AES
Advanced Encryption Standard

Le concours

En 1997, le NIST américain (*National Institute of Standards and Technology*) lance un concours ouvert à tous pour choisir un algorithme public qui sera utilisé dans le monde entier pour chiffrer des données.

Au départ, quinze candidats sont sélectionnés, de plusieurs pays (Etats-Unis, France, Belgique, Corée, Allemagne,...) et disposant de budgets variés (allant d'équipes universitaires jusqu'à IBM).

Les critères du concours étaient :

- la sécurité (résistance aux attaques connues) ;
- les coûts (libre de droits, coûts d'implémentation) ;
- l'algorithme (flexibilité, simplicité et élégance du système).

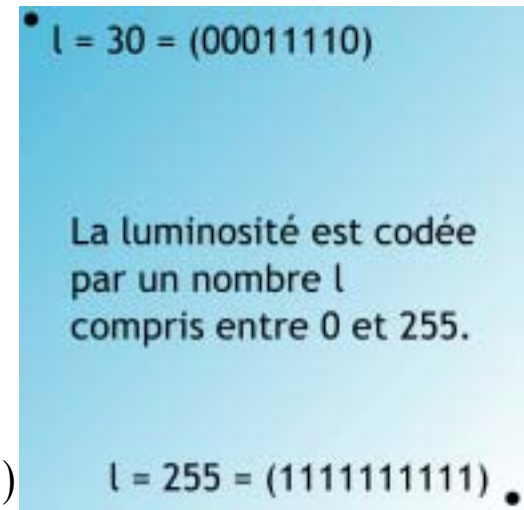
L'évaluation a été faite par la communauté internationale des cryptologues lors de conférences dans différents pays.

Le vainqueur est le Rijndael, un système créé par deux Belges : Joan Daemen et Vincent Rijmen, chercheurs au COSIC de la KUL.

Le principe du codage des informations

Un ordinateur enregistre toute information (texte, image, etc.) comme une suite de uns et de zéros. Les composants de ce *code binaire*, les bits, valant chacun 0 ou 1, sont regroupés par 8 pour former des *octets*. Chaque octet peut prendre 2^8 = 256 valeurs différentes.

$$\begin{pmatrix} d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \end{pmatrix}$$



Dans le système AES, les informations sont regroupées par *blocs* de 128 bits, c'est-à-dire 16 octets, placés dans un tableau.

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Une **clé**, dite clé maître, également de 128 bits, est utilisée pour coder les informations. Il s'agit d'une clé *secrète*, qui ne peut en aucun cas être révélée. Elle sert à générer les clés *esclaves*, qui seront réellement utilisées pour le cryptage.

On définit une opération particulière sur les octets, l'addition bit à bit modulo 2 ou, en langage informatique, le **OU exclusif (XOR).**

$\oplus$	0	1
0	0	1
1	1	0

$$\begin{array}{rcccccccc} & (1 & 0 & 0 & 1 & 1 & 0 & 1 & 1) \\ \oplus & (1 & 1 & 1 & 0 & 0 & 0 & 0 & 1) \\ \hline = & (0 & 1 & 1 & 1 & 1 & 0 & 1 & 0) \end{array}$$

10 tours de 4 étapes

Avant tout, une clé esclave est additionnée au bloc.

1. Transformation non linéaire d'octets : chaque octet est transformé par la même transformation non linéaire S.

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \rightarrow \begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{pmatrix}$$

Où b_{00} est égal à S(a_{00}), etc.

2. Décalage des lignes

$$\begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{pmatrix} \quad \begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{11} & b_{12} & b_{33} & b_{10} \\ b_{22} & b_{23} & b_{20} & b_{21} \\ b_{33} & b_{30} & b_{31} & b_{32} \end{pmatrix}$$

3. Multiplication par une matrice à coefficients égaux à 1, 2 ou 3, c'est-à-dire sous forme binaire 00000001, 00000010 ou 00000011 :

$$\begin{pmatrix} 1 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \times \begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{11} & b_{12} & b_{33} & b_{10} \\ b_{22} & b_{23} & b_{20} & b_{21} \\ b_{33} & b_{30} & b_{31} & b_{32} \end{pmatrix} = \begin{pmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{pmatrix}$$

Par exemple: $c_{12} = 1 \bullet b_{02} + 2 \bullet b_{13} + 3 \bullet b_{20} + 1 \bullet b_{31}$

4. Addition de la clé de tour : une clé différente à chaque tour est ajoutée bit par bit (XOR) au bloc.

$$\begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{11} & b_{12} & b_{33} & b_{10} \\ b_{22} & b_{23} & b_{20} & b_{21} \\ b_{33} & b_{30} & b_{31} & b_{32} \end{pmatrix} \oplus \begin{pmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{pmatrix}$$

Le tout est répété 10 fois pour obtenir le résultat final !

Les champs finis

• **Introduction :** les étapes 1 et 3 du tour de l'AES utilisent le champ fini F_{256} pour définir la permutation non linéaire et la permutation linéaire.

• **Exemples :** $F_2 = \{0, 1\}$ $F_4 = \{0, 1, i, 1+i\}$ avec la règle $i^2 = i+1$

$\oplus$	0	1
0	0	1
1	1	0

$\bullet$	0	1
0	0	0
1	0	1

$\oplus$	0	1	i	$1+i$
0	0	1	i	$1+i$
1	1	0	$1+i$	i
i	i	$1+i$	0	1
$1+i$	$1+i$	i	1	0

$\bullet$	0	1	i	$1+i$
0	0	0	0	0
1	0	1	i	$1+i$
i	0	i	$1+i$	1
$1+i$	0	$1+i$	1	i

• **Formellement :** $F_4 = F_2[X] / idl(X^2+X+1)$

Ce qui revient à $F_4 = \{0, 1, x, 1+x\}$, avec l'addition $\oplus$ usuelle de polynômes modulo 2, la multiplication $\bullet$ modulo 2 et modulo X^2+X+1.

• **Le champ à 2^8 éléments tel qu'utilisé par l'AES :**

$F_{256} = F_2[X] / idl(X^8+X^4+X^3+X+1)$.

Donc $F_{256} = \{0, 1, x, x+1, x^2, ..., x^7+x^6+x^5+x^4+x^3+x^2+x+1\}$

• **Exemples de calculs dans F_{256} :**

$(x^7+x^4+x^2+1) \oplus (x^6+x^2+x+1) = x^7+x^6+x^4+x$

$x \bullet x = x^2$ $x^4 \bullet x^4 = X^8 = x^4+x^3+x+1$ $x \bullet (x^7+x^3+x^2+1) = 1$

Ce dernier exemple signifie : $x^{-1} = (x^7+x^3+x^2+1)$

•**La permutation non linéaire S** de la première étape de l'AES est une application qui envoie x sur $x^{-1} \oplus a$ si $x \neq 0$, et sur a si x=0, avec un a fixé, x^{-1} étant l'inverse de x dans le groupe multiplicatif $(F_{256}\backslash\{0\},\bullet)$.

•**La permutation linéaire** de la troisième étape de l'AES utilise la multiplication de F_{256} en envoyant b sur $1 \bullet b$, $x \bullet b$ ou $(x+1) \bullet b$. Ici, 1, x et x+1 sont respectivement représentés dans la matrice par 1, 2 et 3.

$(F,+)$ est un groupe commutatif :	$(F_0,\bullet)$ est un groupe commutatif :
+ est interne et partout définie ;	• est interne et partout définie ;
$\forall x,y \in F : x + y = y + x$;	$\forall x,y \in F : x \bullet y = y \bullet x$;
$\forall x,y,z \in F : (x+y)+z = x+(y+z)$;	$\forall x,y,z \in F : (x \bullet y) \bullet z = x \bullet (y \bullet z)$;
$\exists\, 0 \in F, \forall x \in F : x+0 = x = 0+x$;	$\exists\, 1 \in F, \forall x \in F : x \bullet 1 = x = 1 \bullet x$;
• distribue + : $\forall x,y,z \in F : x \bullet (y + z) = x \bullet y + x \bullet z$.	
$\forall x \in F, \exists y \in F : x$	$\forall y \in F_0 : x \bullet y = 1$.

•**Galois et la théorie des champs finis :**
Évariste Galois (1811-1832) développe la théorie des champs finis. Ces derniers sont appelés *Galois Fields* en anglais, parfois noté GF(q), où q est le nombre d'éléments du champ.