Université Claude Bernard Lyon 1
Institut de Science financière et d'Assurances (ISFA)

50 avenue Tony Garnier
69007 Lyon, France

Master 1 informatique
Université Claude Bernard Lyon 1

Cryptologie : TP n° 2

RSA

L'objectif de cette question est d'implanter le système cryptographique à clé publique le plus répandu, à savoir RSA.

Exercice 1 (RSA)**.** Vous utiliserez la bibliothèque Sympy de Python qui contient en particulier toutes les méthodes nécessaires pour implanter la cryptographie standard (exponentiation modulaire, test de primalité, génération de grands entiers aléatoires,...). Dans un prochain TP, vous implanterez vous-même certaines méthodes, en particulier l'exponentiation modulaire.

Vous allez créer une classe RSA. Elle possèdera une méthode KeyGen, qui prendra en entrée un paramètre λ qui définira la taille des clés, et produira la paire de clés, une méthode Encrypt, qui prendra en entrée un message clair et une clé publique, et une méthode Decrypt qui prendra en entrée un chiffré et une clé secrète.

Pour implanter RSA, vous allez devoir générer deux grands nombres premiers p et q, de même taille $\lambda/2$, et dont le produit sera égal à N (et donc de taille λ bits), deux entiers e et d inverses l'un de l'autre modulo $(p-1) \times (q-1)$. La clé publique est alors le couple (e, N), et la clé secrèté est (d, p, q). Pour rappel, le chiffrement se fait en calculant $m^e \pmod N$, et le déchiffrement en calculant $c^d \pmod N$.

Échanger avec vos collègues des clés publiques et des messages chiffrés pour tester vos programmes.

Elgamal

L'objectif de cet exercice est d'implanter un des cryptographiques à clé publique les plus répandus après RSA, à savoir le chiffrement Elgamal. Il est en particulier utilisé dans les protocoles de votes électroniques.

Vous allez créer une classe Elgamal. Elle possèdera une méthode KeyGen, qui prendra en entrée un paramètre λ qui définira la taille des clés, et produira la paire de clés, une méthode Encrypt, qui prendra en entrée un message clair et une clé publique, et une méthode Decrypt qui prendra en entrée un chiffré et une clé secrète.

Elgamal.

Vous travaillerez dans (un sous-groupe de) $(\mathbb{Z}/p\mathbb{Z})^\star$, avec un premier p de la forme $p = 2p' + 1$ où p' est (probablement) premier. Construisez un tel entier.

Je rappelle que l'ordre d'un élément g de $(\mathbb{Z}/p\mathbb{Z})^\star$ est le plus petit entier r tel que $g^r = 1 \mod p$.

Commencez par créer une méthode ordre() qui calcule l'ordre d'un élément de $(\mathbb{Z}/p\mathbb{Z})^\star$.

Tirez aléatoirement plusieurs entiers $g \in_R (\mathbb{Z}/p\mathbb{Z})^*$. À chaque fois, Calculez g^2, $g^{p'}$ et $g^{2p'}$. Que pouvez-vous dire des résultats ? Vous en déduirez une méthode alternative eg_ordre() qui calcule l'ordre multiplicatif des éléments de groupe dans ce cas particulier.

Une fois construit un élément g d'ordre p', vous génèrerez un entier x compris entre 1 et p' qui constituera la clé secrète, puis vous calculerez la clé publique $h = g^x$.

Le chiffrement Elgamal fonctionne de la façon suivante : pour chiffrer $m \in \mathbb{Z}/p\mathbb{Z}$, tirer r aléatoirement dans $[1, p-1]$, et produire comme chiffré le couple $(g^r, m \cdot h^r)$, où h est la clé publique. Comment déchiffre-t-on ?

Échanger avec vos collègues des clés publiques et des messages chiffrés pour tester vos programmes.