Université de Limoges

Développement Logiciel Cryptographique

TP n° 1

Exercice 1

Écrivez un programme qui lit un entier k sur la ligne de commande, puis génère et affiche des nombres aléatoires d'au plus k bits. Le programme s'arrêtera lorsque le nombre aléatoire généré sera multiple de 20 (variante : lorsque le nombre généré sera premier).

Améliorer votre programme pour que les nombres générés soient différents à chaque exécution.

Améliorer votre programme pour qu'il affiche la graine utilisée.

Améliorer votre programme pour qu'il puisse prendre en entrée une valeur de graine à laquelle initialiser le générateur (très utile pour faire du rejeu).

Exercice 2

Modifiez le programme de l'exercice précédent pour que les nombres générés aient une taille :

— exactement égale à k bits
— au plus égale à k chiffres décimaux
— exactement égale à k chiffres décimaux

Exercice 3

Le fichier *private_key.txt* contient les valeurs du module n et de l'exposant privé d d'une clé RSA de 1024 bits.

Écrivez un programme qui utilise cette clé pour déchiffrer le chiffré c contenu dans le fichier *ciphertext.txt* par la méthode du "square and multiply" de gauche à droite.

Votre programme devra comparer la valeur obtenue avec la valeur du clair m se trouvant dans le fichier *plaintext.txt*.

Variante de votre programme : utilisez la méthode d'exponentiation "square and multiply" de droite à gauche.