Agnon Kurteshi, Zabiullah
Ahmadi, Dawid Dymarczyk
Groupe 12

h e p i a
Haute école du paysage, d'ingénierie
et d'architecture de Genève

TP Reed Solomon

1. Introduction

Pour commencer, au cours du cours de math on nous a demander de décoder un message crypté par un codage Reed-Solomon

Pour y arriver nous avons créé une librairie de fonctions qui va nous permettre de faire des opérations entre Polynôme, ensuite d'adapter la méthode Reed Solomon pour le décodage du message.

Le cryptage Reed-Solomon consiste à rajouter 2 fois le nombre d'erreurs possible crypté par le même polynôme du départ qui est du même degré que le message envoyé. Sois dans notre cas 31-1=30, pour permettre au receveur de décoder le message même si des erreurs se sont glissées dedans. S'il y a plus de 2 fois le nombre d'erreur alors il est presque impossible de trouver le bon polynôme de départ

2. Fonction de manipulation de polynômes

Afin de pouvoir faire le TP du Reed Solomon, on a estimé qu'il y aurait deux fonctions majeures qui allaient nous servir pour le calcul de polynômes. C'est l'addition de deux polynômes, et le produit de deux polynômes multiplier par un scalaire (nombre premier). Ces deux fonctions vont nous servir pour faire Lagrange.

3. Fonction de Euclide étendue et inverse modulaire

Un point très important à noter, c'est lors de Lagrange, il y aura des nombres immenses à gérer, car nous manipulons un polynôme de degré 31 et cela risque de dépasser la mémoire de l'ordinateur et de faire crasher le programme, ou bien de le faire surchauffer un peu vu les nombres immenses qu'il va devoir calculer. Donc pour éviter cela, il faut qu'on utilise la fonction de Euclide étendu qui va nous retourner l'inverse modulaire du nombre. Ainsi, tous les calculs qui se feront seront congruent au nombre premier utilisé. Cela diminue grandement les chiffres utilisés.

Agnon Kurteshi, Zabiullah
Ahmadi, Dawid Dymarczyk
Groupe 12

h e p i a
Haute école du paysage, d'ingénierie
et d'architecture de Genève

4. Fonction Lagrange

Comme nous avons 53-24=29 point juste après le 24eme nombre, et que notre message a 31 éléments, nous fessons des combinaisons de 2 des 24 éléments potentiellement faux, et fabriquons des polynômes Lagrange de degré 30 avec 29 points juste + 2 potentiellement faux. Si le polynôme Lagrange correspond au 2 donné on a trouvé notre polynôme du départ

Nous avons 11 faux dans 24 éléments, sois 13 juste, 13 juste font 78 possibilités d'être juste à trouver le bon polynôme

5. Conclusion Décodage

Pour décoder une fois le polynôme trouvé il suffit de remplacer x par la position de la lettre dans le message et de faire modulo le nombre premier pour trouver le message d'origine, nous avons tester les messages d'autre groupe et nous les avons aussi décodés, pour décoder d'autre message il suffit de remplacer dans notre code, la liste rentrée, le nombre premier, et séparé la liste à l'indice ou les erreurs se finisse.

Les données que le groupe 12 a Reçu. Et le message décoder.

```
Le nombre premier              : 337
La Longueur du message de base : 31
Le nombre de points ajoutés : 22
Le nombre maximal d'erreurs    : 11
Qui sont situées avant l'indice : 24
La liste reçue : [231, 97, 38, 100, 233, 99, 111, 121, 101, 32, 145, 233, 154,
271, 114, 146, 80, 151, 151, 110, 115, 95, 99, 170, 32, 103, 114, 111, 117, 112,
101, 246, 42, 223, 132, 270, 304, 146, 149, 234, 187, 250, 62, 146, 25, 192, 273,
142, 2, 218, 193, 202, 222]
```

Le message : ça décode sévère dans ce groupe