MATH EN INFO

Générer la clé publique

FICHE EXPLICATION Nº 24

1 Descriptif

L'objectif de ce défi est d'implémenter la méthode permettant de générer la clé publique de l'algorithme de RSA. Pour ce défi, on pourra ajouter des getters aux attributs P, Q, N, phi et e de la classe GenerateurDeClesRSA. Pour générer une clé publique RSA, il faut :

- Choisir deux grands nombres premiers distincts P et Q.
- Calculer $N = PQ$.
- Calculer $Phi = (P-1)(Q-1)$.
- Choisir un nombre e premier avec ϕ.

2 Protocole

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

   -- Debut du defi : Generer cle publique --

   Ce message n'attend pas de réponse.
2. Vous devez envoyer une série de quintuplet (P,Q,N,Phi,e) de nombres binaires.
3. Pour chaque quintuplet, le serveur enverra un message commençant par "OK" ou "NOK" suivant si celui-ci convient ou non.
4. A la fin du défi, le serveur enverra un message indiquant "Defi valide" ou "Defi echoue !". Aucune réponse n'est attendue.
5. 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 : Generer cle publique --
> 10010000010101011011011010111101100100110010000000100011100100 11
> 11101101010110111000000000111010001100001000110110001011011000 11
> 10000101110100101111000011110001001000001100001000011101111011 10...
> 10000101110100101111000011110001001000001100001000011101111011 01...
> 10101011010111110101100110000011100101000011011010011011001001 01...
< OK
> 10111110000000110010000101101011001010010111110000111011101100 11
> 11000111100100100010100000110110110101010001000110101010011101 11
> 10010100001000001110101010011100100011000110010010111101111000 10...
> 10010100001000001110101010011100100011000110010010111101111000 00...
> 11011110101010111011101010110100100000001001100101011011101110 00...
< OK
> 11110100101101011001011000111101101101101100111001100001010111 11
> 11001011010011101001011001110100001110001000000100101110100010 11
> 11000010010101110001110101001000001101010001011011001100000001 11...
> 11000010010101110001110101001000001101010001011011001100000001 01...
> 11101000101110000100110111101101011101010101100000011001101110 00...
< OK