Introduction à la cryptographie

Devoir maison

BOUCHER Yohan
JACQUIER Timéo

Master 1 - Informatique

Langage utilisé: Python

Configuration: Compilation à la racine du projet : **python3 cryptographie.py**

Temps d'exécution: environ 2 minutes

Question 1

Quel langage de programmation avez-vous choisi ?

Nous avons utilisé le langage Python pour réaliser l'entièreté du devoir maison.

Quelle bibliothèque permettant de gérer des nombres entiers de grande taille allez-vous utiliser ? Quelles sont les opérations implémentées dans cette bibliothèque (multiplication, addition,...)

En python, il n'est pas nécessaire d'utiliser une bibliothèque afin de générer des nombres entiers de grande taille puisqu'en Python, il n'y a pas de limite de taille dans les nombres. En revanche, nous avons importer le module **sys** qui nous a permis de modifier le nombre maximal de récursions possible dans une même fonction.

sys.setrecursionlimit(2048)

Question 2

En vous aidant d'internet, donnez la définition d'un nombre aléatoire cryptographiquement sûr.

Nombre aléatoire cryptographiquement sûr:

Un Nombre aléatoire cryptographiquement sûr est un générateur de nombre pseudo-aléatoire ayant des propriétés adaptées pour le domaine de la cryptographie. Il s'agit entre autres d'une fonction qui prend une valeur d'entrée appelée "graine" et qui sort un nombre qui à l'apparence d'un nombre généré aléatoirement.

Source :
https://boowiki.info/art/les-generateurs-de-nombres-pseudo-aleatoires/generateur-de-nombres-pseudo-aleatoires-cryptographiquement-securise.html
https://openclassrooms.com/fr/courses/1757741-securisez-vos-donnees-avec-la-cryptographie/6031863-generez-des-nombres-aleatoires

En Python, le module **random** permet de générer ces nombres aléatoires.

Pour les questions suivantes, des commentaires permettent d'expliquer le code et à quel endroit dans le code sont exécutées les fonctions. Chacunes des fonctions enregistreront les données dans les fichiers textes correspondants.

Question 3

Fonction **Euclide()** implémentée avec 10 000 test enregistrés dans le fichier *Euclid.txt*

Question 4

Fonction **ExpoMod()** implémentée avec 10 000 test enregistrés dans le fichier *ExpoMod.txt*

Question 5

Fonction **KeyGen()** implémentée avec 10 000 test enregistrés dans le fichier *KeyGen.txt*

Fonction **Encrypt()** implémentée avec 10 000 test enregistrés dans le fichier *Encrypt.txt*

Fonction **Decrypt()** implémentée avec 10 000 test enregistrés dans le fichier *Decrypt.txt*

Question 6

Fonction **Homomorphique()** implémentée avec 10 000 test enregistrés dans le fichier *Homomorphique.txt*