Moahammade Dalati
Delconte Florian
Master 1ère année

RAPPORT CRYPTOGRAPHIE

INFORMATION COMPLÉMENTAIRE :
Fichiers : main.py , elGamal.py
Execution : python3 main.py

Pour tester notre programme il vous faut télécharger et installer la librairie GMPY2.
L'exécution de main va faire les différents tests demandés et écrire dans le fichier test.txt les 5 premières occurrences de chaque test.

Question 1.

Nous avons choisi d'utiliser Python comme langage de programmation pour deux raisons :

- Premièrement : nous avions jamais essayé ce langage et nous avions envie de le découvrir.
- Deuxièmement : Pour notre projet d'initiation a la recherche, nous allons être confronté à son utilisation, donc autant apprendre à le manier dès maintenant.

En parcourant des forums sur le langage python nous avons découvert la librairie GMPY2 pour gérer les grands nombres et les nombres aléatoires. Nous utilisons le module mpz de GMPY2 pour manier des nombres entiers.
Beaucoup d'opérations sont implémentées dans la bibliothèque dont : Addition, soustraction, multiplication, division, modulo, random , etc.

Question 2.

Nombre aléatoire cryptographiquement sûr : C'est un nombre produit par un générateur capable de produire une sortie suffisamment peu discernable d'un aléa parfait. Ce générateur doit résister à des attaques d'injection de données. Et il doit être très difficile en connaissant x nombres, de prédire le prochain nombre que le générateur va générer.
La bibliothèque GMPY2 permet la génération de ces nombres.

Question 3.

Remarque : Nous testons Euclide avec 10 000 valeurs différentes de 'a' que $a \cdot u + p \cdot v = 1$, et nous écrivons seulement les 5 premiers tests dans le fichier test.txt. Ce qui peut expliquer la lenteur de l'exécution du main.py

Question 4.

Remarque : Nous testons Expmod avec 10 000 valeurs différente de 'a', et nous écrivons seulement les 5 premiers tests dans le fichier test.txt. Ce qui peut expliquer la lenteur de l'exécution du main.py

Question 5,6.

Même remarque.