TP N° 1 du Module Principes de la Cryptographie : Algorithme d'Euclide étendu et Inversion Modulaire

Master 1ière Année-S2 : WIC,RSSI,ISI

Il s'agit d'implémenter l'algorithme d'Euclide étendu qui permet de résoudre les équations diophantienne de la forme a*x + b*y=d (a,b,d sont des entiers connus , x et y sont les variables). Cette équation admet une solution si et seulement si d est un multiple du PGCD(a,b). Il est aussi montré que si d=PGCD(a,b), alors d est le plus petit entier qui peut s'écrire comme combinaison linéaire de a et b : a*x+b*y=d.
L'algorithme d'Euclide étendu permet de trouver la solution à cette équation si elle existe. Il prend en entré deux valeurs à et b, et donne en sortie les valeurs de x, y et bien sûre d. Voici une version algorithmique de cet algorithme :

Algorithme d'Euclide étendu

```
Entrée : a, b entiers (naturels)

Sortie : r entier (naturel) et  u, v entiers relatifs tels que r = pgcd(a, b) et r = a*u+b*v

Initialisation : r := a, r' := b, u := 1, v := 0, u' := 0, v' := 1

                 q  quotient entier

                 rs, us, vs  variables de stockage intermédiaires

tant que (r' ≠ 0) faire

    q := r÷r'

    rs := r, us := u, vs := v,

    r := r', u := u', v := v',

    r' := rs - q*r', u' = us - q*u', v' = vs - q*v'

   fait

renvoyer (r, u, v)
```

L'une des grand utilités de cet algorithme et qu'il permet de calculer efficacement l'inverse modulaire d'un nombre modulo un autre (si l'inverse existe). Il suffit de l'appliqué pour résoudre l'équation a*x+n*y=1 pour trouver l'inverse de a modulo n (sachant que PGCD(a,n) doit être égale à 1 pour que l'inverse existe). L'inverse de a serait tout simplement la valeur de x.

Travail à faire :

- Implémenter une class Java « BigIntMath» avec une méthode « EuclideEtendu » qui prend en entré a et b et renvoi x,y et d.
- Implémenté une autre méthode « InverseMod » qui prend en entrée a et n et renvoi l'inverse de a modulo n. La méthode affiche un message d'erreur si a n'est pas inversible modulo n, sinon elle affiche son inverse.

NB : les valeurs de a, b, x, y et n doivent être des grand entiers de la classe Java prédéfinie « Biginteger ».