**Rapport TP RSA**



**Plan :**

* Introduction
* Implémentation
* Taille petite
* Chiffrement d’un texte
* Tester les failles RSA
* Conclusion

**Réaliser par : Khenifra Anis**

**Introduction :**

Le but du TP est d’implémenter le cryptage à clé publique RSA. On va avoir besoin d’effectuer un certain nombre d’opérations arithmétiques, notamment un calcul du pgcd, un test de primalité, une inversion modulaire et une exponentiation modulaire.

**Implémentation :**

**Test local**

Pour l’implémentation de l’algorithme RSA en local, nous aurons p et q nombre premiers qui seront données (avec vérification de primalité avec la fonction isPrime) pour que le chiffrement soit réaliser nous calculerons d’abord n puis phi de n afin de déduire e de n qui doit être premier avec phi de n (avec la fonction pgcd(e,phiden)=1), grâce à la clé public Kp=(e,n) nous pouvons chiffrer le message avec la formule : C= Me mod n

Pour le déchiffrer il faut trouver la clé privé Kpr=(d,n) qui s’obtient très difficilement a partir de la clé publique, et de ce fait on doit déterminé d grâce à l’algorithme d’Euclide étendu. Et une fois d trouvé on peut chiffrer grâce à la formule :

M=cd mod n=med  mod n

**Test distant :**

On donne notre clé public a un camarade pour qu’il crypte avec un nombre et nous l’envois pour le décrypté, le nombre est le même.

**Taille petite :**

Pour décrypté les messages suivant sans que les clés publiques soit connues on doit donner n pour trouver p et q grâce à la fonction de factorisation puis calculer d pour déchiffrer les messages.

**Chiffrement d’un texte :**

Pour chiffrer un texte nous utiliseront la même méthode que précédemment pour chiffrer un nombre avec p et q qui seront donné.

Nous allons le chiffrer en block avec une boucle qui sera inférieur à la taille du mot et pour les block une boucle qui prend la taille du block ensuite nous convertirons chaque caractère ascii et nous le chiffrerons.

**Tester les failles RSA :**

Il existe plusieurs failles comme p et q trop petit pourrait facilement être trouvé mais il existe aussi les Attaque ou l’on essaye de deviner le texte clair (en asymétrique) ou l’attaquant dispose d’un texte chiffré et choisit un texte clair. Il chiffre ce texte clair à l’aide de la clé publique de la victime, dans le cas d’un algorithme à clé publique. En comparant le résultat avec le texte chiffré dont il dispose, il peut déterminer si sa supposition était correcte ou aussi la méthode de Cryptanalyse différentielle qui est une attaque à texte clair choisi, initialement faite sur le DES. On choisit deux textes clairs avec des différences connues (l’opérateur étant le XOR), puis on analyse l’évolution des différences entre les textes chiffrés avec une même clé à chaque ronde. En regardant les différences finales entre les paires de textes chiffrés on donne diverses probabilités aux clés testées. En affinant, on arrive à la clé la plus probable.

On a également la Cryptanalyse linéaire qui utilise des approximations linéaires pour décrire l’action d’un algorithme de chiffrement par blocs". La conception du DES ne le protège pas particulièrement contre cette attaque, encore plus récente que la différentielle.

**Conclusion :**

Dans ce tp nous avons vu que la fiabilité du RSA réside sur la complexité et non pas sur l’impossibilité de le décrypté c’est pour ça qu’il existe plusieurs failles.