MINES Saint-Étienne

Une école de l'IMT

Rapport Projet VHDL

CHIFFREMENT AES

Guillaume Reymond

Anna Christiane Kolandjian

Chiffrement AES

RAPPORT VHDL

Kolandjian Anna Christiane

À MON PROFESSEUR GUILLAUME REYMOND.

Avec l'espoir que ce projet VHDL répondra à toutes les exigences.

CONTENTS

LISTE DE FIGURES

1 Introduction

1.1 Introduction de l'AES

Advanced Encryption Standard (AES), également appelé Rijndael (son nom d'origine), est une spécification pour le cryptage électronique des données établies par les États-Unis. Institut national des normes et de la technologie (NIST) en 2001. L'AES peut être programmé en logiciel ou construit avec du matériel pur.
Pour AES, le NIST a sélectionné trois membres de la famille Rijndael, chacun avec une taille de bloc de 128 bits, mais trois longueurs de clé différentes : 128, 192 et 256 bits.
L'AES a été adopté par le gouvernement américain et est maintenant utilisé dans le monde entier. Il remplace le Data Encryption Standard (DES), qui a été publié en 1977. L'algorithme décrit par AES est un algorithme à clé symétrique, ce qui signifie que la même clé est utilisée pour le cryptage et le décryptage des données.

Ce projet représente une méthode pour intégrer l'algorithme de chiffrement AES et le logiciel Module Sim est utilisé pour la simulation. D'où, Putty et Xming sont utilisés dans les systèmes d'exploitation Windows 10. De plus, nous avons utilisé la solution FTP gratuite, FileZilla, pour accéder facilement au serveur.

1.2 Evolution de l'AES

Le DES est désormais considéré comme insécurisé pour de nombreuses applications. Cela est principalement dû au fait que la taille de la clé 56 bits est trop petite ; les clés DES ont été cassées en moins de 24 heures. Il existe également des résultats analytiques qui démontrent des faiblesses théoriques dans le chiffrement, bien qu'ils soient impossibles à monter dans la pratique. On pense que l'algorithme est pratiquement sécurisé sous la forme de Triple DES, bien qu'il existe des attaques théoriques. Ces dernières années, le chiffrement a été remplacé par le standard de chiffrement avancé (AES).

Au cours de l'été 2001, AES a remplacé le DES vieillissant en tant que Norme de cryptage de traitement (FIPS). Le DES atteignait la fin de sa vie, car le déchiffrement de son chiffre est considéré comme plus maniable sur le matériel informatique actuel. L'algorithme AES sera utilisé pour de nombreuses applications au sein du gouvernement et dans le secteur privé. Briser un texte chiffré (AES) en essayant toutes les clés possibles est actuellement impossible à calculer avec l'évolution de la technologie.

Pour chiffrer un texte, il suffit d'avoir un message avec une clé. Le chiffrement AES transforme le message à un texte incompréhensible, chiffré.

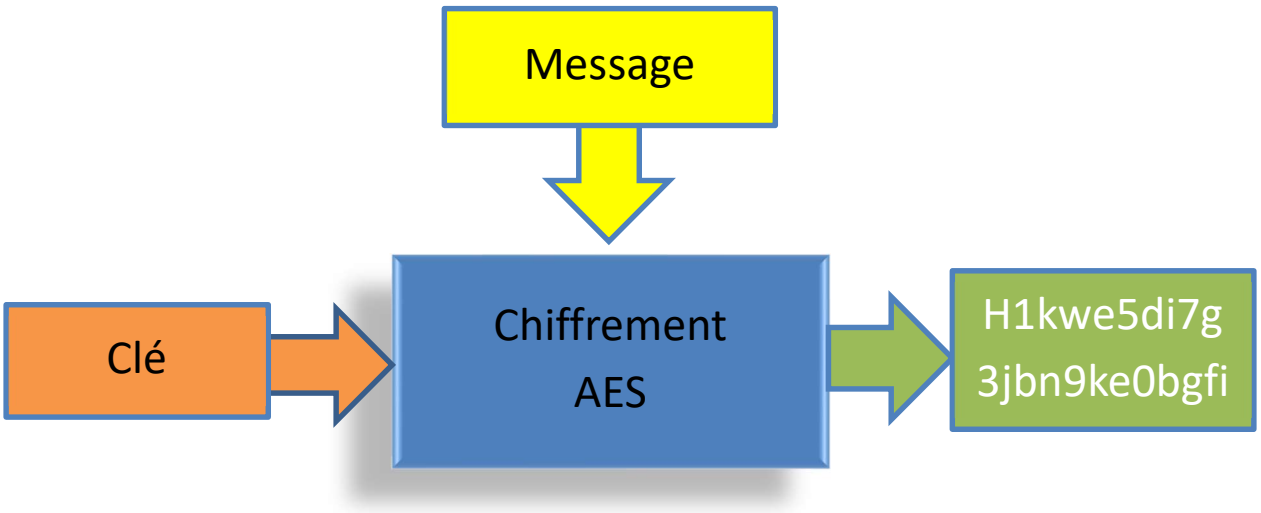

Figure 1. Bloc du chiffrement AES

2 Notations et conventions

2.1 Entrées et sorties

L'entrée et la sortie de l'algorithme AES se composent de séquences de 128 bits. Ces séquences sont appelées blocs et le nombre de bits qu'elles contiennent est leur longueur. La clé de chiffrement pour l'algorithme AES est une séquence de 128 bits.
Les bits de ces séquences sont numérotés à partir de zéro et se terminant à un nombre inférieur de 1 à la longueur de la séquence, également appelée longueur de bloc ou longueur de clé.
La lettre « i » ou « j » attachée à un bit est connu comme son index et sera dans l'une des plages $0 < i < 128$.

2.2 Octets

L'unité de base du traitement dans l'algorithme AES est un octet, qui est une séquence de huit bits traités comme une seule entité. Les séquences de bits d'entrée, de sortie et de clé de chiffrement décrites dans les sections 1.2 et 2.1 sont traitées comme des tableaux d'octets qui sont formés en divisant ces séquences en groupes de huit bits contigus pour former des tableaux d'octets. Pour une entrée, sortie ou clé de chiffrement notée a, les octets du tableau résultant sont référencés à l'aide de l'une des deux formes, a_n ou a[n], où n sera dans une plage qui dépend de la clé longueur. Pour une longueur de clé de 128 bits, n se situe dans la plage $0 \leq n < 16$.
La plupart des valeurs d'octets dans l'algorithme AES sont présentées comme la concaténation des valeurs binaires individuelles, (0 ou 1), entre accolades dans l'ordre {b7, b6, b5, b4, b3, b2, b1, b0}. On aura une exception dans le code de l'opération « Mix Column » qui sera expliqué dans la section 3.4.
Ces octets sont interprétés comme des éléments de champ fini utilisant une représentation polynomiale : $b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0x^0 = \sum_{i=0}^{7} b_i x^i$

Par exemple, {01100011} identifie l'élément de champ fini spécifique $x^6 + x^5 + x + 1$. C'est également pratique pour désigner les valeurs d'octets en utilisant la notation hexadécimale avec chacun des deux des groupes de quatre bits étant désignés par un seul caractère hexadécimal.

Bit pattern	Charactère
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

Bit pattern	Charactère
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Figure 2. Représentation hexadécimale des bits patterns

2.3 Tableau d'octets

Les tableaux d'octets sont représentés sous la forme $a_0a_1a_2 \cdots a_{15}$. Les octets et l'ordre de bit dans les octets est dérivé de la séquence d'entrée de 128 bits, $input_0input_1input_2 \cdots input_{126}input_{127}$ tel que a_0= {$input_0$, $input_1$, ⋯, $input_7$}, a_1= {$input_8$, $input_9$, ⋯, $input_{15}$} avec le motif se poursuivant jusqu'à a_{15}= {$input_{120}$, $input_{121}$, ⋯, $input_{127}$}. En général, a_n = {$input_{8n}$, $input_{8n+1}$, ⋯, $input_{8n+7}$}, où n varie de 0 à 15. Un exemple de désignation et de numérotation d'octets dans les octets pour une séquence d'entrée donnée est présenté par la figure 3.

Entrée de séquence de bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	…
Numéro octet	0								1								2				
Numéro de bit dans l'octet	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	…

Figure 3. Indices des Bits et des octets

Comme on a déjà précisé précédemment, l'opération mix column aura une exception sur les numéros de bit dans les octets, qui seront représentés de la façon suivante : 0 1 2 3 4 5 6 7 et non pas 7 6 5 4 3 2 1 0.

2.4 Le State

A l'intérieur, les opérations de l'algorithme AES sont effectuées sur un plan bidimensionnel, tableau d'octets appelé l'État. L'État se compose de quatre lignes d'octets. Chaque ligne contient un Nb nombres d'octets, où Nb est la longueur de bloc divisée par 32. Dans le Tableau d'état, qui est désigné par le symbole S, chaque octet individuel a deux indices. L'indice du premier octet est le numéro de ligne r, qui se situe dans la plage $0 \le r \le 3$ et celle du deuxième octet est le numéro de colonne c, qui se situe dans la plage $0 \le c \le Nb - 1$. Cette indexation permet un octet individuel de l'État, appelé $S_{r,c}$ ou S [r, c]. Pour l'AES Nb = 4, qui signifie que $0 \le c \le 3$. Au début du chiffrement, l'entrée, qui est le tableau d'octets symbolisé par $in_0in_1 \cdots in_{15}$ est copié dans le tableau State. Cette activité est illustrée dans la figure 3. Les opérations de chiffrement sont effectuées dans le tableau d'état. Une fois la manipulation du tableau d'état terminée, sa valeur finale est copiée à la sortie, qui est un tableau d'octets symbolisé par $out_0out_1 \cdots out_{15}$.
La plupart du temps, j'ai utilisé le std_logic_vector au lieu du State dans mes opérations.

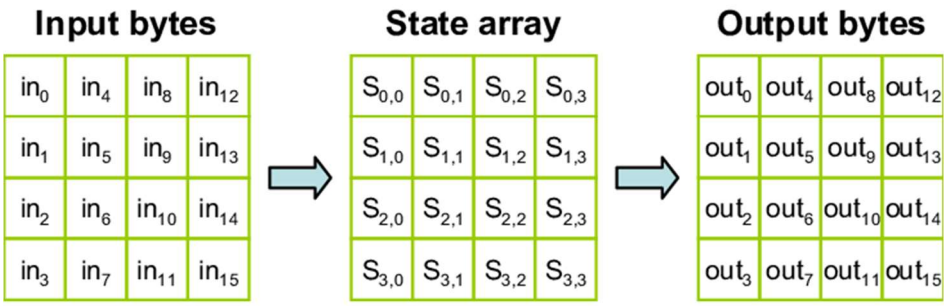

Figure 4. Représentation de la matrice d'état de l'AES

Le State comme tableau de colonnes

Les quatre octets dans chaque colonne de l'état forment des mots de 32 bits, où le numéro «r» qui est la ligne fournit un indice pour les quatre octets de chaque mot. Par conséquent, l'État peut être interprété comme un tableau unidimensionnel de mots de 32 bits, symbolisé par w_0 ... w_3. Le numéro de colonne c fournit un indice dans ce tableau d'état linéaire. Considérant le State (l'État) représenté sur la figure , l'État peut être considéré comme un tableau de quatre mots tel que :

$$w_0 = S_{0,0}\, S_{1,0}\, S_{2,0}\, S_{3,0},$$
$$w_1 = S_{0,1}\, S_{1,1}\, S_{2,1}\, S_{3,1},$$
$$w_2 = S_{0,2}\, S_{1,2}\, S_{2,2}\, S_{3,2},$$
et
$$w_3 = S_{0,3}\, S_{1,3}\, S_{2,3}\, S_{3,3}.$$

2.5 Formations mathématiques

Chaque octet de l'algorithme AES est interprété comme un élément de champ fini à l'aide de la notation introduite dans les sections précédentes. Tous les éléments de champ fini peuvent être additionnés et multipliés. Cependant, ces opérations diffèrent de celles utilisées pour les nombres et leur utilisation nécessite une investigation.

Addition

L'addition de deux éléments dans un champ fini est obtenue en « ajoutant » les coefficients pour les puissances correspondantes dans les polynômes pour les deux éléments. L'addition est effectuée en utilisant l'opération XOR, qui est indiquée par le symbole opérateur $\oplus$. Une telle addition est effectuée modulo-2.

$$1 \oplus 1 = 0,$$
$$1 \oplus 0 = 1,$$
$$0 \oplus 1 = 1$$
et
$$0 \oplus 0 = 0.$$

Multiplication

Dans la représentation polynomiale, la multiplication dans le champ de Galois GF (2^8) (notée par •) correspond à la multiplication des polynômes modulo un polynôme irréductible de degré 8. Un polynôme est irréductible si ses seuls diviseurs sont un et lui-même.

Pour l'algorithme AES, ce polynôme irréductible est donné par l'équation : m (x) = $x^8 + x^4 + x^3 + x + 1$
En pratique :

$$A_i = (C2)_{hex} = (11000010)_2 \rightarrow x^7 + x^6 + x$$

On cherche Bi' tel que : $B_i' = A_i^{-1}$, c-à-d : $A_i * B_i' = 1 \bmod x^8 + x^4 + x^3 + x + 1$

$$B_i' = x^5 + x^3 + x^2 + x + 1 \rightarrow (00101111)_2$$

La multiplication dans le GF va être utilisé dans le Mix Column.

3 Algorithme AES

3.1 Description de l'algorithme AES

L'organigramme ci-dessous est générique pour les spécifications AES. Il se compose d'un certain nombre de différentes transformations appliquées consécutivement sur les bits du bloc de données, dans un nombre d'itérations, appelées rondes. Le nombre de tours dépend de la longueur de la clé utilisée pour le processus de cryptage.

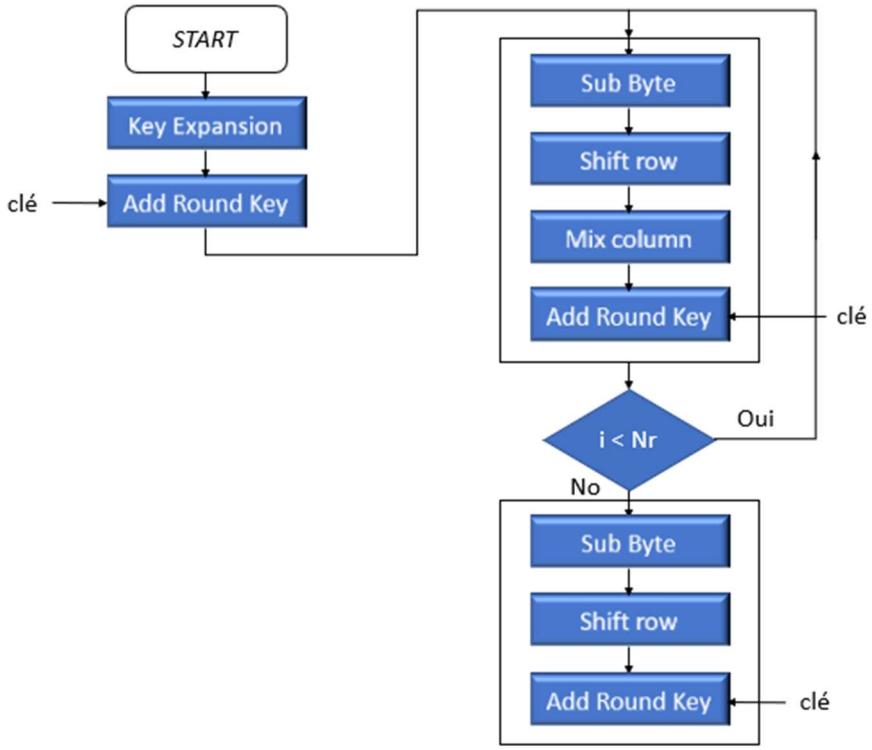

Figure 5. Le processus du chiffrement AES

3.2 Etape Sub-Bytes

3.2.1 Définition

La transformation de substitution d'octets Sub-Bytes est une substitution non linéaire d'octets qui opèrent indépendamment sur chaque octet de l'État à l'aide d'une table de substitution (S-Box) présentée dans la figure 6. Cette S-Box qui est inversible, est construit en composant deux transformations : Multiplication et Addition dans le GF (2^8).
Sous forme matricielle, la transformation des éléments de la S-box peut être exprimée comme :