AES - Rapport

Corentin Bouchard

May 25, 2024

1 Implémentation

Cette implémentation permet de chiffrer un fichier avec l'AES-128 en mode ECB. Après lecture d'une clé dans un fichier, le programme peut chiffrer/déchiffrer bloc par bloc et écrire dans un fichier de sortie le résultat. Les modules de l'implémentation vont être décrits plus en détail ci-après. Pour la notice du programme voir le fichier "notice.txt".

1.1 Main

Ce module gère un parsing d'options pour faciliter l'utilisation du programme. L'utilisateur peut alors choisir une clé, un texte à chiffrer/déchiffrer via l'exécution du programme. Un texte et une clé par défaut sont à disposition pour tester le programme plus facilement. Tous les résultats d'opération de chiffrement/déchiffrement sont stockés dans un fichier de sortie, qui peut être réutilisé par le programme ou lu tel quel. Une fonction permet de lire une clé depuis un fichier, voir "notice.txt" pour plus de détails.

1.2 Multiplication

Ce module gère la partie multiplication dans $GF(2^8)$. L'implémentation ne fait pas de compromis sur la mémoire et toutes les multiplications sont toujours calculées par le programme. Toutes les manipulations sont faites sur des ensembles de 8 bits, ici représentés par des char.

1.3 Cipher

De nombreuses structures sont définies dans ce module pour stocker les différentes valeurs calculées lors du chiffrement/déchiffrement. Essentiellement, toutes ces structures se retrouvent être des tableau de char, étant facile à manipuler et tous de la même taille (et cela, quelque soit le processeur). Le module peut convertir un block en State, effectuer le chiffrement/déchiffrement sur celui-ci, et reconvertir en un bloc. Il assure donc toutes les fonctions nécessaires au chiffrement/déchiffrement, et à la génération des clés de rondes. Il ne fonctionne cependant qu'avec des clés de taille 128, plus de détails sur le choix des structures dans le paragraphe "Difficultés rencontrées".

1.4 Modes

Implémentation du mode ECB, la fonction lis un fichier, la coupe block par block et utilise les fonctions de chiffrement/dechiffrement pour sortir le fichier de sortie.

2 Difficultés rencontrées

La principale difficulté a été de choisir comment stocker les données, clés, state, block. Au départ, je voulais utiliser une union, pour permettre selon le contexte d'utiliser la forme la plus adéquate pour manipuler les données. Le problème avec l'union, c'est que la manière dont est gérée la mémoire dépend de l'endianness ("boutisme" en français) du processeur. L'implémentation aurait été plus difficilement portable, ou il aurait fallu réécrire nombre de fonctions en prenant en compte l'endianness. Non satisfait des deux options, j'ai choisi de faire plusieurs structures, s'approchant en termes de concept

des objets de l'algorithme d'AES, mais cela au coût de quelques conversions de temps à autre. Le document du NIST étant particulièrement clair, je n'ai pas eu de soucis particuliers pour programmer l'algorithme.