MATH EN INFO

Générer la clé privée

FICHE EXPLICATION Nº 20

1 Descriptif

L'objectif de ce défi est d'implémenter la méthode permettant de générer la clé privée de l'algorithme de RSA. Pour ce défi, on pourra ajouter un setter aux attributs P, Q, phi et e de la classe GenerateurDeClesRSA. On rappelle que la clé privée d est l'inverse de e modulo phi et que $phi = (P-1)(Q-1)$.

2 Protocole

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

   – Début du défi : Generer cle privee –

   Ce message n'attend pas de réponse.
2. Le serveur envoie ensuite une série de triple (P,Q,e) où P,Q et e sont des nombres binaires.
3. Pour chaque triplet (P,Q,e), le serveur doit recevoir en retour la clé privée d associée d'un MotBinaire de la taille de la clé.
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 "Défi validé" ou "Défi échoué !". 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.

< -- Début du défi : Generer cle privee --
< 11101111001111100010110001101100111111110111001100010001011100 01
< 11110001101011001010100010110011000000100010000000100101111000 11
< 10100000111101111001010110010010111010010101001110010100100101 11010011 10...
> 10001000001111100001001101000100000101011011011011110000100100010 01011111...
< OK
< 10101000001100011100000101100000101100111001010000000001001100 01
< 11000110111011000001100110000010001011010010100111110110111110 11
< 10100001101110001011000011001101011100000101111010111111011101 01111101011...
> 00000110000001010111000101110010001101011010011010011001001111 00111001111...
< OK
< 11110011010000010111100001011111101101110011100011000001110000 01
< 10110100011010101111001110100111000011110001011011101011010111 11
< 10101100010101110000111100000110100000010100110110111100000111 01000011111...
> 01000010011011001100111001101100111010011000010111100110000100 11101110 10...
< OK