Advanced Encryption Standard

Dans ce cours, nous nous intéresserons à l'AES (Advanced Encryption Standard) qui est le standard actuel de chiffrement pour la cryptographie à clé secrète. Nous aborderons les grandes lignes du chiffrement avec ce standard.

L'AES est un schéma de chiffrement symétrique qui prend en entrée un message de taille 128 bits et retourne un message de la même taille. La clé utilisée est également de taille 128 bits. Le chiffrement d'un message se fait en dix **tours**. Un tour est constitué des transformations SubBytes, ShiftRows, MixColumns et AddRoundKey. Ces transformations s'effectuent sur des **états**. Un état est un tableau de 4×4 octets (voir section 1).
L'opération AddRoundKey est un XOR bit à bit entre deux états. L'opération SubBytes est une transformation sur chacun des octets d'un état (voir section 3). L'opération MixColumns est une transformation sur les colonnes d'un état (voir section 4). Enfin, l'opération ShiftRows est une transformation sur les lignes d'un état (voir section 5).

Avant d'entamer le processus de chiffrement, une étape préliminaire d'expansion de clé est nécessaire. L'objectif de cette étape est de produire, à partir de la clé initiale, 11 sous-clés de taille 128 bits. Cette opération d'expansion ne sera pas abordée dans ce cours. Cet article (AES key schedule) présente l'essentiel de cette opération. Une fois le processus d'expansion de clé mis en place, on commence par transformer le message à chiffrer (de taille 128 bits) en un état.

1 Transformation d'un bloc en état

Il s'agit d'une transformation très simple. Supposons que notre message M de 128 bits est tel que $M = (m_0 m_1 m_2 \dots m_{127})$. On commence par découper M en blocs d'octets (8 bits).
Ce qui nous donne un message $B = b_0 b_1 \dots b_{15}$, où $b_i = (m_{8i} m_{8i+1} \dots m_{8i+7})$. On construit alors l'état correspondant comme suit :

b_0	b_4	b_8	b_{12}
b_1	b_5	b_9	b_{13}
b_2	b_6	b_{10}	b_{14}
b_3	b_7	b_{11}	b_{15}

Un état est donc un tableau de 16 cases, où chaque case est un octet, donc un entier compris entre 0 et 255. Cet entier peut donc être interprété comme le code ASCII d'une lettre.

Exercice 1 : Transformer le texte ci-dessous en un état en remplaçant chaque caractère par son code ASCII. Noter que tous les caractères doivent être pris en compte (apostrophe, espace, ...). Le texte : j'aime ce module

2 L'algorithme de chiffrement

Le chiffrement se fera avec le message et les sous-clés, transformés en états. Pour rappel, on dérive 11 sous-clés à partir de la clé initiale. L'algorithme de chiffrement est le suivant :

1. AddRoundKey : XOR bit à bit du message avec la première sous-clé, i.e. pour chaque case du message (transformé en état), on fait un XOR avec la case correspondante dans la sous-clé.
2. On répète 9 fois la séquence qui suit :
   a. SubBytes : transformation sur chacun des octets de l'état courant, i.e. transformation sur chaque case du tableau.
   b. ShiftRows : transformation sur les lignes de l'état courant.
   c. MixColumns : transformation sur les colonnes de l'état courant.
   d. AddRoundKey : XOR bit à bit de l'état courant avec la i-ème sous-clé.
3. Pour le dernier tour, on effectue la séquence de transformation suivante :
   a. SubBytes
   b. ShiftRows
   c. AddRoundKey : XOR bit à bit de l'état courant avec la dernière sous-clé.

Chacune des opérations de cet algorithme prend en entrée un état et retourne un autre état. Il est donc évident que le résultat du chiffrement est aussi un état.

Exercice 2 : Quel est le résultat de l'opération AddRoundKey avec l'état obtenu pour l'exercice 1 et la clé de tour suivante :

77	5	250	63
0	111	23	15
120	88	69	99
11	57	0	31

3 La transformation SubBytes

Cette opération effectue une substitution sur chacun des octets de l'état. Soit $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$ et $P(X) = X^8 + X^4 + X^3 + X + 1 \in \mathbb{F}_2[X]$. On a vu au semestre précédent que l'ensemble $\mathbb{F}_{2^8} = \mathbb{F}_2[X]/P(X)$ est un corps, car le polynôme $P(X)$ est irréductible (dans $\mathbb{F}_2[X]$). Soit a un octet, donc $0 \leqslant a < 255$. On peut ainsi identifier a à un élément de $\mathbb{F}_{2^8}$, à travers sa représentation binaire. Par conséquent, l'inverse x de a existe dans $\mathbb{F}_{2^8}$, si $a \neq 0$. On prendra $x = 0$, si $a = 0$.

Soit $(x_0 x_1 \dots x_7)$ la décomposition en base 2 de l'entier x (x_0 est le bit de poids faible), c'est donc un élément de $\mathbb{F}_2^8$. Considérons l'opération ci-dessous, où $y = (y_0 y_1 \dots y_7)$ est un élément de $\mathbb{F}_2^8$:

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

Le résultat de la transformation SubBytes appliquée à a est l'entier dont la décomposition en base 2 est donnée par le vecteur y. Notons que les additions et multiplications, lors du calcul de y, sont effectuées dans $\mathbb{F}_2$.

Soit α une racine de $P(X)$. Le corps $\mathbb{F}_{2^8}$ étant fini, le groupe multiplicatif $(\mathbb{F}_{2^8} \setminus \{0\}, \times)$ est cyclique. En d'autres termes, il admet (au moins) un générateur que nous désignerons par g dans la suite de ce cours. Par exemple, $g = \alpha + 1$ est un générateur de ce groupe. Ce qui signifie que tous les éléments non nuls de $\mathbb{F}_{2^8}$ peuvent s'exprimer sous la forme g^i, pour $i = 0$ à 254.

Exercice 3 : Construire un tableau **tab** qui associe à la représentation Python de chaque élément de $\mathbb{F}_{2^8}$, la puissance de g associée.
Par exemple, on a :

- $g^0 = 1$, donc tab[1] = 0.
- $g^1 = \alpha + 1$, donc tab[3] = 1, car $\alpha + 1$ est représenté par l'entier 3.
- $g^2 = \alpha^2 + 1$, donc tab[5] = 2, car $\alpha^2 + 1$ est représenté par l'entier 5.
- $g^3 = \alpha^3 + \alpha^2 + \alpha + 1$, donc tab[15] = 3, car $\alpha^3 + \alpha^2 + \alpha + 1$ est représenté par l'entier 15.

Ne pas oublier que α étant une racine de $P(X)$, on a $P(\alpha) = 0$. Donc, $\alpha^8 = \alpha^4 + \alpha^3 + \alpha + 1$.

Si on cherche l'inverse de 5 (par exemple), il faut procéder de la façon suivante. 5 est l'entier représentant l'élément $\alpha^2 + 1$ de $\mathbb{F}_{2^8}$. On a tab[5]=2, donc $\alpha^2 + 1$ peut s'exprimer comme g^2 dans $\mathbb{F}_{2^8}$. Comme mentionné plus haut, g est un générateur de $(\mathbb{F}_{2^8} \setminus \{0\}, \times)$, donc $g^{255} = 1$. Ainsi $g^{255} = g^{253} \times g^2$, donc l'inverse de g^2 est g^{253}. Il faut donc maintenant savoir quel est l'entier qui en Python représente g^{253}.

Exercice 4 : Précédemment, vous avez créé une table **tab**, telle que tab[g^i] = i. Vous pouvez en même temps créer la table **invtab**, telle que invtab[i] = g^i. Avec cette table, on doit obtenir que g^{253} est représenté par l'entier invtab[253]. L'inverse de 5 est donc donné par l'entier invtab[255-tab[5]], pourquoi ?

Exercice 5 : Appliquer l'opération SubBytes à l'état suivant :

5	71	100	88
19	211	26	0
47	4	62	33
55	37	41	1

4 La transformation MixColumns

Cette transformation est une multiplication matrice-vecteur, où chaque colonne de l'état est multipliée par la matrice circulante suivante :

$$\begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}$$

Remarques :

- Les entiers de la matrice ci-dessus sont les représentations des éléments de $\mathbb{F}_{2^8}$. Donc, 2 est la représentation en Python de l'élément α, 1 représente l'élément 1 et 3 l'élément $\alpha + 1$.
- Lors de ces multiplications matrice-vecteur, les additions et multiplications sont effectuées dans le corps $\mathbb{F}_{2^8}$. En outre, pour la multiplication vous devez vous servir de votre table qui vous dit à quelle puissance de g correspond un élément de $\mathbb{F}_{2^8}$.

Exemple 1. Si on veut multiplier 15 par 5 dans $\mathbb{F}_{2^8}$, on procède de la façon suivante. Avec la table tab on sait que 15 représente g^3 (car tab[15] = 3) et 5 représente g^2 (car tab[5] = 2). Donc, dans $\mathbb{F}_{2^8}$, le résultat de 15 multiplié par 5 revient à multiplier g^3 par g^2, ce qui donne g^5. Grâce à la table invtab, on a invtab[5] = 51, donc g^5 est représenté par l'entier

51. Le résultat de 5 multiplié par 15 est donc 51, c'est à dire : invtab[(tab[5]+tab[15]) % 255]. Comprenez-vous pourquoi on réduit modulo 255 ? (Indice : $g^{255} = 1$ dans $\mathbb{F}_{2^8}$)

Exemple 2. L'opération ci-dessous calcule le première colonne de la transformation MixColumns appliquée à l'état de l'exercice précédent.

$$\begin{pmatrix} 39 \\ 101 \\ 17 \\ 93 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \times \begin{pmatrix} 5 \\ 19 \\ 47 \\ 55 \end{pmatrix}$$

On peut observer que cette multiplication matricielle transforme un vecteur de 4 octets en un autre vecteur de 4 octets.

Exercice 6 : Terminer l'exemple précédent, en appliquant la transformation MixColumns sur les autres colonnes de l'état de l'exercice 5.

5 La transformation ShiftRows

Cette transformation consiste en de simples rotations circulaires vers la gauche sur les lignes d'un état. Plus précisément, à la ligne i de l'état, on applique une rotation circulaire de $(i-1)$ positions vers la gauche, avec $1 \leqslant i \leqslant 4$.

Par exemple, si (a, b, c, d) est la deuxième ligne d'un état, alors cette opération transforme cette ligne en (b, c, d, a).

Exercice 7 : Appliquer la transformation ShiftRows à l'état de l'exercice 5.