PX222 IRC - Jalon intermédiaire
Advanced Encryption Standard

Alexandre VINHAS - Vincent MOUCADEAU — 2A

29/05/2023

1 Objectifs du projet

Le but de ce projet est de réaliser un programme de chiffrement et de déchiffrement de textes/-fichiers en utilisant l'algorithme AES (Advanced Encryption Standard). Ce programme devra être capable de chiffrer et déchiffrer des fichiers de taille quelconque, en utilisant une clé de 128, 192 ou 256 bits. Nous devrons implémenter l'algorithme en Haskell et en C. Cela nous permettra de comprendre le fonctionnement d'AES, un algorithme de chiffrement très utilisé (car très robuste), et de mettre en application nos connaissances en Mathématiques aquises au premier semestre (polynômes, structures algébriques...).

2 Etat du projet

A l'heure actuelle, voici les tâches que nous avons réalisées :

- Recherches sur l'algorithme AES et sur les structures algébriques utilisées (corps finis, polynômes...)
- Implémentation complète de l'algorithme AES en Haskell (chiffrement et déchiffrement avec des clés de 128, 192 et 256 bits)
- Implémentation complète de l'algorithme AES en C (chiffrement et déchiffrement avec des clés de 128 bits)

A ce stade, il nous reste donc à corriger l'implémentation en C pour qu'elle fonctionne avec des clés de 192 et 256 bits. Les objectifs que nous nous étions fixés sont donc presque atteints. Nous aimerions par la suite poursuivre le projet en ajoutant des fonctionnalités supplémentaires, comme par exemple la possibilité de chiffrer/déchiffrer des fichiers de taille quelconque en C ou encore une interface graphique (si nous avons le temps).

3 Fonctionnement des programmes

3.1 AES en Haskell

Le programme en Haskell est composé d'un dossier `Math` qui rassemble les fichiers définissant les structures algébriques utilisées (corps finis, polynômes...), la partie 4 de la documentation FIPS (préliminaires mathématiques) avec les opérations sur les polynômes de GF256, les `words` de 32 bits (polynômes à coefficients dans GF256) ainsi que les states. Le fichier `Cipher.hs` importe l'ensemble des modules du dossier `Math` et définit l'ensemble des transformations sur les states, les clés (keyexpension) ainsi que les fonctions de chiffrement et de déchiffrement. Voici les fonctions principales du programme :

- `encode :: String -> String -> String` : prend en entrée une clé et un texte et renvoie le texte chiffré
- `decode :: String -> String -> String` : prend en entrée une clé et un texte chiffré et renvoie le texte déchiffré

Exemple avec une clé de 128 bits et un chaine de 17 caractères :

```
ghci> encode "2B7E151628AED2A6ABF7158809CF4F3C" "azertyuiopqsdfgh5"
"\147\169\152\188\180R\245\219\190\228%b\248..."
```

```
ghci> decode "2B7E151628AED2A6ABF7158809CF4F3C" "\147...7x"
"azertyuiopqsdfgh5"
```

3.2 AES en C

Le programme en C est composé d'un dossier `maths` qui contient les fichiers définissant les types et les fonctions sur les polynômes et les mots de 32 bits. Le dossier `algorithm` contient les fichiers définissant les fonctions sur les states, les clés et l'algorithme Cipher et InvCipher. Le fichier `main.c` contient quelques tests et définit les fonctions principales du programme :

- `char *encodetext(char *key, char *text)` : prend en entrée une clé et un texte et renvoie un pointeur vers le texte chiffré en hexadécimal (allocation dynamique).
- `char *decodetext(char *key, char *text)` : prend en entrée une clé et un texte chiffré (en hexadécimal) et renvoie un pointeur vers le texte déchiffré (allocation dynamique).

Le programme ne peut pour l'instant pas intéragir avec l'utilisateur, il faut modifier le code pour changer la clé et le texte à chiffrer/déchiffrer. Voici un exemple avec une clé de 128 bits et un chaine de caractères :

```c
char testkey1[] = "2b7e151628aed2a6abf7158809cf4f3c";
char toencode[] = "J'aime les pates aux basilic !";
char *encoded = encodetext(testkey1, toencode);
char *decoded = decodetext(testkey1, encoded);
printf("To encode: %s\n", toencode);
printf("Ciphered: %s\n", encoded);
printf("Unciphered: %s\n", decoded);
free(encoded);
free(decoded);
```

Listing 1: Contenu de la fonction `main`

```
To encode: J'aime les pates aux basilic !
Ciphered: f52083fb00fd77b552cd17a53eadca86e10b2a444038cddfae11dcd9d94a1622
Unciphered: J'aime les pates aux basilic !
```

Listing 2: Résultat de l'exécution du programme `aes.out`