & \{11\} & \{13\} \\ \{13\} & \{09\} & \{14\} & \{11\} \\ \{11\} & \{13\} & \{09\} & \{14\} \end{pmatrix} \times \begin{pmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{pmatrix}$$

# "Pré-cacullS" des fonctions SubBytes et InvSubBytes

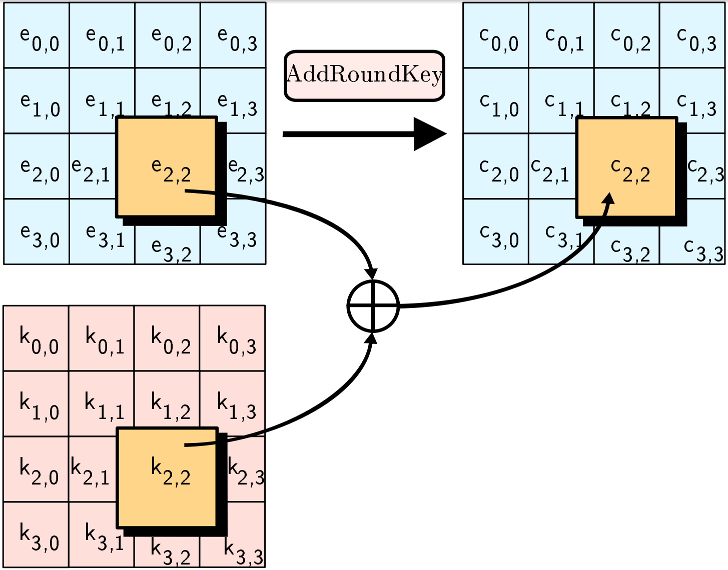
## Tabulation

On peut constater que l'on utilise uniquement pour les fonctions MixColumns et InvMixColumns des multiplications par $(1), 2, 3, 9, 11, 13$ et $14$.
Il est donc possible de tabuler les $6$ multiplications utilisées (ce qui requiert $6 \times 256$ éléments) pour ramener les MixColumns et InvMixColumns à de simple sommes ($\oplus$ booléens).

## Exercice / Question

Soit les Mots :
$m_1 = 0x(04, 66, 81, e5) =$
$(0000\,0100, 0110\,0110, 1000\,0001, 1110\,0101)$
$m_2 = 0x(8e, 4d, a1, bc) =$
$(1000\,1110, 0100\,1101, 1010\,0001, 1011\,1100)$
Appliquer la fonction InvMixColumns à ces deux mots pour retrouver les mots précédents.

# Représentation graphique de la fonction AddRoundKey



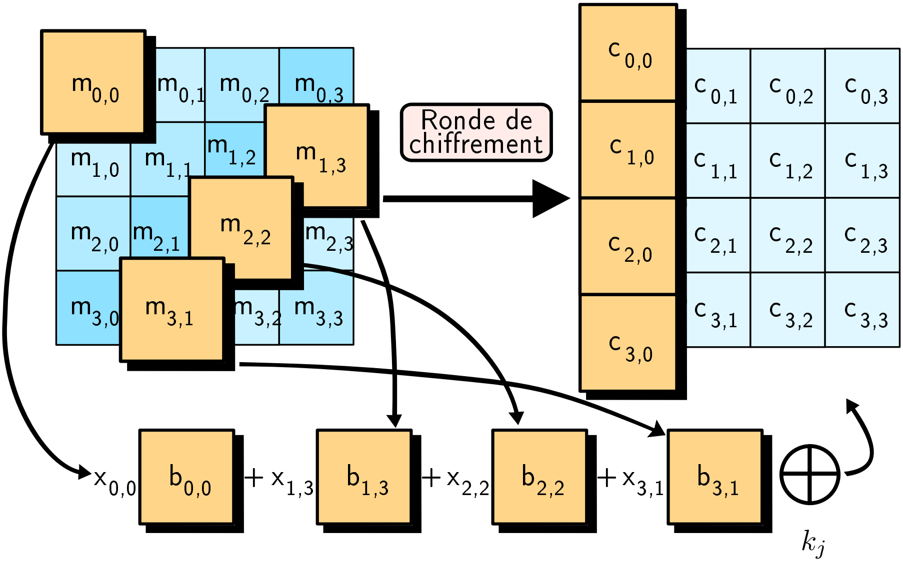

Application de la fonction AddRoundKey sur un octet.

# Implémentation rapide du chiffrement AES

## Calcul rapide d'une ronde de chiffrement

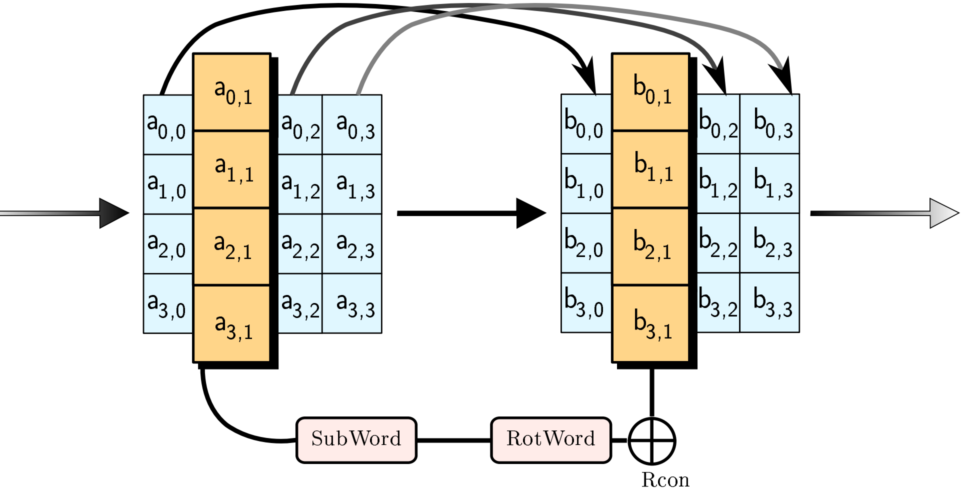
- En sauvegardant (en “dur”) les tables correspondant aux S-box $\mathrm{SubBytes}$ et $\mathrm{InvSubBytes}$, il est possible de simplifier le calcul d'une ronde de chiffrement.
- De manière pratique, un ordinateur “moderne” (pouvant sauvegarder ces tables) peut éxécuter une rounde de chiffrement par un simple $\oplus$ $(\mathrm{xor})$, quatre produits matriciel et $16$ consultations de la table correspondant à la S-box $\mathrm{SubBytes}$.

# Représentation graphique d'une ronde de chiffrement




Calcul rapide d'une ronde de chiffrement.

# Représentation graphique de la fonction KeyExpansion




Application de la fonction KeyExpansion sur un mot (4 octets).