

Cryptage RSA

F. Kany. ISEN-Brest & La Croix-Rouge

Position du problème

On propose de réaliser la méthode de cryptage due à Rivest, Shamir et Adleman (dite méthode RSA). Cet algorithme date de 1977, il repose sur le fait que, si l'on multiplie ensemble deux nombres premiers très grands (p et q), il est très difficile¹ de redécomposer le résultat obtenu ($n = p.q$) en facteurs premiers (p et q).

La méthode RSA, dite à clef publique, est la suivante² :

1. à chaque lettre ℓ de l'alphabet, on affecte un nombre n_ℓ compris entre 1 et n_{max} .
(Si on ne prend que les lettres majuscules non accentuées et le caractère blanc, on a $n_{max} = 27$).
 2. on transforme le texte à transmettre (formé de ℓ_{max} lettres) en un nombre : $message = \sum_{\ell=1}^{\ell_{max}} n_\ell \times n_{max}^{\ell}$.
 3. un interlocuteur 1 choisit secrètement deux nombres premiers très grands p_1 et q_1 congrus³ à 2 modulo 3. Il publie leur produit (la clef publique) : $n_1 = p_1.q_1$. Il garde secrètement le résultat d'un autre calcul (la clef privée) : $k_1 = \frac{2.(p_1-1).(q_1-1)+1}{3}$ (k_1 est forcément un entier).
- La fonction codage de x consiste à calculer une certaine fonction $f_{n_1}(x)$;
la fonction décodage de x consiste à calculer la fonction inverse : $f_{n_1,k_1}^{-1}(x)$.
4. un interlocuteur 2 fait de même avec $p_2, q_2, n_2 = p_2.q_2$ et $k_2 = \frac{2.(p_2-1).(q_2-1)+1}{3}$.
 5. lorsque 1 veut transmettre le *message* à 2, il envoie : $xxxx = f_{n_2}(f_{n_1,k_1}^{-1}(message))$.
 6. lorsque 2 veut décoder $xxxx$, il calcule $f_{n_1}(f_{k_2,n_2}^{-1}(xxxx)) = message$.
 7. le récepteur décompose le nombre *message* dans la base n_{max} et transforme chaque nombre de la décomposition en lettre de l'alphabet pour retrouver le texte.

La fonction codage $f_{n_i}(yyyy)$ consiste à décomposer $yyyy$ dans la base n_i puis à élever chaque terme de la décomposition au cube modulo n_i .

La fonction décodage $f_{n_i,k_i}^{-1}(zzzz)$ consiste à élever $zzzz$ à la puissance $3.k_i$ modulo n_i . Cette procédure utilise le petit théorème de Fermat généralisé :

- soit m et n deux nombres premiers entre eux ($m < n$) ;
- soit $\Phi(n)$ la fonction d'Euler donnant le nombre des entiers inférieurs et premiers avec n .

Ici $\Phi(n) = (p-1) * (q-1)$.

On a alors : $m^{\Phi(n)} \equiv 1 [n] \Rightarrow m^{(p-1).(q-1)} \equiv 1 [n] \Rightarrow \forall k \in \mathbb{N}, m^{k.(p-1).(q-1)} \equiv 1 [n]$

$\Rightarrow \forall k \in \mathbb{N}, m^{k.(p-1).(q-1)+1} \equiv m [n]$


Comme $p \equiv q \equiv 2 [3]$, on a : $2.(p-1).(q-1) + 1 \equiv 0 [3]$ et donc $\exists k \in \mathbb{N}$ tel que $2.(p-1).(q-1) + 1 = 3.k$

Code avec PYTHON

On a besoin d'un algorithme pour tester si un nombre est premier. On peut utiliser l'algorithme de Miller-Rabin ou bien la fonction `isprime` de la bibliothèque PYPRIME.

Téléchargez la bibliothèque PYPRIME à l'adresse :

<https://pypi.python.org/pypi/pyprimes/>

et installez le package `pyprimes-0.1.1a.tar.gz` à l'aide de WINPYTHON CONTROL PANEL  (Add packages suivi de `Install packages`).

1. Historiquement, p était un nombre de 64 chiffres et q un nombre de 65 chiffres. En 1994, le nombre $n = p.q$ de 129 chiffres (RSA129) a pu être décomposé en p et q grâce à 1600 ordinateurs travaillant en parallèle pendant deux jours.

2. Les 2 premières étapes sont communes à toutes les méthodes de codage.

3. i.e. dont le reste de la division euclidienne par 3 est 2.