Chiffrement AES

LEMAIRE Théo

November 5, 2023

Contents

1 Exactitude du Projet **2**
1.1 Test du code 2

2 Structure du projet **2**
2.1 Fichiers 2
2.2 Fonctions 2

3 Implémentation de AES **2**
3.1 Choix du langage 2
3.2 Choix de types de données 2
3.3 Fonctions 3

4 Sources **3**

1 Exactitude du Projet

1.1 Test du code

J'ai développé ce projet moi-meme, et j'ai appuyé mes tests sur cet article: https://www.kavaliro.com/wp-content/uploads/2014/03/AES.pdf Le message en clair et la clé de chiffrement sont les memes que dans l'article pour correspondre à l'exemple afin de comparer les resultat de mon implémentation de AES en langage C aux resultats de l'algorithme décris dans l'article.

2 Structure du projet

2.1 Fichiers

Le projet à été réalisé dans un repertoire "AES", il contient un fichier "AES.c", ou se trouve la fonction main() ainsi que l'ensemble des fonctions utiles au projet. Les prototypes des fonctions ainsi que les constantes sont dans le fichier "AES.h". Un makefile à été mis à disposition pour compiler directement le projet en portant une attention particulière à la gestion de la mémoire avec "-fsanitize=address".

2.2 Fonctions

Les fonctions sont rédigées séparéments dans "AEC.c", la fonction AES() se charge de l'execution du programme, nous devons simplement lui passer en paramètre le message en clair avec la clé de chiffrement correspondante. Une variable "debug" a été mise à disposition dans "AES.h" pour controler la bonne execution du chiffrement etapes par etapes, comme dans l'article.

3 Implémentation de AES

3.1 Choix du langage

J'ai personnellement choisis d'utiliser le langage C dans le cadre d'un chiffrement AES car il est fortement typé, j'avais besoin de connaitre précisément les types des variables que je manipulais, car utilisant des données de types très différent, coder ce projet en python aurais créer un flou quant à la gestion de ces types. De plus, l'aspect performance rentre également en compte dans un projet de chiffrement, on ne veux pas qu'il y ais des fuites de mémoire invisibles à petite échelle, qui pourraient poser problème sur des messages plus grands. Le langage C est donc plus adapté selon moi.

3.2 Choix de types de données

- La structure "DATA"
  J'ai choisis d'implémenter une structure DATA, elle à un attribut "matrix" qui est une matrice deux dimensions d'unsigned char qui représente la donnée, la représentation matricielle est importante pour moi dans le cadre d'AES car elle permet une visualisation claire des opérations de chiffrement, l'attribut "size" représente la taille de la matrice, pour faciliter le parcours de celle-ci.

- Unsigned char
  J'ai choisis le type unsigned char pour représenter des données en hexadécimal, dans AES, je suis amené à effectuer des décalages de bits, et à séparer les bits de poids faible et de poids fort, également à faire des Xor, ce type de donnée est donc adapté.

- Taille des données
  J'ai choisis de travailler avec des données de 128 bits, ce qui représente un message de 16 octets, soit 16 caractères, car un caractère équivaut à deux caractères hexadécimaux, pesant chacuns 4 bits. J'explique mon choix par la facilité de visualisation des données, trivialement, les matrices des datas de taille 128bits sont plus lisibles que des matrices de 256 bits lors d'un deboguage.

3.3 Fonctions

Les fonctions d'AES utilisées sont les suivantes:

- AddRoundKey
  Effectue un Xor entre une matrice temporaire (etat du message en cours de chiffrement), et une clé de tour.
- SubBytes
  Décompose le caractère Héxadécimal en deux, et cherche dans la S-Box sa correspondance avec en absice le premier caractère et en ordonné le second.
- ShiftRows
  Décalage d'un octet vers la gauche de la ligne 1 de la matrice de la data, de 2 pour la ligne 2 et de 3 pour la dernière.
- MixColumns
  Etape la plus complexe, on effectue une transformation sur chacunes des colones de la matrice actuelle. On multiplie chaque octets de chaque colone de manière linéaire dans le champ fini de Galois.
- AES
  Reproduit le chiffrement AES en prenant en argument un texte clair et une clé de chiffrement, cette fonction utilise les fonctions précedemment citées.

4 Sources

Liens vers la documentation utilisée:

- https://en.wikipedia.org/wiki/Rijndael$_{M}ixColumns$
- https://www.emse.fr/ dutertre/documents/synth$_{A}ES128.pdf$
- https://www.youtube.com/watch?v=qBj1l3jF$_{g}E$
- https://www.kavaliro.com/wp-content/uploads/2014/03/AES.pdf