Doriane Le Cam
Paul Decanter
6 mars 2018

Rapport Cryptographie

Question 1 :

Nous avons choisi de développer le programme en C, avec la bibliothèque GMP, comme proposé dans le sujet. En effet, le langage C parait le plus adapter, car il ne nécessite pas d'application tiers, il permet d'utiliser des fonctions de bas niveau du système et l'execution du programme est rapide. En outre il permet d'utiliser la bibliothèque GMP recommandée dans le sujet pour le traitement de nombres entiers de grande taille de précision.
Cette bibliothèque inclut plus de 150 fonctions pour le traitement d'entiers signés, tel que les calculs de base (soustraction, addition, multiplication, division), la génération de grands entiers aléatoire, la division euclidienne ou encore le modulo.

Question 2 :

Un nombre aléatoire cryptographiquement sûr nécessite deux propriétés :

- Il doit être impossible, étant donné les k premiers bits d'une séquence, de trouver le (k+1)-ème bit à l'aide d'un algorithme polynomial avec un taux de succès de plus de 50%.
- Il doit être impossible, qu'a partir d'une partie du nombre, de retrouver la manière dont ce nombre a été généré par un générateur. De plus, il doit être impossible de trouver les prochains états de ce générateur.

La bibliothèque GMP permet de généré des nombres aléatoire, mais ceux-ci ne sont pas cryptographiquement sûr. Nous avons donc décidé d'utiliser la bibliothèque Sodium, qui permet de générer des nombre aléatoire sur 1024 bits.
Les tests d'intégrité de Sodium peuvent être trouvés à la page :

https://download.libsodium.org/doc/installation/