Sujet de travaux pratiques supplémentaire

Méthode de cryptage RSA

Description de la méthode

L'algorithme de cryptage RSA a été imaginé en 1978 par Rivest, Shamir et Adelman.

La création des clés s'effectue comme suit. On choisit deux grands nombres premiers p et q. On pose $n = pq$. On calcule l'indicatrice d'Euler $\varphi(n) = (p-1)(q-1)$. Puis, on choisit un grand entier c, premier avec $\varphi(n)$. Finalement, on calcule $d = c^{(-1)} \mod \varphi(n)$. La *clé publique*, connue de tous, est alors le couple (n, c). La *clé privée*, gardée secrète, est le couple (n, d).

Le nombre $\varphi(n)$ est inconnu. De plus, la décomposition en facteurs premiers $n = pq$, qui permettrait de retrouver $\varphi(n)$, est impossible à réaliser en temps raisonnable, les nombres premiers p et q étant très grands. Ainsi, il est impossible de retrouver la clé privée d à partir de la clé publique (n, c).

Pour crypter son message, par exemple l'entier x, plus petit que p et q, l'expéditeur effectue l'opération :

$$y = x^c \mod n.$$

Pour décrypter le message y, le destinataire effectuera :

$$z = y^d \mod n.$$

Le théorème d'Euler, $x^{\varphi(n)} = 1 \mod n$, assure que : $z = x \mod n$.

1 Création des clés

▶ Choisir deux grands nombres premiers p et q, de 4 chiffres ou plus. (Commande `nextprime`.) Calculer n et $\varphi(n)$. Choisir l'entier c, lui aussi assez grand. (Commande `igcd`.) Calculer l'entier d. (Commande `mod`.) Définir la clé publique comme la liste `[n,c]` et la clé privée comme la liste `[n,d]`.

▶ Tester les clés ainsi créées en cryptant puis décryptant un entier de 3 chiffres. (Commandes `&^` et `mod` pour une exponentiation rapide modulo n.)

2 Cryptage et décryptage d'un entier

▶ Ecrire une procédure `CryptageEntier` qui prend comme arguments un entier `EntierInitial` et une liste de 2 entiers `ClePublique`. La procédure retourne l'entier crypté.

▶ Ecrire une procédure `DecryptageEntier` qui prend comme arguments un entier `EntierCrypte` et une liste de 2 entiers `ClePrivee`. La procédure retourne l'entier décrypté.

▶ Tester les 2 procédures en cryptant puis décryptant l'entier 803.

3 Cryptage et décryptage d'une liste d'entiers

▶ Ecrire une procédure `Cryptage` qui prend comme arguments une liste d'entiers `L` et une liste de 2 entiers `ClePublique`. La procédure retourne la liste des entiers cryptés.

▶ Ecrire une procédure `Decryptage` qui prend comme arguments une liste d'entiers `L` et une liste de 2 entiers `ClePrivee`. La procédure retourne la liste des entiers décryptés.

▶ Tester les 2 procédures en cryptant puis décryptant une liste d'entiers.

4 Codage et décodage d'une chaîne de caractères

Soit un message donné sous la forme d'une chaîne de caractères ASCII. Il s'agit ici de coder ce message, c'est-à-dire de le transformer en une liste d'entiers. Il suffit de remplacer chaque caractère par son code ASCII.

Utiliser le paquetage `StringTools`. (Commande `with`.)

► Ecrire une procédure `Codage` qui prend comme argument une chaîne de caractères `C` et qui retourne la liste des codes ASCII de ses caractères. (Commande `Ord`.)

► Ecrire une procédure `Decodage` qui prend comme argument une liste d'entiers `L` et qui retourne la chaîne des caractères ASCII correspondant. (Commandes `Char` et `cat`.)

► Tester les 2 procédures en codant puis décodant une chaîne de caractères.

5 Cryptage et décryptage d'un message

► Définir une chaîne de caractères assez longue. Ne pas hésiter à employer des symboles de la table ASCII. Coder puis crypter ce message initial, grâce à la clé publique. Finalement, décrypter puis décoder le message crypté, grâce à la clé privée.

6 Regroupement par blocs de 3 caractères

Plus grands sont les entiers à crypter, plus grande est la sécurité de RSA. Les codes ASCII sont donc regroupés par bloc de trois et leurs écritures en base 10 sont concaténées. Par exemple, $[154, 101, 230]$ donne l'entier 154101230, et $[4, 1, 23]$, qui correspond à $[004, 001, 023]$, donne l'entier 4001023.

► Réécrire l'ensemble des procédures précédentes pour pouvoir fonctionner dans le cadre d'un cryptage RSA par bloc de trois caractères.

Indications :
- Il faudra redéfinir des clés plus grandes, créées à partir de nombres premiers p et q de plus de 9 chiffres.
- Il faudra modifier la procédure `Cryptage` pour qu'elle crypte les entiers par blocs de trois. Il faudra donc créer une procédure qui concatène les écritures décimale de 3 entiers.
- Il faudra au préalable préparer la liste d'entiers, pour qu'elle soit de longueur multiple de 3. (On pourra éventuellement la compléter par un ou deux zéros.)

Application : Votre clé publique est connue de tous. Votre clé privée est $n = 49444521988556447$ et $d = 14155696501478203$. Vous recevez le message suivant, crypté par blocs de 3 caractères :

$[14281766484470818, 44311361271739585, 403447230906928, 11613977548533433, 5702568047869352,$
$11762883383343029, 42438688981373315, 47096901522799288, 26165406213691979, 19679394969728398]$.

A vous de jouer !