TP1

Implémentation de RSA : forces et faiblesse

Remarque : Ce TP a pour but de vous permettre de bien intégrer l'algorithme RSA en l'implémentant vous-même.

Le but du TP est de mettre en œuvre le chiffrement par clé publique/privée RSA. Nous devrons effectuer un certain nombre d'opérations arithmétiques, notamment un calcul de pgcd, un test de primalité, une inversion modulaire et une exponentiation modulaire. Encore une fois, je vous recommande fortement de tester ces opérations une par une au fur et à mesure que vous les écrivez.

Le TP se fera en trois étapes :

1. **Implémentation** :
    a. Écrire un programme permettant de générer un couple clé privée/clé en fonction de la taille de la clé (modulo n) en bits.
    Rappel : une clé de taille k bits implique que la valeur de q ou p devrait être inférieure à $2^{K/2}$.
    b. Écrire un programme permettant de (dé)chiffrer un nombre selon l'algorithme RSA. Dans un premier temps, ce nombre pourra être entré en dur dans le code ou généré aléatoirement.
2. **Test avec une clé de taille petite** :
    a. Essayez de décrypter les messages suivants, où seules les **clés publiques** sont données.
    Le message : [9197, 6284, 12836, 8709, 4584, 10239, 11553, 4584, 7008, 12523, 9862, 356, 5356, 1159, 10280, 12523, 7506, 6311] a été crypté avec la clé publique **(e= 12413 ; n=13289)**.

    Rappel : Il faut faire la factorisation pour trouver q et p et calculer l'inverse de e modulo phi(n).

    b. Mesurer la complexité de la factorisation en fonction de la taille du modulo n (en bits).

    c. Attention, celui-ci est ardu. Clé publique: (e=163119273;n=755918011); message crypté : [671828605, 407505023, 288441355, 679172842, 180261802]

3. **RSA et Codage :** Le but de cette partie est de chiffrer un texte (alphabet) par RSA. Pour cela il est nécessaire de transformer le texte en chiffre avant de le chiffrer (**voir Exercice 3 TD2**).

    a. Générer une paire clé privée/clé publique basée sur la taille de la clé (modulo n) en

bits.

b. On choisit un alphabet à 40 lettres A= 0, B= 1, C= 2, D= 3, E= 4, F= 5, ... , X= 23, Y= 24, Z= 25, _=26, .= 27, ?= 28, €= 29, 0= 30, 1= 31, . . . , 8= 38, 9= 39.
   Pour un message en clair, on le découpe en blocs avant de le chiffrer. Calculez la taille des blocs pour le message en clair ?
   Rappel : La taille du bloc est calculée en fonction du nombre de caractères utilisé (40 dans notre cas) et le modulo n.
c. Utilisez le codage approprié pour transformer le texte en chiffres.
d. Chiffrer le message obtenu par la clé publique.
e. Transformer le message obtenu en texte alphabet (**fait attention à la taille du bloc pour le message chiffré**).
f. Refaire les mêmes étapes pour le déchiffrement.
g. Bonus : Adapter votre programme si on utilise le codage binaire.

4. Signature et Vérification
   a. Générer une paire de clés RSA de taille 32 bits
   b. Hacher un fichier avec SHA-256.
   c. Signer le hash du fichier avec la clé privée.
   d. Vérifier la signature avec la clé publique.