Projet étoilé de L1

UNIVERSITÉ PARIS DESCARTES

Grands nombres premiers et système RSA

Système RSA

Q1 : Rappeler le théorème fondamental et sa démonstration qui permet la mise en place de la méthode RSA.
Q2 : Expliciter la méthode de cryptage RSA.

Q3 : Ecrire un algorithme qui prend un texte, et réalise un cryptage suivant la méthode RSA.
Puis écrire un algorithme de décryptage. (On pourra évaluer le cout de calcul de $x^k \bmod n$).

Grands nombres premiers

Q4 : Donner des familles de nombres premiers (Nombres de Mersenne, de Fermat, …)
Q5 : Donner la définition d'Euler d'un nombre pseudo-premier.
Q6 : Exposer et implémenter quelques tests de primalités (Evaluer les temps de calcul)

Bibliographie

- *Merveilleux nombres premiers* - Jean Paul Delahaye – Belin
- *Arithmétique et cryptologie* – Gilles Bailly-Maitre – Ellipses
- *Arithmétique modulaire et cryptologie* – Pierre Meunier – Cépaduès Editions
- *Arithmétique modulaire* – Jean-Pierre Lamoitier – Ellipses
- *Arithmétique pour l'informatique* – Pierre Wassef – Vuibert
- *Exercices et problèmes de cryptogrpahie* – Damien Vergnaud – Dunod

  …

Cryptage RSA - Xcas

Rappel du principe de cryptage RSA

- Chaque personne souhaitant coder ou signer un message dispose d'une clef privée, un entier s connu de lui seul, et d'une clef publique, une paire d'entiers (c,n).
- n est le produit de 2 nombres premiers p et q, et s et c sont inverses modulo $\phi(n)$, où $\phi(n)=(p-1)(q-1)$ (le nombre d'entiers de l'intervalle $[1,n[$ premiers avec n), on a alors

$$(a^c \pmod n)^s \pmod n = (a^c \pmod n)^s \pmod n = a \pmod n \qquad (1)$$

- Pour coder un message à destination d'une personne dont la clef publique est (c,n), on commence par le transformer en une suite d'entiers, On peut par exemple remplacer chaque caractère par un entier compris entre 0 et 255, son code ASCII.
- Puis on envoie la liste des nombres $b=a^c \pmod n$. En principe, seule la personne destinataire connait s et peut donc retrouver a à partir de b en calculant $b^s \pmod n$.
- On peut aussi authentifier qu'on est l'auteur d'un message en le codant avec sa clef privée, tout le monde pouvant le décoder avec la clef publique.

Quelques compléments

1 **Calcul du reste de a^n par un entier fixé m**

On distingue n pair et n impair et en utilisant pour $n>0$ pair : $a^n \pmod m = (a^{n/2} \pmod m)^2 \pmod m$ et pour $n>1$ impair : $a^n \pmod m = (a \times (a^{(n-1)/2} \pmod m)^2) \pmod m$

Récursif : si $c==0$ on renvoie 1, si $c==1$ on renvoie a, sinon, si c est pair,

On pose $b=a^{c/2} \pmod n$ (calculé par appel récursif) et on renvoie $b\times b \pmod n$, et enfin si c est impair, on pose $b=a^{(c-1)/2} \pmod n$ (calculé par appel récursif) et on renvoie $b\times b \times a \pmod n$

itératif : Initialisation $A \leftarrow a$, $b \leftarrow 1$, puis tant que $c \neq 0$

si c est impair, $b \leftarrow A \times b \pmod n$,

$c \leftarrow$ quotient euclidien de c par 2,

$A \leftarrow A\times A \pmod n$

Renvoyer b.

Remarque : on pourra utiliser `time(instruction)` pour connaitre le temps d'exécution d'une instruction.

2. Test de primalité probabiliste de Miller-Rabin

Ce test utilise le fait que si p est premier, $a^{p-1}=1 \pmod p$.

On écrit p premier différent de 2 sous la forme $2^t s$ avec s impair. Comme l'équation $x^2=1$ n'admet que 1 et -1 comme racines modulo p, pour a fixé on peut avoir soit $a^t=1 \pmod p$, sinon en prenant t fois le carré modulo p on doit prendre la valeur -1. Si le test échoue, p n'est pas premier. Si le test réussit, p n'est pas forcément premier, mais on peut montrer qu'il y a au plus 1 entier sur 4 pour lequel il réussit. On reprend donc le test pour quelques autres valeurs de a jusqu'à être raisonnablement sur que p est premier .

3. Numérisation de texte

Xcas utilise l'UTF8 pour coder des chaines de caractères. Les commandes asc et char permettent de convertir une chaine de caractères en une liste et réciproquement. On peut travailler sur la liste d'entiers obtenus, par exemple en considérant qu'il s'agit d'un seul entier écrit en base 256 (instruction convert) ou en découpant la liste en blocs de taille donnée et en générant des listes d'entiers écrits en base 256.

Exemple de programmation du cryptage RSA

Q1 Génèrer deux nombres premiers p et q>256 au hasard,

en utilisant par exemple les fonctions nextprime et rand. Calculez $n=p \times q$ puis $\varphi(n)=(p-1)(q-1)$ puis choisissez une clef secrète s inversible modulo $\varphi(n)$ et calculez son inverse c.

Q2: Codage et décodage d'un message

On transforme une chaine de caractère en une liste d'entiers et réciproquement avec asc et char. Pour l'appliquer à une liste l, on peut utiliser map(l,powmod,c,n). En utilisant la paire de clefs de Q1, coder un message puis décoder ce message pour vérifier.