MATH EN INFO

FICHE EXPLICATION Nº 15

Inverse modulaire

1 Descriptif

L'objectif de ce défi est d'implémenter une méthode calculant l'inverse du nombre binaire courant modulo un nombre binaire donné en paramètre. On rappelle que b est l'inverse de a modulo N si et seulement si $ab = 1$ modulo N.

Le calcul de l'inverse modulaire est réalisé par l'algorithme d'Euclide étendu (une version "améliorée" de l'algorithme du PGCD). Cet algorithme est usuellement facilement implémentation. Hélas il nécessite d'utiliser temporairement des nombres négatifs, ce que notre structure de donnée ne permet pas. Il est possible de faire quelques modifications pour éviter l'utilisation des nombres négatifs mais ces modifications sont assez délicates. C'est pourquoi cette méthode est déjà implémentée !

2 Protocole

1. Une fois la connexion établie, le serveur commence par envoyer un premier message annonçant le début du défi :

   -- Debut du defi : Inverse Modulaire --

   Ce message n'attend pas de réponse.
2. Le serveur envoie ensuite une série de nombres binaires (de taille aléatoire) deux par deux.
3. Pour chaque paire de nombres binaires, le serveur doit recevoir en retour un nombre binaire (sous forme binaire) égal à l'inverse du premier nombre modulo le deuxième.
4. Après chaque réponse, le serveur enverra un message commençant par "OK" ou "NOK" suivant si la réponse est correcte ou non.
5. A la fin du défi, le serveur enverra un message indiquant "Defi valide" ou "Defi echoue !". Aucune réponse n'est attendue.
6. Le serveur terminera la communication par le message "FIN", votre client devra alors fermer la socket. Aucune réponse n'est attendue.

3 Exemple de communication

Voici un exemple (incomplet) d'une communication pour ce défi. Dans cet exemple les "<" et ">" indiquent le sens de transfert de chaque message et ne doivent pas être présents dans la communication.

< -- Debut du defi : Inverse Modulaire --
< 1101001010
< 1001000101101
> 11000110010
< OK
< 1110001
< 11011110010000001
> 1101110001001001
< OK
< 10110010000011
< 110101100
> 100111111
< OK