1ère année du cycle d'ingénieur

[MSEI]

Module : Composants Programmables & VHDL

Rapport du Mini-Projet

Advanced Encryption Standard

Réalisé par :

- **Wahib Houda**
- **Salim Ahlam**
- **Esabbar Nouhaila**

Encadré par : Pr. S.EL MOUMNI

Introduction générale

La cryptographie est une discipline essentielle dans le domaine de la sécurité informatique, permettant de protéger l'information contre les accès non autorisés et les modifications malveillantes. Elle englobe un ensemble de techniques et d'algorithmes visant à assurer la confidentialité, l'intégrité et l'authenticité des données. Parmi les algorithmes de cryptographie symétrique les plus utilisés, l'Advanced Encryption Standard (AES) occupe une place prépondérante en raison de sa robustesse et de son efficacité.

L'AES, établi par le National Institute of Standards and Technology (NIST) en 2001, est un standard de chiffrement largement adopté pour sécuriser les communications et les données sensibles. Il repose sur une architecture de réseau de substitution-permutation et supporte des clés de 128, 192 ou 256 bits, offrant ainsi une sécurité adaptée à divers besoins.

L'objectif de ce projet est de concevoir et de mettre en œuvre l'algorithme AES en utilisant le langage de description matériel VHDL (VHSIC Hardware Description Language). VHDL est un langage puissant et flexible utilisé pour modéliser et simuler des systèmes électroniques numériques. En exploitant VHDL, nous pouvons non seulement décrire le comportement de l'AES de manière précise, mais aussi synthétiser le circuit correspondant pour une implémentation matérielle, telle qu'un FPGA (Field-Programmable Gate Array).

Ce projet se déroulera en 2 chapitres :

Chapitre 1. Étude Théorique de l'AES : Comprendre les principes de fonctionnement de l'AES, y compris ses transformations principales (SubBytes, ShiftRows, MixColumns, AddRoundKey) et la gestion des clés.

Chapitre 2. Conception en VHDL : Traduire les opérations de l'AES en modules VHDL, chacun correspondant à une étape spécifique du processus de chiffrement et de déchiffrement.

Chapitre I : Étude Théorique de l'AES

i. Introduction

AES (Advanced Encryption Standard) est un algorithme de chiffrement symétrique largement utilisé pour protéger les données sensibles. Il a été développé par deux cryptographes belges, Joan Daemen et Vincent Rijmen, et a été adopté par le gouvernement des États-Unis comme norme pour sécuriser les informations classifiées en 2001. AES utilise un chiffrement par bloc, ce qui signifie qu'il chiffre les données en blocs de taille fixe, généralement de 128 bits de longueur. L'algorithme utilise une série d'opérations mathématiques, incluant la substitution, la permutation et l'XOR, pour transformer le texte en clair en texte chiffré. AES prend en charge des tailles de clés de 128, 192 ou 256 bits, et est considéré comme l'un des algorithmes de chiffrement les plus sûrs disponibles aujourd'hui. Il est utilisé dans une large gamme d'applications, y compris la banque en ligne, le commerce électronique et les communications sécurisées.

ii. Cas d'usages :

1. Sécurisation des communications en ligne : AES est utilisé pour chiffrer les communications en ligne, telles que les e-mails, les messages instantanés et les appels vidéo, afin de protéger les données sensibles contre les cyberattaques.

2. Stockage de données sensibles : AES est utilisé pour chiffrer les données stockées sur des disques durs, des clés USB et d'autres supports de stockage, afin de protéger les informations sensibles contre les accès non autorisés.

3. Transactions financières : AES est utilisé pour sécuriser les transactions financières en ligne, telles que les paiements par carte de crédit et les transferts de fonds, afin de protéger les informations financières des clients contre les fraudes.

4. Sécurisation des réseaux : AES est utilisé pour chiffrer les données qui circulent sur les réseaux, tels que les réseaux d'entreprise et les réseaux sans fil, afin de protéger les informations sensibles contre les interceptions et les attaques.

5. Sécurisation des données gouvernementales: AES est utilisé par les gouvernements pour protéger les données sensibles, telles que les informations militaires et les données de renseignement, contre les cyberattaques et l'espionnage.

iii. Principe de l'AES :

Le fonctionnement d'AES est compliqué mais ce qu'il faut retenir c'est que c'est un algorithme de chiffrement par bloc. C'est à dire que les données à chiffrer vont être découpées par blocs. A ces derniers, on applique différentes opérations mathématiques pour obtenir des blocs de sorties.

En cryptographie, c'est le principe de réseau de permutation-substitution.

AES est une variante de Rijndael, avec :

Une taille de bloc fixe de 128 bits

Une taille de clé de 128, 192 ou 256 bits.

La taille de la clé utilisée pour un chiffrement AES spécifie le nombre de cycles de transformation qui convertissent l'entrée.

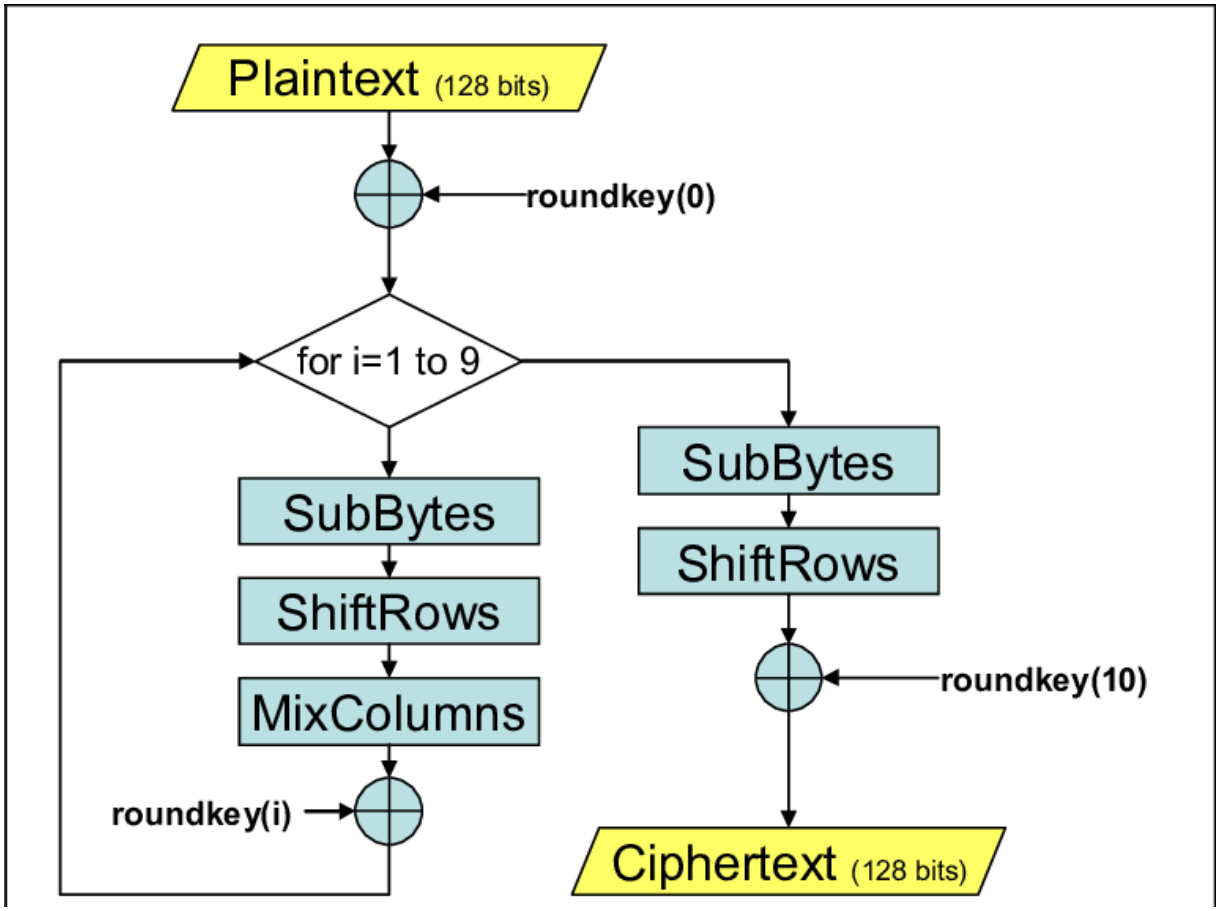
On obtient le nombre de tours le suivant:

10 tours pour les clés 128 bits.

12 tours pour les clés 192 bits.

14 tours pour les clés 256 bits.

Ainsi, plus la taille de la clé est importante, plus le nombre de combinaison possibles.



- Étape une : le stockage des données dans un « carré » de 4 x 4 = 16 cases ensuite dans une matrice 4 x 4 appelée Ai,j . Chaque case contient 1 octet (8 x 16 = 128 bits d'état interne).

- ## Initial Round :

  - ### Add RoundKey

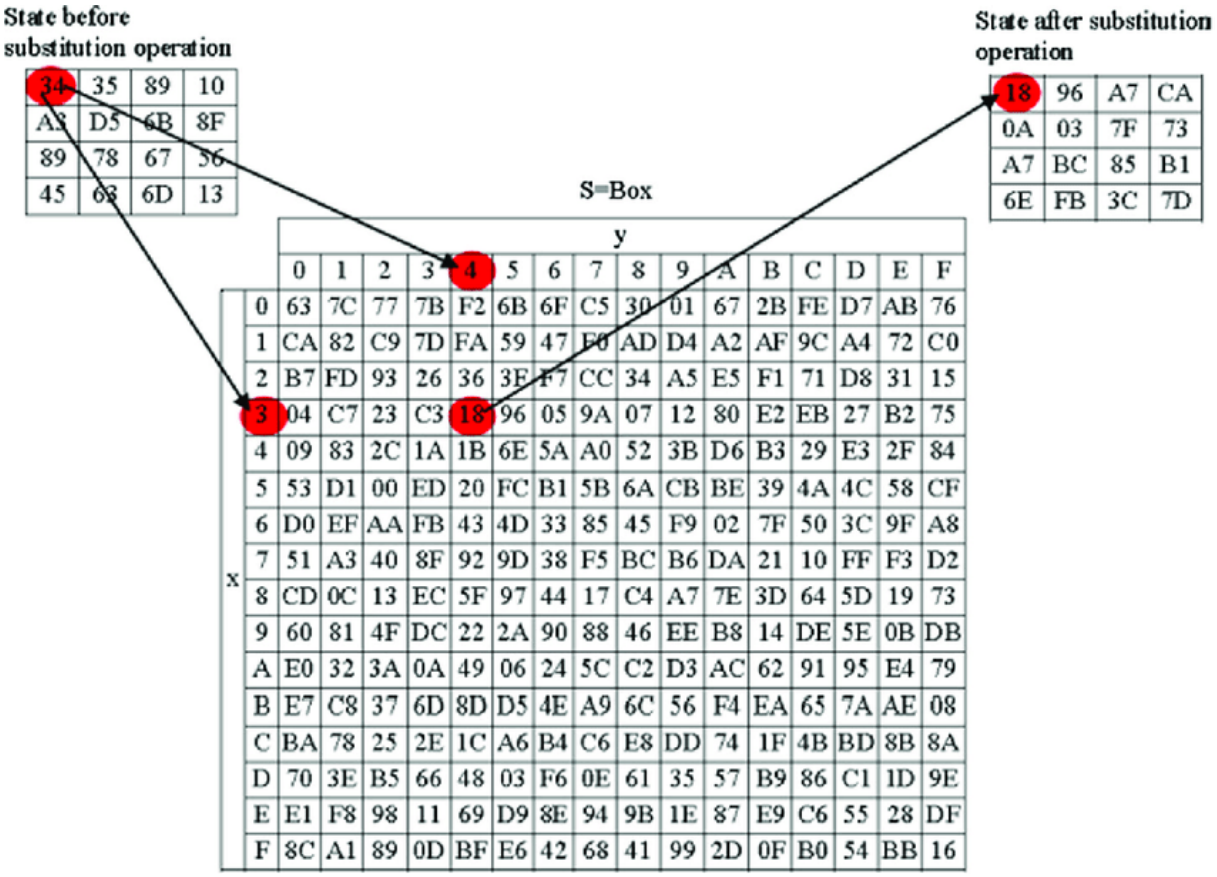
XOR la matrice avec la sous-clé , $Ai,j \oplus Ki,j = Bi,j$

- ## Main Rounds (répétés pour chaque tour sauf le dernier):

  - ### SubBytes

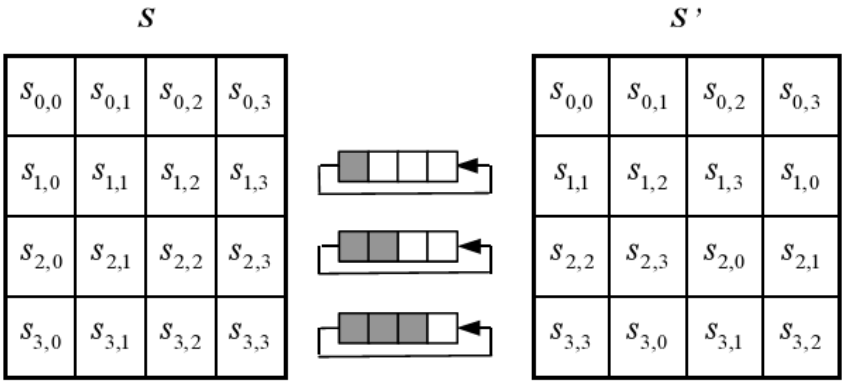
Une substitution non linéaire est appliquée à chaque octet du bloc en utilisant une table de substitution appelée S-box. (Confusion).

S-Box pour AES



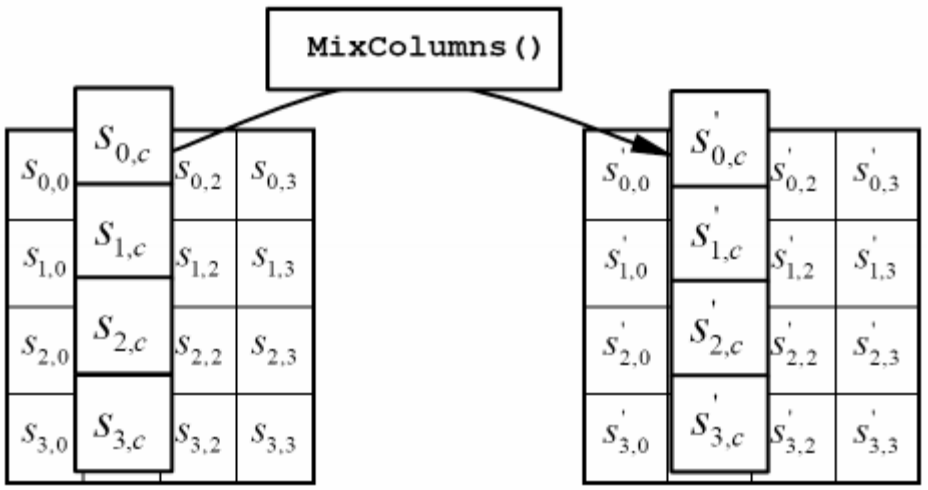
x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	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	13	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	C6	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

- **ShiftRows** : Consiste à décaler les lignes en rotation: transformation linéaire (diffusion). Décalage circulaire (vers la gauche) de i cases pour la ligne numéro i, $0 \leq i \leq 3$



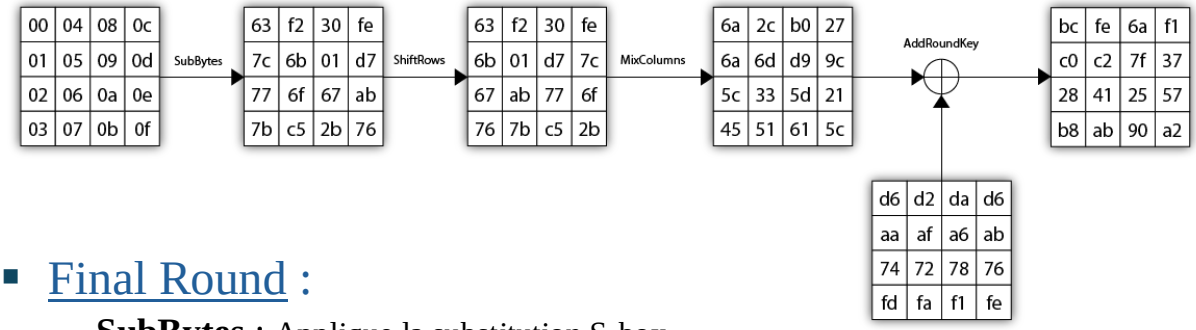
- **MixColumns :** Les colonnes du bloc sont mélangées en utilisant une transformation linéaire. Pour chaque colonne on applique une multiplication par une matrice circulante.

Cette étape combine les octets de chaque colonne pour diffuser les bits à travers le bloc.



- **AddRoundKey :** Un XOR est effectué entre le bloc résultant et la clé de ronde suivante.

Exemple :



- Final Round :
  - **SubBytes** : Applique la substitution S-box.
  - **ShiftRows** : Applique le décalage circulaire des lignes.
  - **AddRoundKey** : Un XOR est effectué avec la dernière clé de ronde. La transformation MixColumns n'est pas utilisée dans le dernier tour.

iv. Gestion des Clés (Key Expansion)

L'un des aspects cruciaux de l'AES est la gestion des clés, ou l'expansion de la clé. La clé initiale est étendue pour générer une série de clés de ronde, chacune utilisée à un tour différent de l'algorithme. Le processus d'expansion de la clé comprend plusieurs opérations, notamment des rotations de mots, des substitutions à l'aide de la S-box, et des XOR avec des constantes de ronde spécifiques. Cette expansion assure que chaque clé de ronde est unique et dérivée de la clé originale, renforçant ainsi la sécurité globale de l'algorithme.

Étapes de l'Expansion de Clé

1. **Initialisation** :
   - La clé d'entrée est copiée dans les premières positions du tableau de clé élargie.
2. **Expansion des Blocs** :
   - **Rcon (Round Constant)** : Une constante de ronde est utilisée pour chaque groupe de 16 octets générés.
   - **Rotation** : Les 4 derniers octets du bloc de clé sont tournés à gauche d'un octet.
   - **Substitution** : Chaque octet du bloc tourné est substitué à l'aide de la S-box.
   - **Ajout de Rcon** : Le premier octet du bloc tourné est XORé avec Rcon.
3. **Génération de la Clé Élargie** :

- Les blocs de 4 octets sont générés en utilisant une combinaison de XOR entre les blocs précédemment générés et les transformations effectuées avec Rcon et la S-box.

v. Sécurité de l'AES

La sécurité de l'AES repose sur plusieurs principes :

- ***Diffusion*** : Assurée principalement par les transformations ShiftRows et MixColumns, qui répandent les bits de la clé et du texte chiffré à travers tout le bloc.
- ***Confusion*** : Apportée par les substitutions non linéaires de la S-box, rendant la relation entre le texte clair et le texte chiffré complexe et non linéaire.
- **Résistance aux attaques** : AES est conçu pour être résistant à diverses attaques cryptographiques, notamment les attaques par force brute, les attaques linéaires et les attaques différentielles. La taille des clés (128, 192, ou 256 bits) rend les attaques par force brute impraticables avec les technologies actuelles.

Chapitre II : Conception en VHDL.

La conception de l'algorithme AES en VHDL est une étape cruciale qui consiste à traduire les opérations théoriques de l'AES en modules VHDL distincts. Chaque module correspond à une étape spécifique du processus de chiffrement, permettant ainsi une modélisation claire et modulaire du système. Cette section détaille le processus de conception, en mettant l'accent sur les principaux composants et transformations de l'AES.

Dans Le programme VHDL, on va implémenter chaque transformation de l'AES à l'aide de composants spécifiques.

1. **Clé Initiale et Message**
   Le message initial et la clé d'entrée sont enregistrés dans des registres pour une utilisation ultérieure.

```
Reg_init : register_16_Bytes port map (
    D => message,
    clk => clk,
    rst => rst,
    Q => init_message
);
```

2. **Expansion de Clé**
   La composante key_expansion s'occupe de générer les clés pour toutes les rondes de chiffrement.

- Initialisation des variables et de la clé d'entrée.
- Lecture et transformation des blocs de 4 octets (cores).
- Application de transformations spécifiques (rotation, substitution, Rcon) tous les 16 octets.
- Génération des clés élargies via des opérations XOR.

3. **Ronde Initiale**
   La première étape de transformation ajoute la clé initiale au message.
   « add_round_key »

4. **Rondes Principales**
   Chaque ronde applique successivement les transformations SubBytes, ShiftRows, MixColumns, et AddRoundKey.

5. **Ronde Finale**
   La dernière ronde ne contient pas la transformation MixColumns.

```
SubBytes_FinalRound : sub_bytes port map (
    state_in => trans_states_round(64*N_rounds + 15 downto 64*N_rounds),
    enable => enable,
    state_out => trans_state_subbytes
);
ShiftRows_FinalRound : shift_rows port map (
    state_in => trans_state_subbytes,
    enable => enable,
    state_out => trans_state_shiftrows
);
AddRoundKey_FinalRound : add_round_key port map (
    state_in => trans_state_shiftrows,
    key => expanded_key(175 downto 160),
    enable => enable,
    state_out => trans_state_addroundkeys
);
```

6. **Enregistrement du Message Chiffré**
   Le message chiffré final est enregistré.

Conclusion

En conclusion, l'AES est un algorithme de chiffrement robuste et efficace, largement adopté en raison de sa capacité à protéger les données contre les accès non autorisés et les altérations malveillantes. Sa structure modulaire et ses étapes clairement définies facilitent son implémentation en matériel, notamment à l'aide de langages de description comme VHDL, ce qui en fait un choix idéal pour les applications nécessitant des solutions de sécurité embarquées.