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

1 GPG

1. Installez GnuPG sur votre machine. Il existe une version pour chaque système d'exploitation. Ce logiciel libre est disponible ici : `https://www.gnupg.org/index.html`.
2. Explorez la documentation.
3. Explorez les commandes de `gpg` avec l'option `-h`.
4. Générez une paire de clés de taille raisonnable. Vérifiez la création des clés *via* les commandes `list` de `gpg`.
5. Préparez un certificat de révocation.
6. Extrayez votre clé publique et échangez-les vous par mail.
7. Importez alors les clés.
8. Vérifiez les clés et signez-les avec votre propre clé secréte (retestez les commandes `list`).
9. Chiffrez des fichiers. Déchiffrez-les. (Vous pouvez préciser plusieurs destinataires)
10. Expérimentez les différents commandes de signatures.
11. Examinez et éditez les valeurs de confiances des clés de votre trousseau.

Chiffrer et signer des emails

Des logiciels permettent d'intégrer la cryptographie pour protéger ses emails. Si vous utilisez Thunderbird, vous pouvez sécurisez vos envois d'emails avec Enigmail disponible ici :
`https://addons.mozilla.org/fr/thunderbird/addon/enigmail/`

Il existe des solutions pour d'autres mailers, et il existe des plugins pour les navigateurs, compatible avec gmail : cherchez-les par vous-même.

Expérimentez ces logiciels, une fois installés.

2 Chiffrement Elgamal

Comme vous l'avez constaté, le chiffrement Elgamal est implanté dans `gpg`. Vous allez, dans ce TP, programmer votre propre version du chiffrement Elgamal, en java, en utilisant la bibliothèque BigInteger de gestion des grands nombres.

2.1 Génération de grand nombres premiers

Vous allez créeer la classe Elgamal. Le chiffrement Elgamal utilise un groupe cyclique $(\mathbb{Z}/p\mathbb{Z})^*$, avec p premier. Nous utiliserons des premiers de la forme $p = 2p' + 1$ où p' est premier.

1. Quel est, dans ce cas, le cardinal de $(\mathbb{Z}/p\mathbb{Z})^*$? Quels sont les ordres possibles des éléments de $(\mathbb{Z}/p\mathbb{Z})^*$? Quelle est la taille de p par rapport à la taille de p' ?
2. Écrire dans Elgamal une méthode de classe ayant pour signature `BigInteger getPrime(int nb_bits, int certainty, Random prg)` renvoyant un premier représenté sur `nb_bits` et de la forme $p = 2p'+1$ où p' est premier avec probabilité `certainty`.
3. Écrire dans Elgamal un programme principal qui tire aléatoirement un grand premier (par exemple 2048 bits) p de la forme $p = 2p' + 1$ à l'aide de la méthode `getPrime` ci-dessus puis tire aléatoirement un entier $g \in_R (\mathbb{Z}/p\mathbb{Z})^*$ et calcule g^2, $g^{p'}$ et $g^{2p'}$. Que pouvez-vous dire de ces entiers ?
4. Écrire une méthode `ordre()` qui calcule, par recherche exhaustive, l'ordre multiplicatif des éléments de groupe (de type BigInteger).
5. Enumérez dans un fichier texte les ordres multiplicatifs de chacun des éléments de $(\mathbb{Z}/23\mathbb{Z})^*$ et de $(\mathbb{Z}/21\mathbb{Z})^*$. Commentez.
   Tester la méthode `ordre()` sur le même générateur g précedent. Que constatez-vous ?
6. Écrire une méthode `ordre_p()` dédié à ce type de groupe.
7. Écrire une méthode `randNum` dont la signature est `BigInteger randNum(BigInteger N, Random prg)` et qui génére uniformément un entier entre 0 et $N - 1$.

2.2 Implémentation

Vous êtes libres de gérer la partie génie logicielle par vous-même. Pensez néanmoins qu'un protocole de chiffrement est toujours constitué d'un algorithme de génération de clés, d'une méthode de chiffrement et d'une méthode de déchiffrement, dont les entrées sont toujours les mêmes.

1. La méthode de génération de clés Elgamal fonctionne de la façon suivante :
   (a) tirer aléatoirement un premier p à l'aide de la méthode `getPrime`. Celui-ci est de la forme $p = 2p'+1$ avec p' premier,
   (b) tirer aléatoirement un élément de $(\mathbb{Z}/p\mathbb{Z})^*$ d'ordre p' ;
   (c) tirer aléatoirement un entier x de $[0, p' - 1]$;
   (d) calculer $h = g^x$ dans $\mathbb{Z}/p\mathbb{Z}$;
   (e) la clef secrète est le couple (p, x), la clef publique est (p, g, h).
2. Le chiffrement 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 ?
3. Implantez la méthode de déchiffrement.