Maxime Cordier
mcordier@enssat.fr

Module :
Cryptographie

Promotion :
FISA IAI 2

Devoir maison 1 « Chiffrement RSA »

ENSSAT LANNION | Université de Rennes

1 Introduction

L'objectif de ce compte rendus est de présenter les étapes de confection d'un chiffrement RSA. Le chiffrement RSA (Rivest, Shamir, Adleman) est un algorithme de cryptographie asymétrique. Il est basé sur le fait que la clé de codage ne peut pas permettre de trouver la clé de décodage. On utilise une clé publique pour chiffrer et une clé privée pour déchiffrer.

2 Fonctions de chiffrement et de déchiffrement

Deux fonctions ont été implémentées en Python pour chiffrer et déchiffrer un message.

- chiffrement_RSA(message, cle_publique) : Le message M a coder est le premier argument de la fonction. On a ensuite la clé publique qui est un tuple de deux entiers (e, n). Cette fonction retourne un entier X subissant le traitement suivant : $X = M^e[n]$. Afin de réaliser le calcul exponentiel, nous utilisons une fonction nommée exp_rapide prenant en paramètres M et e.

- dechiffrement_RSA(message, cle_publique, p, q) : Le message X à décoder est le premier argument de la fonction. On a ensuite la clé publique qui est toujours le tuple de deux entiers (e, n) et les deux nombres premiers p et q. Les nombres p et q sont nécessaires pour obtenir la clé privée D permettant de déchiffrer le message.

Le calcul pour déchiffrer X et ré-obtenir M est le suivant : $M = X^D[n]$. Afin de réaliser le calcul exponentiel, nous utilisons une fonction nommée exp_rapide prenant en paramètres X et D.

Pour obtenir la clé privée D, nous suivons 3 étapes :
- Calcul de $\varphi(n)$: $\varphi(n) = (p-1)(q-1)$
- Calcul des coefficients de Bézout entre $\varphi(n)$ et e : Les coéfficients de Bézout sont les coefficients u et v tels que $u * \varphi(n) + v * e = 1$. Pour obtenir ces coefficients nous utilisons une fonction appellée bezout prenant en paramètre $\varphi(n)$ et e.
- Calcul de D : $D = v[\varphi(n)]$. D est aussi appelé l'inverse modulaire de e modulo $\varphi(n)$.

3 Fonction exp_rapide

Notre fonction *exp_rapide* prend en paramètres deux entiers a et b. Cette fonction a pour but de retourner le résulat tel que a^b. La première façon de calculer cette puissance est de multiplier n par lui-même p fois. Cependant, la méthode de l'exponentiation rapide est plus simple. Elle permet d'obtenir le résultat a^b via un algorithme récursif. Le voici :
- Si b = 0, alors on retourne 1.
- Si b est pair, alors on retourne exp_rapide($a * a$, $\frac{b}{2}$).
- Si b est impair, alors on retourne a *exp_rapide($a * a$, $\frac{b-1}{2}$).

4 Fonction bezout

La fonction *bezout* prend en paramètres deux entiers a et b. Cette fonction a pour but de retourner les coefficients de Bézout u et v tels que $u * a + v * b = 1$ et le d le PGCD de a et b. Pour cela, nous utilisons un algorithme récursif. Le voici :
- Si b = 0, alors on retourne (1, 0, a).
- Sinon, on retourne $\left(v, u - \left(\frac{a}{b}\right) * v, d\right)$ avec (u, v, d) = bezout(b, $a[b]$).