Codes correcteurs et cryptosystème de Mc Eliece

Auclair Pierre

15 mai 2014

Le cryptosystème de Mc Eliece est un système de cryptage qui s'appuie sur la théorie des codes correcteurs. Il se situe à intersection des soucis de fiabilité et de sécurité de l'information. C'est pourquoi nous avons implémenté ce système en Python dans le cadre du sujet transfert et échange. La totalité des ressources est disponible ici : https://github.com/kalaspa/mc-eliece

1 Préliminaires

Codes correcteurs

Tout d'abord, définissons quelques notions des codes correcteurs. Un code correcteur est l'image C d'une application linéaire f de $\mathbb{F}^k$ vers $\mathbb{F}^n$. On définit, avec les conventions du cours de mathématiques, les matrices génératrices G et de parité H :

$$\begin{aligned} G &= Mat(f) \\ Ker(H) &= Im(f) \end{aligned}$$

La distance minimale d du code est le poids de Hamming le plus faible d'un mot non nul du code. La capacité de correction t d'un code correcteur vérifie cette relation :

$$t \leq \frac{d-1}{2}$$

Dans la limite de cette capacité de correction, le syndrome Hx d'un mot $x \in \mathbb{F}^n$ est caractéristique de l'erreur, ce qui permet la correction.

Outils informatiques

Pour démarrer le projet, il a fallu développer des modules pour utiliser des matrices et des polynômes sur des corps finis. Pour cela, nous avons utilisé la programmation orientée objet et la surcharge des opérateurs permise par Python. Nous avions ainsi des classes de polynômes et de matrices à coefficients aussi bien réels que de $\mathbb{F}_{p^m}$.

Une optimisation que nous avons trouvé consiste à considérer uniquement le cas $p = 2$. Ainsi, un élément de $\mathbb{F}_{2^m}$ équivalent à un polynôme dans $\mathbb{F}_2[X]/P$ peut être stocké comme un nombre en base binaire. Les opérations sur ces éléments deviennent des opérations binaires beaucoup plus rapides que l'implémentation plus mathématique.

Algorithmes supplémentaires

Pour compléter le projet, nous avons implémenté les algorithmes de Gauss et de Berlekamp-Hensel. En plus du calcul de l'inverse, le pivot de Gauss nous a servi à déterminer le noyau d'une matrice ou un inverse à gauche d'une matrice non carré. L'algorithme de Berlekamp-Hensel permet de vérifier l'irréductibilité d'un polynôme dans un corps fini. Étant donné qu'on doit en générer aléatoirement pour chaque clef, son usage était indispensable.