Advanced Encryption Standard

Ilyas Bambrik

Table des matières

Opérateur modulo, Anneau et Finite Field (Galois Field)

Définition : Définition de l'opérateur modulo

Soit a,r,m des nombres entiers avec m>0

$a \equiv r \bmod m$

Si (a-r) est divisible par m

- r est le reste de la division entière
- m est le modulo
- ≡ signe d'équivalence

Exemples :

$17 \equiv 2 \bmod 5$

$-7 \equiv 3 \bmod 5$

$25 \equiv 7 \bmod 9$

Le reste de la division entière n'est pas unique :

$17 \equiv 2 \bmod 5 = 7 \bmod 5 = 12 \bmod 5 = -3 \bmod 5$

Par convention, le chiffre utilisé pour le reste est le plus petit entier positif

L'opération de division est particulière avec le modulo :

$b/a \equiv b \times a^{-1} \bmod m$

Où a^{-1} est l'inverse de a :

$a \times a^{-1} \equiv 1 \bmod m$

En utilisant le modulo arithmétique de m, on obtient une structure dite Anneau N_m (N_m= {0, 1, 2, 3,...., m-2, m-1}). Cette structure possède les propriétés suivantes :
Pour tout élément a, b, c ϵ N_m :

- **Clôture (Closure)** : (a + b) mod m et (a x b) mod m appartiennent à N_m
- L'adition et la multiplication sont **Associatives** : (a + b) +c = a + (b + c) et (a x b) x c mod m = a x (b x c) mod m
- **Loi distributive** a x (b + c) mod m= ((a x b) + (a x c)) mod m
- 0 est l'élément neutre pour l'opération d'adition a + 0 ≡ a mod m
- 1 est l'élément neutre pour l'opération de multiplication a x 1 ≡ a mod m
- Pour chaque élément a, un élément inverse –a existe pour l'opération d'adition: a + (-a) ≡ 0 mod m (-a ϵ N_m)
- **Pas touts les éléments ne possèdent un** inverse pour la multiplication : a x a^{-1} ≡ 1 mod m (a^{-1} ϵ N_m)

L'élément inverse de **a** existe pour modulo m seulement si gdc(a, m)= 1 (a et m sont premiers entre eux)
Par exemple :
Soit l'anneau N_9 ={0,1,2,3,....,8}
2^{-1} ≡ 5 mod 9 , 4^{-1} ≡ 7 mod 9 , par contre il n'existe pas d'inverse de 3.
2+7 ≡ 0 mod 9, 4+5 ≡ 0 mod 9, etc

Galois Field (Finite Field)

Brièvement, Finite Field (FF)possède les mêmes propriétés d'un anneau avec deux particularités :

- FF doit etre composé de P^m éléments avec P nombre primaire et m un entier >0 (FF(P^m)= {0,1,2,3,4,...,P^m-1}
- Contrairement à un Anneau, chaque élément (sauf 0) possède un élément inverse pour la multiplication (a x a^{-1} = 1 mod P^m) [car pour chaque élément A_i inférieure à P, *GDC(A_i,P)=1*]

Si m= 1, FF(P) est appelé Prime Field

Si m>1,FF(P^m) est appelé Extended Finite Field

La différence entre FF(P) et FF(P^m) c'est que les éléments de FF(P) sont des entiers alors que les éléments des FF(P^m) sont des polynômes.

Pour chaque a ϵ $FF(P^m)$, a est exprimé comme suite :

$a_{m-1} x^{m-1} + a_{m-2} x^{m-2} + a_{m-3} x^{m-3} + ...+ a_i x^i +.... + a_1 x^1 + a_0$

avec a_i ϵ **FF(P)**

Fondamental

En informatique, nous sommes particulièrement intéressés $FF(2^8)$ (P=2 , m=8)

Exemple

Dans $FF(2^3)$, les éléments sont représentés sous la forme : $a_2 x^2 + a_1 x + a_0$ (avec a appartenant à FF (2))

$FF(2^3)$={ 0, 1 , x, x+1,x^2,x^2+1,x^2+x,x^2+x+1}

1. Opération arithmétiques sur FF(Pm)

Opérations arithmétiques sur $FF(P^m)$:

Soit $A(x)$, $B(x) \in FF(P^m)$

$A(x)= a_{m-1} x^{m-1} + a_{m-2} x^{m-2} + a_{m-3} x^{m-3} + ...+ a_i x^i +.... + a_1 x^1 + a_0$

$B(x)= b_{m-1} x^{m-1} + b_{m-2} x^{m-2} + b_{m-3} x^{m-3} + ...+ b_i x^i +.... + b_1 x^1 + b_0$

$$C(x) = A(x) + B(x) = \sum_{i=0}^{m} c_i x^i$$

$c_i \equiv a_i + b_i \bmod P$

$$C(x) = A(x) - B(x) = \sum_{i=0}^{m} c_i x^i$$

$c_i \equiv a_i - b_i \bmod P$

Exemple :

$A(x)= x^2 + x + 1$

$B(x) = x^2 + 1$

$C(x)= A(x) + B(x) = ((1+1) \bmod 2)\, x^2 + (1 \bmod 2)\, x + ((1 + 1) \bmod 2)$

$C(x)= x$

Pour la multiplication:

$C(x)=A(x) \,.\, B(x) \bmod PM(x)$

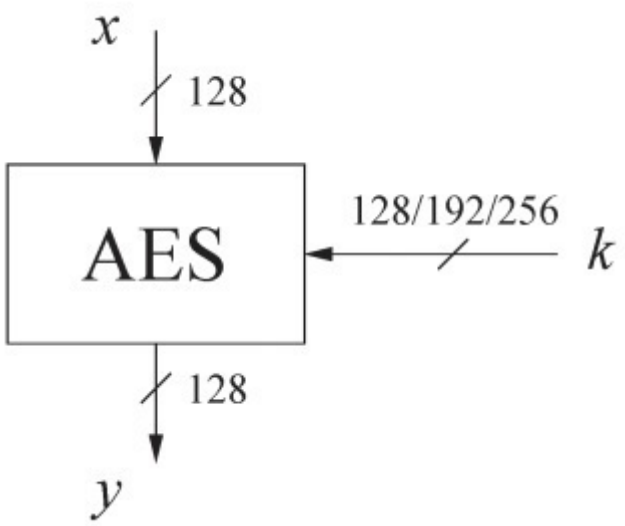
Où $PM(x)$ est appelé le polynôme modulo de $FF(P^m)$. Pour chaque $FF(P^m)$, il existe plusieurs polynômes modulos (la propriété de ces polynômes c'est qu'ils ne peuvent pas être factorisés).

Par exemple pour $FF(2^3)$, $P(x) = x^3 + x + 1$

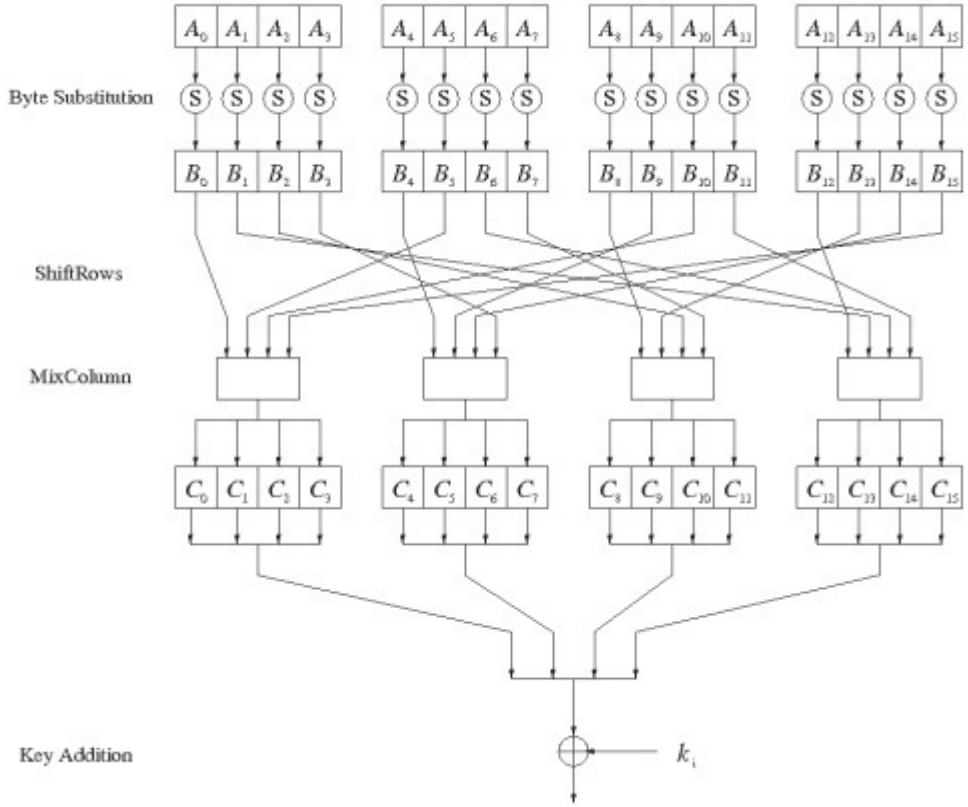
II

AES Advanced Encryption Standard (Rijndael)

- Proposé pour la compétition organisée par NIST (US National Institute of Standards and Technology) afin de définir un nouvel algorithme de cryptographie pour remplacer DES.
- Standardisé en 2000.
- Algorithme de chiffrement par bloque. Cet algorithme est utilisable avec des tailles de bloques de 128 bits et clé de taille 128bits (10 rounds), 192bits (12 rounds) ou 256 bits (14 rounds).
- Contrairement à DES, AES opère sur des octets et non pas des bites. Ce-ci augmente la rapidité de chiffrement.



1. Algorithme AES



Chaque round de AES se compose de trois étapes :

- *SubBytes :* Substitue la valeur d'un octet par une autre valeur (*Confusion*) ;
- *ShiftRows :* Effectue une permutation sur le bloque des octets (*Diffusion*);
- *MixColumns :* Effectue une multiplication matricielle (*Diffusion*) ;
- *AddRoundKey* (XOR avec la clé du round courant)

Le dernier round de AES ne comporte pas *MixColumns* ;

- L'implémentation la plus utilisée actuellement est celle de AES-128 bits (tous les navigateurs web et serveurs https utilisent AES).
- La première étape de AES consiste à organiser le bloque de 128 bits (16 octets) en entré dans une matrice de 4 x 4 (chaque cellule est un octet).

A_0	A_4	A_8	A_{12}
A_1	A_5	A_9	A_{13}
A_2	A_6	A_{10}	A_{14}
A_3	A_7	A_{11}	A_{15}

1.1. SubBytes

- Contrairement à DES, la table de substitution S est la même pour chaque octet (DES définie 8 S-Box).
- La fonction de cette couche est simplement de substituer la valeur d'un octet par une autre.
- Pour faire ce-ci, la valeur de l'octet est représentée en hexadécimale. Par exemple si la valeur de l'octet = 83, la représentation hexadécimale= 53 ;
- La substitution de 53 (83 décimale) consiste simplement à prendre la valeur à la ligne 5, colonne 3. Ce-ci donne *ED* (237) ;

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	3	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

1.2. ShiftRow

- Comme expliqué précédemment , le texte claire est représenté avec une matrice de 4x4 octets (de A_0 jusqu'à A_{15}) ;
- Après l'application du SubByte, le résultat sera une matrice 4 x 4 (de B_0 jusqu'à B_{15}) ;
- Par la suite, chaque ligne i de la matrice est décalée à gauche de i positions ;

Input matrix

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}

Output matrix

B_0	B_4	B_8	B_{12}	no shift
B_5	B_9	B_{13}	B_1	← one position left shift
B_{10}	B_{14}	B_2	B_6	← two positions left shift
B_{15}	B_3	B_7	B_{11}	← three positions left shift

1.3. MixColumns

Après la phase Shiftrow, chaque colonne est multipliée avec une matrice fixe :

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

La multiplication d'un octet avec un autre est faite en $FF(2^8)$ (avec modulo polynôme primaire $P(x) = x^8 + x^4 + x^3 + x + 1$)

Fondamental : Propriété de Diffusion dans MixColumns

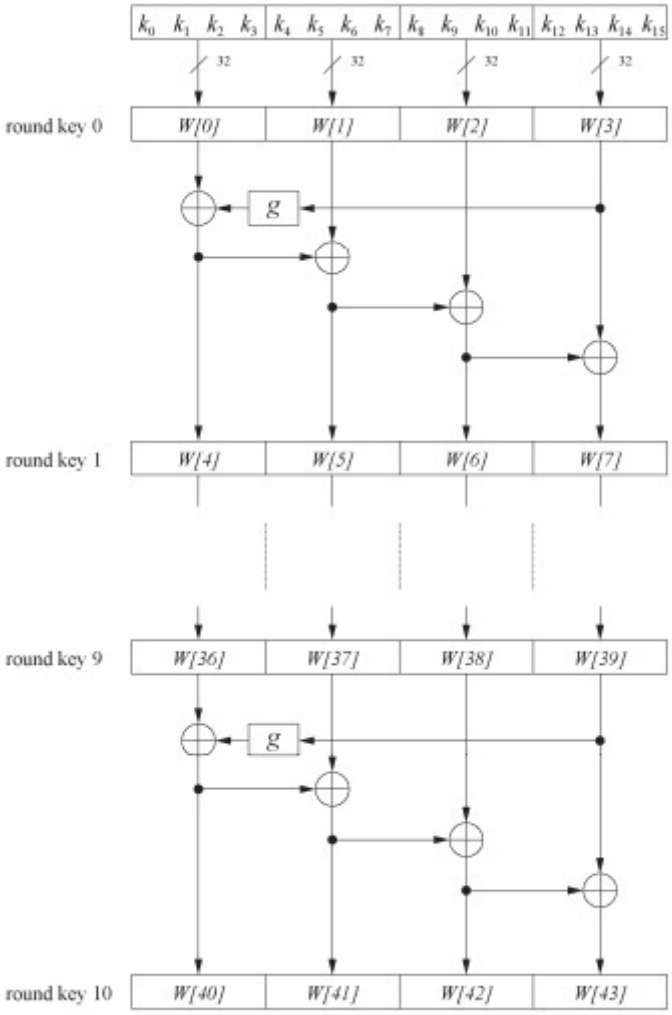
Grâce à la phase MixColumns , si un bit de l'un des octets B_0, B_5, B_{10} ou B_{15} , les quatre octets en sorties C_0, C_1, C_2 et C_3 seront tous affectés.

2. Génération des clés

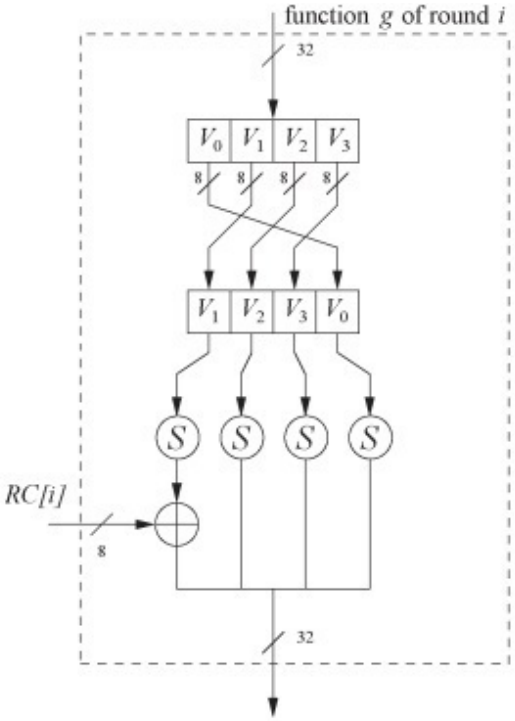
- AES utilise 11 clés dérivées de la clé originale (la 1ere clé est utilisée avant de commencer les 10 itérations, le reste des clés sont utilisée pendant les rounds);
- Pour commencer, la clé K_0 est divisée en 4 bloque [W0, W1, W2, W3](chaque bloque contient

4 octets).

- La première clé est égale à la clé initiale (aucun changement sur les bloques de la clé) ;
- La deuxième clé [W4,W5,W6,W7] est dérivée tel que :
  W4= W0 Xor g(W3) , W5= W1 Xor W4 , W6= W2 Xor W5, W6= W2 Xor W5, W7= W3 Xor W6
- La fonction g comporte un LeftShift d'une position de chaque octet et un Xor avec le numéro du round (RC[i]) pour l'octet le plus fort seulement ;



$$RC[1] = x^0 = (00000001)_2$$
$$RC[2] = x^1 = (00000010)_2$$
$$RC[3] = x^2 = (00000100)_2$$
$$\ldots$$
$$RC[10] = x^9 = (00110110)_2$$



3. Déchiffrement

- Contrairement à DES, le texte chiffré avec AES ne peut pas être déchiffré par la même procédure.
- L'ordre des étapes de AES est inversé lors du déchiffrement et l'ordre des clés est inversé :
  - AddKey
  - InvrseMixColomns
  - InvrseShiftRaw
  - InvrseSubByte
- Particulièrement les valeurs de la matrice MixColumns sont remplacée par l'inverse dans FF(2^8).
- La procédure ShiftRaw est remplacée par des RightShift lors du déchiffrement au lieu de LeftShift du chiffrement ;
- Pour rappel, le dérnier round de chiffrement ne contient pas MixColumns. Lors du déchiffrement, le premier round de chiffrement ne contient pas MixColumns ;

$$\begin{pmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{pmatrix} = \begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix} \cdot \begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix}$$

Input matrix

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}

Output matrix

B_0	B_4	B_8	B_{12}	no shift
B_{13}	B_1	B_5	B_9	→ one position right shift
B_{10}	B_{14}	B_2	B_6	→ two positions right shift
B_7	B_{11}	B_{15}	B_3	→ three positions right shift