Niveau : 1 **Jaune**

Inverser un nombre modulo n

1 Présentation du problème

Soit n un entier > 1. On cherche à trouver l'inverse de a modulo n s'il existe. On sait que a est inversible modulo n (c'est-à-dire dans l'anneau $\mathbb{Z}/n\mathbb{Z}$) si et seulement si a est premier avec n. Le problème est donc de calculer un représentant de la classe inverse de celle de a.

2 Application du théorème de Bezout

Si a et n sont premiers entre eux alors il existe u et v tels que $ua + vn = 1$ En prenant cette égalité modulo n on en conclut que :

$$ua \equiv 1 \quad (n).$$

En conséquence u est un représentant de la classe inverse de celle de a. Or on dispose d'un algorithme très efficace pour calculer u, c'est l'algorithme d'Euclide étendu. On a donc un moyen pratique de résoudre le problème posé, même pour des nombres de taille élevée, ayant plusieurs milliers de bits.

3 Application d'un calcul de puissance

Si on connaît $\Phi(n)$ (fonction d'Euler) alors on peut raisonner de la façon suivante : on sait que $a^{\Phi(n)} = 1$ dans le groupe $(\mathbb{Z}/n\mathbb{Z})^*$ des éléments inversibles de $\mathbb{Z}/n\mathbb{Z}$. En conséquence l'inverse de a est $a^{\Phi(n)-1}$. On calcule alors cette dernière puissance par l'algorithme square and multiply qui est aussi très efficace.

4 Comparaison des deux méthodes

Evidemment, la deuxième méthode suppose la connaissance de $\Phi(n)$ et lorsqu'on ne dispose pas de la factorisation de n on n'a pas accès à cette valeur. Mais ce n'est pas la seule raison qui en limite l'utilisation. À première vue, le nombre de boucles effectuées par chacun des deux algorithmes est du même ordre (linéaire en la taille de n). Cependant si on analyse chacun des algorithmes au niveau du calcul sur les bits, on se rend compte que l'algorithme d'Euclide étendu est un peu plus rapide que l'algorithme utilisant l'exponentiation. On utilise donc en général l'algorithme d'Euclide étendu pour calculer un inverse modulo n.

Auteur : Ainigmatias Cruptos
Diffusé par l'Association ACrypTA