

TD - Programmation et structures de données

Guillaume Bonfante - Pierre-Etienne Moreau



Exercice 1

Nous allons représenter les polynômes dans $\mathbb{Z}_q[x]/(x^{n+1})\mathbb{Z}_q[x]$ avec q et n deux entiers. On rappelle : les coefficients sont dans $\mathbb{Z}_q$, c'est-à-dire dans $[0, q[$

et que

$$x^n = -1 \text{ mod. } (x^{n+1}).$$

Proposer une classe pour représenter un tel polynôme. On pourra reprendre le TD2.

Étendre cette classe pour offrir la méthode `add` et `mul`. Les deux polynômes doivent avoir les mêmes paramètres q et n .

Écrire un jeu de tests avec des `assert` (ou le module `unittest` si vous vous sentez à l'aise, <https://docs.python.org/fr/3/library/unittest.html>)

Exercice 2

Toujours dans la classe des polynômes, rajouter une fonction `scalar` qui multiplie les coefficients d'un polynôme par un scalaire

```
def scalar(self, c): # c * self
    ...
```

En sus, une fonction `rescale` qui passe de P dans $\mathbb{Z}_q[x]/(x^{n+1})\mathbb{Z}_q[x]$ au polynôme avec les mêmes coefficients mais dans $\mathbb{Z}_r[x]/(x^{n+1})\mathbb{Z}_r[x]$:

```
def rescale(self, r)
    ...
```

Et enfin, une fonction `fscalar` qui attribue à P dans $\mathbb{Z}_q[x]/(x^{n+1})\mathbb{Z}_q[x]$, r et un paramètre α dans $\mathbb{R}$, le polynôme Q avec

$$Q = \sum_{i=0}^{n-1} [\text{round}(P[i] * \alpha) \% r] X^i$$

Exercice 3

Définir une fonction permettant de construire un polynôme dont les coefficients sont tirés au hasard dans un intervalle $[a,b]$. On fera l'hypothèse que le polynôme est dans $\mathbb{Z}_q[x]/(x^{n+1})\mathbb{Z}_q[x]$.

```
def gen_uniform_random(q, n, a, b):
    ...
```

La génération des clés publiques et secrètes se définit comme suit :

La clé privée sk est un polynôme dans $Z_q[x]/(x^{n+1})Z_q[x]$ dont tous les coefficients sont tirés au hasard dans $[0, 1]$

La clé publique est une paire de polynômes (b,a) dans $Z_q[x]/(x^{n+1})Z_q[x]$ avec :

$$b = -(a * sk + e)$$

où a est tiré au hasard avec des coefficients dans $[0, q[$ et pareil pour e sauf que les coefficients sont dans $\{0, 1, q-1\}$.

Exercice 4

Etant donné une clé publique $pk = (b,a)$ dans $Z_q[x]/(x^{n+1})Z_q[x]$, un polynôme p dans $Z_t[x]/(x^{n+1})Z_t[x]$, le chiffrement de p par la clé publique est la paire de polynôme $(c1,c2)$ avec

$$\left\{ \begin{array}{lcl} \delta & = & q/t \\ s_p & = & p * \delta \\ u & = & \text{gen_uniform_random}(q, n, 0, 1) \\ e_1 & = & \text{gen_uniform_random}(q, n, -1, 1) \\ c_1 & = & (b * u) + e_1 + s_p \\ e_2 & = & \text{gen_uniform_random}(q, n, -1, 1) \\ c_2 & = & a * u + e_2 \end{array} \right. \in Z_q/(x^n + 1)$$

Exercice 5 (enfin)

Pour déchiffrer un polynôme chiffré $(c1,c2)$ dans $Z_q[x]/(x^{n+1})Z_q[x]$ avec la clé secrète sk , on calculera $(c1 + c2*sk).fscalar(t/q,t)$

Vérifier que l'addition chiffrée est l'addition séparée des deux composantes chiffrées.

Exercice 6 (pour la faim)

Comment faire la multiplication ?