

## RIJNDAEL – AES

Nous décrivons les grandes lignes de l'algorithme de chiffrement symétrique RIJNDAEL, utilisé comme *Advanced Encryption Standard* (AES) depuis 2002, en remplacement de DES et TRIPLE DES. C'est un chiffrement itératif par bloc, qui admet des clés de 128, 192 ou 256 bits ; le nombre  $\text{Nb}$  de tours est respectivement de 10, 12 ou 14 pour chacune des tailles de clé.

Chaque bloc contient  $16 \times 8 = 128$  bits et est constitué de 16 octets, rangés dans un tableau  $4 \times 4$  :

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

On considère le corps fini  $\mathbb{F}_{256} = \mathbb{F}_2[X]/(T)$ , défini par le polynôme irréductible  $T(X) = X^8 + X^4 + X^3 + X + 1 \in \mathbb{F}_2[X]$  ; on note  $\alpha$  la classe de  $X$  dans  $\mathbb{F}_{256}$ . Chaque octet  $(b_7, \dots, b_0) \in (\mathbb{F}_2)^8$  est identifié avec l'élément  $\sum_{i=0}^7 b_i \alpha^i \in \mathbb{F}_{256}$ .

RIJNDAEL utilise 4 opérations élémentaires, agissant successivement sur un bloc donné ; elles sont spécifiées plus loin :

- **SubBytes** (*S*-box),
- **ShiftRows**,
- **MixColumns**,
- **AddRoundKey**.

Un algorithme de *cadencement de clé* calcule à partir de la clé  $K$  une suite de  $\text{Nb} + 1$  sous-clés de tour  $(K_0, \dots, K_{\text{Nb}})$ , comportant toutes 16 octets. Le texte à chiffrer est découpé en blocs de 16 octets et, pour chaque bloc clair, on effectue les opérations suivantes pour produire un bloc chiffré de même longueur :

- (1) On initialise un tableau  $4 \times 4$  avec 16 octets de texte clair ; on applique successivement sur ce tableau les opérations suivantes.
- (2) **AddRoundKey** $(K_0)$ .
- (3) On effectue  $\text{Nb} - 1$  tours comportant les 4 étapes **SubBytes**, **ShiftRows**, **MixColumns**, **AddRoundKey** $(K_i)$  dans cet ordre, pour  $i = 1, \dots, \text{Nb} - 1$  ;
- (4) Un dernier tour ne comporte plus que 3 étapes : **SubBytes**, **ShiftRows**, **AddRoundKey** $(K_{\text{Nb}})$ .
- (5) Le contenu actuel du tableau donne les 16 octets du texte chiffré.

**SubBytes (S-box):** opère indépendamment sur chacun des 16 octets. C'est la composée  $S = f \circ I$  des applications

$$I : \mathbb{F}_{256} \rightarrow \mathbb{F}_{256} \quad f : \mathbb{F}_{256} \simeq (\mathbb{F}_2)^8 \rightarrow (\mathbb{F}_2)^8 \simeq \mathbb{F}_{256}$$

$$x \mapsto \begin{cases} x^{-1} & \text{si } x \neq 0, \\ 0 & \text{si } x = 0, \end{cases} \quad \text{et} \quad x \mapsto Ax + B$$

où  $A$  est une matrice  $8 \times 8$  à coefficients dans  $\mathbb{F}_2$  et  $B \in (\mathbb{F}_2)^8$  :

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

**ShiftRows:** fait subir une permutation circulaire vers la gauche aux lignes du tableau, respectivement de 0, 1, 2, 3 cases :

$$\begin{array}{|c|c|c|c|} \hline a_{0,0} & a_{0,1} & a_{0,1} & a_{0,2} \\ \hline a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ \hline a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ \hline a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|c|} \hline a_{0,0} & a_{0,1} & a_{0,1} & a_{0,2} \\ \hline a_{1,1} & a_{1,2} & a_{1,3} & a_{1,0} \\ \hline a_{2,2} & a_{2,3} & a_{2,0} & a_{2,1} \\ \hline a_{3,3} & a_{3,0} & a_{3,1} & a_{3,2} \\ \hline \end{array}$$

**MixColumns:** s'interprète comme une multiplication matricielle :

$$\begin{array}{|c|c|c|c|} \hline a_{0,0} & a_{0,1} & a_{0,1} & a_{0,2} \\ \hline a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ \hline a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ \hline a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|c|} \hline b_{0,0} & b_{0,1} & b_{0,1} & b_{0,2} \\ \hline b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ \hline b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ \hline b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \\ \hline \end{array}$$

où

$$\begin{pmatrix} b_{0,0} & b_{0,1} & b_{0,1} & b_{0,2} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} = \begin{pmatrix} \alpha & \alpha + 1 & 1 & 1 \\ 1 & \alpha & \alpha + 1 & 1 \\ 1 & 1 & \alpha & \alpha + 1 \\ \alpha + 1 & 1 & 1 & \alpha \end{pmatrix} \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,1} & a_{0,2} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

est le produit des matrices à coefficients dans  $\mathbb{F}_{256}$ .

**AddRoundKey( $K_i$ ):** addition bit à bit (XOR) de la clé de tour  $K_i$ , case par case.