Advanced Encryption Standard AES :

- plusieurs longueur de clés possible 128, 192 ou 256 bits
- le nombre de cycle "ronde" varie en fonction de la longueur de la clé et des blocs de 10 à 14.
- la structure générale ne comprend qu'une serie de transformation / permutation / selection.
- il est beaucoup plus performant que le DES.

A chaque ronde quatre transformation sont appliquées :

* substitution d'octet dans le tableau d'état (S-box)
* Décalage de rangée
* Déplacement de colonne dans le tableau (sauf à la dernière ronde)
* addition d'une clé à chaque ronde.

+ Table d'état du texte et des clés

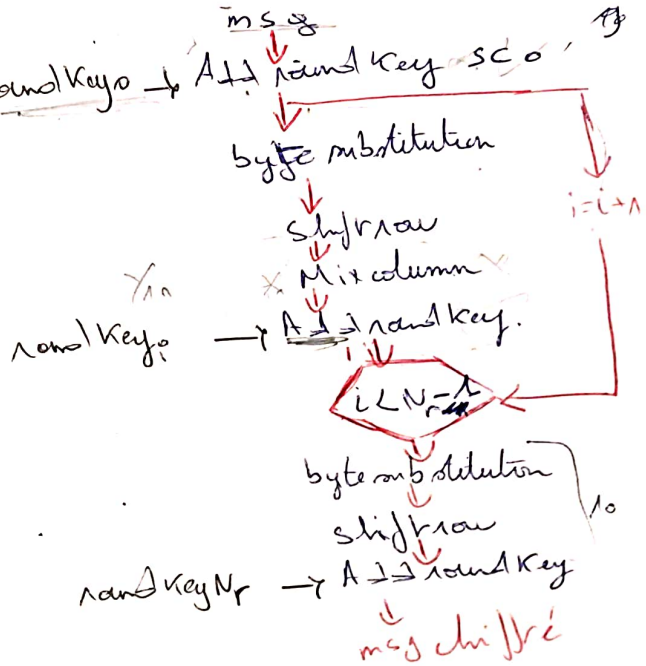
Le msg et la clé sont conservé sous forme de table représenté respectivement comme sur la figure le nbre de colonne dépend de la taille du texte et de la clé. $N_b = L_{bloc}/32$; $N_k = L_{clé}/32$. une colonne du tableau correspond à un mot de 32 bits. ainsi chaque petit bloc représente 8 bits ou 1 octet.

a_{00}	a_{01}	a_{02}	a_{03}	a_{10}	a_{11}	a_{12}	a_{13}	a_{21}	...

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}				

128 bits
192 bits
256 bits

Chaque case contient un octet (2 chiffres)
Hexadécimaux 1er représente la ligne
2ème représente la colonne pour la substitution dans la S-Box



Sub substitution

Les octets sont transformé en appliquant une S-box inversible afin de permettre le déchiffrement.
C'est une table à 16 lignes et 16 colonnes on recherche dans la table et on prend le contenu de la case.

(2)

Shift row

cette étape augmente la diffusion dans la norme les décalages ne sont pas tjrs identique.

	C1	C2	C3
NB=4	1	2	3
NB=6	1	2	3
NB=8	1	3	4

- la ligne 0 n'est jamais décalé
- " " 1 est décalé de C1.
- " " 2 " " " C2.
- " " 3 " " " C3.

Exemple

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}

⇒

a_{00}	a_{01}	a_{02}	a_{03}
a_{11}	a_{12}	a_{13}	a_{10}
a_{22}	a_{23}	a_{20}	a_{21}
a_{33}	a_{30}	a_{31}	a_{32}

Mix column

Une différence sur 1 byte d'entrée se propage sur les 4 bytes de sortie, on a donc encore une étape de diffusion. La matrice utilisé est définie par Rijndael elle contiendra toujours ces valeurs.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} \\ y_{21} & y_{22} & y_{23} & y_{24} \\ y_{31} & y_{32} & y_{33} & y_{34} \\ y_{41} & y_{42} & y_{43} & y_{44} \end{bmatrix}$$

AddRoundKey

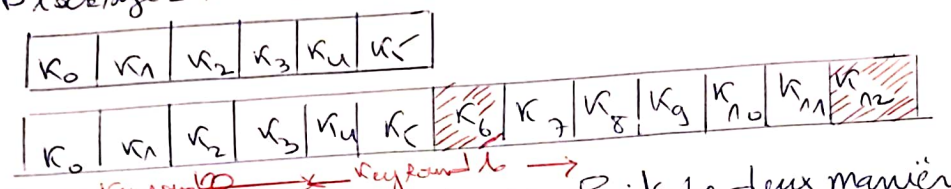
C'est un simple ⊕ des clé il s'agit d'addition des sous clé aux sous blocs correspondant.

Calcul de la clé

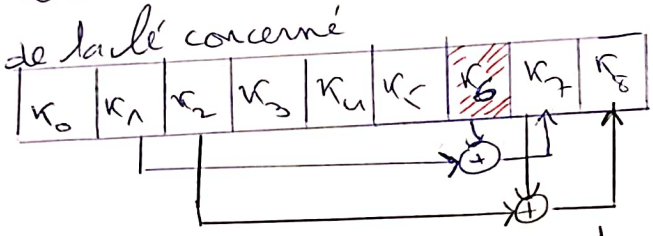
Après avoir subit une extension (key expansion) la clé sera composer sous clés (clés de ronde).

key size = 192 bits ($N_k = 6$)

Block size = 128 bits ($N_b = 4$)



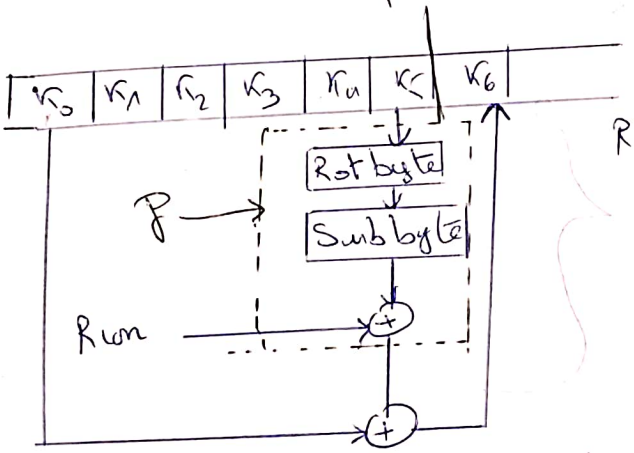
le calcul de l'expansion de la clé se fait de deux manière selon les sous blocs de la clé concerné



$K_i = K_{i-N_k} \oplus K_{i-1}$

$K_1 \oplus K_6$

expansion de la clé avec bloc "commun"



RotByte(abcd) = (bcda)

Rcon[i] = (RC[i], 00, 00, 00)

SubByte utilise la Sbox

$K_i = K_{i-N_k} \oplus f(K_{i-1})$

expansion de la clé avec les bloc "multiple de N_k"

l'ajout de Rcon[x] donne comme resultat un ⊕ sur les bits les plus significatifs. le tableau utilisé pour donner les valeurs de Rcon

i	1	2	3	4	5	6	7	8	9	...
RC[i]	01	02	04	08	10	20	40	80	1B	...

Nombre de rounds

Longueur de la clé

Longueur du bloc	128 bits	192 bits	256 bits
128 bits	10	12	14
192 bits	12	12	14
256 bits	14	14	14