Cryptographie avancée - TP3

Anne Garcia-Sanchez

M2i - CFA CCI Avignon, 15 octobre 2024

Arithmétique pour les chiffrements asymétriques: exponentiation modulaire

Soient des entiers a et b et un entier non nul n, l'exponentiation modulaire est définie par :

$c \equiv a^b \pmod{n}$ avec $0 \leq c < n$

Pour calculer $a^b \mod n$, on peut faire $b-1$ multiplications mais il existe un algorithme bien plus efficace.

1 algorithme d'exponentiation binaire / rapide (*Square and multiply)*

L'idée de l'exponentiation binaire est de décomposer b en carrés successifs.

Exemple: $a^{11} = a \times a^{10} = a \times (a^5)^2 = a \times (a \times a^4)^2 = a \times (a \times (a^2)^2)^2$

Dans ce cas le calcul de a^{11} nécessite 5 multiplications au lieu de 10 comme précédemment.

Les calculs sont effectués modulo n ce qui permet de ne pas manipuler de nombres trop grands.

Algorithme d'exponentiation rapide:

Entrée: entiers a, b, n
avec l'écriture binaire de b: $b_m.2^m + ... + b_1.2^1 + b_0.2^0$

Sortie: $a^b \pmod{n}$

$result \leftarrow 1$
pour i de m à 0 **faire**

$result = result^2$ modulo n

si $b_i = 1$ **alors**

result $\leftarrow$ (result $\times$ a) modulo n

fin si

fin pour
Renvoyer $result$

1. Programmer la fonction d'exponentiation binaire.

2. Retrouver les résultats avec la fonction `pow`.

3. Bonus. Compléter votre programme pour afficher le nombre de multiplications effectuées.

2 Calculs

On veut calculer $42^{12345678} \pmod{99}$

- Effectuer le calcul en utilisant les opérateurs Python ** et %
- Effectuer le calcul avec votre fonction.
- Effectuer le calcul avec la fonction `pow`

Que remarque-t-on ?

3 Bonus: chiffrement RSA

On rappelle le principe du chiffrement RSA.
La clé publique est donnée par deux entiers **N** et **e** et la clé privée est donnée par un entier **d**. On a :

- chiffrement de l'entier m: calcul du chiffré $c \equiv m^e \pmod{N}$
  avec m et c positifs et inférieurs à N

- déchiffrement du chiffré : $m \equiv c^d \pmod{N}$

Dans le fichier joint, on donne la clé publique N, e et la clé privée associée.

Déchiffrer le nombre chiffré: c'est la clé qui vous permettra de déchiffrer le message suivant chiffré avec un AES en mode ECB:

```
b'\x14M,C\xd7\x82(\xa1q\xe6\xf5j\xc3\x07C\xa4'
```

4 Bonus: RSA toujours

Résoudre le challenge SMIC(1) sur `hackropole`