Exercices
Semaine 16

Cours Turing

1 Protocole RSA

Dans cet exercice, vous allez appliquer le protocole RSA pour découvrir concrètement comment celui-ci fonctionne. Le protocole est le suivant :

1. Bob génère tout d'abord deux grands nombres premiers P et Q, puis calcule $N = P \cdot Q$ et publie ce nombre N.

2. Puis il choisit un nombre C compris entre 2 et $N-1$ tel que PGDC$(C, \phi(N)) = 1$ et publie ce nombre C.

Notes : - Pour rappel, $\phi(N)$ est la fonction d'Euler, que Bob peut calculer ici facilement au moyen de la formule $\phi(N) = (P-1) \cdot (Q-1)$.

- Pour calculer le PGDC$(C, \phi(N))$ (afin de vérifier que celui-ci vaut bien 1), vous pouvez utiliser l'algorithme d'Euclide :

```
def gcd(a,b):
    if a==0:
        return b
    return gcd(b % a,a)
```

3. Bob calcule ensuite le nombre D secret tel que $C \cdot D \pmod{\phi(N)} = 1$.

Note : D est donc l'inverse de C modulo $\phi(N)$. Pour le calculer, vous pouvez utiliser l'algorithme d'Euclide *étendu*, qui étant donné des nombres entiers a et b, retourne trois nombres : gcd, x, y tels que $a \cdot x + b \cdot y = 1$ (et gcd est le pgdc de a et b) :

```
def extended_gcd(a,b):
    if a==0:
        return b, 0, 1
    gcd, x, y = extended_gcd(b % a,a)
    return gcd, y - (b // a) * x, x
```

L'algorithme extended_gcd avec les nombres C et $\phi(N)$ en entrée sort donc les nombres G, U, V tels que $G =$ PGDC$(C, \phi(N))$ et $C \cdot U + \phi(N) \cdot V = 1$; $D = U \pmod{\phi(N)}$ est le nombre qui vous intéresse !

4. Pour chiffrer le message X (un nombre compris entre 2 et $N-1$) qu'elle veut envoyer secrètement à Bob, Alice calcule $Y = X^C \pmod N$ et envoie Y à Bob.

5. Bob calcule à son tour $Y^D \pmod N$ qui est censé être égal à X (à vérifier en pratique !).

(à noter que Bob est le seul à pouvoir retrouver X, car il est le seul à connaître la valeur de D)

2 Chiffrement d'El Gamal (exercice à faire sur papier uniquement)

Ce protocole diffère légèrement du protocole RSA, mais permet également à Alice d'envoyer directement un message chiffré à Bob, sans devoir procéder à un échange de clé au préalable, comme c'est la cas avec le protocole de Diffie-Hellman-Merkle.

- Bob choisit tout d'abord un grand nombre premier N, ainsi que deux nombres entiers M et B_1 compris entre 2 et $N-1$; il calcule ensuite $B_2 = M^{B_1} \pmod N$, et publie N, M et B_2.

- Pour communiquer un message X (on supposera ici que le message X peut être représenté par un nombre entre 1 et $N-1$) à Bob, Alice tire au hasard un nombre K entre 2 et $N-1$ et envoie à Bob le message chiffré suivant, composé de deux parties :

$$(M^K \pmod N),\ X \cdot B_2^K \pmod N))$$

a) Pour déchiffrer le message envoyé par Alice, que fait Bob ? A vous de jouer !

b) Et si Eve intercepte le message d'Alice, peut-elle le déchiffrer ?

3 Signature digitale

Dans cet exercice, vous allez appliquer une technique dérivée du protocole RSA qui sert à authentifier un message. Le protocole est le suivant :

1. Alice génère tout d'abord deux grands nombres premiers P et Q, puis calcule $N = P \cdot Q$ et publie ce nombre N.

2. Puis elle choisit un nombre C secret compris entre 2 et $N-1$ tel que $\mathrm{PGDC}(C, \phi(N)) = 1$.

3. Alice calcule ensuite le nombre D tel que $C \cdot D \pmod{\phi(N)} = 1$ et publie ce nombre D.

4. Alice utilise ensuite une *fonction de hachage* H sur le message X qu'elle désire envoyer (et qu'on suppose ici être un nombre compris entre 1 et $N-1$). Vous avez bien sûr le libre choix pour la fonction de hachage H, mais la fonction simple suivante peut convenir :

$$H(X) = X^X \pmod{1'001}$$

5. Alice calcule enfin $S = H(X)^C \pmod N$ et envoie X et S à Bob.

6. Pour finir, Bob effectue la vérification suivante : il calcule

$$S^D \text{ (mod N)}$$

et compare ce nombre à $H(X)$: si ces deux nombres sont égaux, c'est la preuve que le message X provient bien d'Alice.