Projet cryptographie

Lagarde Tristan - Aiachi Morade

March 2, 2025

1 Quelle langage ? Quelles fonctionnalités ?

1.1 Question 1.

Nous avons choisi le langage python et la bibliothèque "gmpy2" qui permet les opérations suivantes :

- Les opérations de bases (Addition, soustraction, multiplication, ...etc)
- Les opérations sur les puissances (exponentiation, exp modulaire, racine carée, racine n-ième, ...etc)
- Les opérations modulaires (inverse, primalité, nombre premier, ...etc)
- D'autres opérations (PGCD, PPCM, logarithme, ...etc)

1.2 Question 2.

Un nombre aléatoire cryptographiquement sûr doit garantir que tous nombres ont les même probabilités de tombés, cela permet de garantir :

- Difficulté à factoriser
- Connaître les valeurs précédentes ne permet pas de faciliter la prédiction des suivantes

La bibliothèque choisi est "secrets", elle permet de générer des nombres aléatoires cryptographiquement sûr à l'aide des fonction "bin()" et "secrets.randbits".

1.3 Question 3.

Hormis le test naîf qui consiste à tester si les seul diviseur d'un nombre sont 1 et lui même (très long), on utilise le petit théorème de fermat qui affirme que si p est premier, alors pour tout a on a :

$$a^{p-1} \equiv 1 \pmod{p}$$

Comme a est choisit au hasard, on a une une probabilité d'erreur , sachant qu'il existe des nombres composés qui passent le testent de Fermat.

La fonction "is_prime(n,methode)" de la bibliothèque gmpy2 permet d'effectuer le test de Fermat (et d'autre)

1.4 Question 4.

La librairie choisi (gmpy) implémente bien la méthode d'exponentiation modulaire avec sa fonction "gmpy2.powmod(g, a, p)". Les résultats sont identiques avec ceux de notre fonction (fichier test.txt)

1.5 Question 7

Une fonction de Hachage cryptographique sûre doit respecter ces conditions :

- Résistance aux pré-images (dur à inverser): Connaissant un haché h , il est difficile de trouver le message m tel que H(m) = h (H : fonction de hachage).
- Résistances aux secondes pré-images: Connaissant m (et donc h = H(m)), il est difficile de trouver m' tel que H(m)=H(m')
- Résistance aux collisions : Il est difficile de trouver m et m' tels que H(m)=H(m') (méthode de la borne des anniversaires)

La fonction de hachage classique et standardisée que l'on va utiliser est SHA 256 (SHA-2,SHA-3). La librairie en python que l'on utilisera pour hacher est "hashlib" avec les fonctions associées : hashlib.SHA256(M) pour hashé un message M et hashlib.digest() pout convertir un hashé h en séquence d'octets.