Université de Limoges

Développement Logiciel Cryptographique

TP n° 2

1 Fonctions du cryptosystème RSA en mode standard

1.1 Génération de clé

Écrivez un programme qui prend en entrée deux entiers k et e, et qui génère une clé RSA (en mode standard) de k bits avec e comme exposant public.

Vous devrez veillez à ce que le module n soit le produit de deux premiers différents p et q de tailles sensiblement égales, et qu'il soit lui-même de taille exactement k bits.

La sortie standard de votre programme devra afficher la clé en hexadécimal sur plusieurs lignes sous la forme suivante :

```
e = 0x......
n = 0x......
d = 0x......
```

Remarque 1 *Pour que l'inverse de e modulo $\phi(n)$ existe, il faut que e et $\phi(n)$ soient premiers entre eux.*

1.2 Chiffrement et déchiffrement

Écrivez une fonction de chiffrement `encrypt_rsa`(c, m, n, e) qui prend en entrée les différents éléments d'une clé publique RSA et un entier $m \in \mathbb{Z}_n$, et qui calcule la valeur du chiffré de m, soit $c = m^e \bmod n$.

Écrivez une fonction de déchiffrement `decrypt_rsa`(m, c, n, d) qui prend en entrée les différents éléments d'une clé privée RSA en mode standard et un entier $c \in \mathbb{Z}_n$, et qui calcule la valeur du déchiffré de c, soit $m = c^d \bmod n$.

Écrivez un programme qui permet de chiffrer ou déchiffrer un texte M (qui sera converti en un entier $m \in \mathbb{Z}_n$) plutôt qu'un entier n'ayant pas de signification évidente.

1.3 Signature et vérification

Écrivez un programme qui prend en entrée un nom de fichier à signer et un nom de fichier contenant une clé privée RSA en mode standard et qui calcule la signature s du fichier sous la forme :

$$s = h^d \bmod n$$

où h est l'entier représentant le haché MD5 du fichier.

2 Fonctions du cryptosystème RSA en mode CRT

Répondre aux mêmes consignes que celles de toute la section 1 mais en générant et utilisant une clé RSA en mode CRT.