DM Cryptographie

Mathis GEORGEL & Joseph SCHLESINGER

Réponse aux questions

Question 1 : Quel langage de programmation avez-vous choisi ? Quelle bibliothèque permettant de gérer des nombres entiers de grande taille allez-vous utiliser ? Quelles sont les opérations implémentées dans cette bibliothèque (multiplication, addition, etc.) ?

J'ai choisi comme langage de programmation Python.

Pour gérer des nombres entiers de grande taille, on pourrait utilise gmpy2 qui permet également d'implémenter les opérations suivantes :

Addition, soustraction, multiplication, division, modulo, puissance, etc.

En somme la plupart des opérations mathématiques.

Mais pour l'utilisation qu'on en fait dans le DM, les bibliothèques par défaut en python sont suffisantes.

Question 2 : En vous aidant d'Internet, donnez la définition d'un nombre aléatoire. Selon le langage de programmation choisi, donnez le nom de la bibliothèque qui va vous permettre de générer ces nombres aléatoires.

C'est un nombre dont chaque chiffre est obtenu par tirage au sort à égalité de chances.

Avec Python, on pourrait utiliser la bibliothèque « random » qui existe par défaut et permettant de générer des nombres aléatoires mais la bibliothèque « secrets » permet de le faire en mieux car elle a des fonctions plus avancés qui assurent la production de chaîne binaire aléatoire suffisamment longue.

Question 9 : Que constatez-vous ?

Plus la taille du nombre augmente, plus le nombre moyen du nombre de répétitions augmente pour trouver un nombre probablement premier. Il y a la nécessité d'augmenter le nombre de répétitions pour maintenir la précision du test et également pour assurer une fiabilité globale du test.

Question 10 : Le test de Miller-Rabin est un test probabiliste, c'est-à-dire qu'il donne la réponse « pseudo-premier » avec une probabilité de se tromper. Cependant il existe un test qui permet de garantir si un nombre est premier ou non. En vous aidant d'Internet, pouvez-vous donner le nom de ce test et sa complexité ? Je ne demande pas de comprendre ce que fait ce test.

C'est le test de primalité AKS (Agrawal-Kayal-Saxena), d'une complexité de $O((\log n)^{6+\epsilon})$, où n est le nombre à tester et ϵ est une constante positive arbitrairement petite.