Introduction à la cryptographie : Implémentation algo de chiffrement d'El Gamal

Romain Duc, Alexandre Serratore

6 mars 2022

Question 1 :
Quel langage de programmation avez-vous choisi ? Quelle bibliothèque permettant de gérer des nombres entiers de grande taille allez-vous utiliser ? Quelles sont les opérations implémentées dans cette bibliothèque (multiplication, addition,...) ?

Réponse :

Nous avons choisi d'utiliser Java. Le langage Java nous a paru l'option la plus adaptée, car il dispose de bibliothèques mathématiques intégrées permettant d'effectuer des opérations complexes, e.g. l'exponentiation modulaire. La classe *java.math.BigInteger* permet de gérer des nombres entiers de grande taille.
Il est possible d'utiliser cette classe pour effectuer des soustractions, des modulos, des multiplications sur les entiers. La classe BigInteger permet de représenter des entiers sans les limitations de taille des types primitifs entiers, ou des enveloppes des types primitifs. La classe BigInteger offre en plus des opérations habituelles sur les entiers, des opérations de calcul en arithmétique modulaire, calcul du pgcd, génération de nombre premier, test pour savoir si un entier est premier, etc...

Question 2 :
De plus, pour générer les valeurs (très grandes elles aussi), x et r, il faut être capable de générer des nombres aléatoires de très bonne qualité et dits cryptographiquement sûrs.
En vous aidant d'internet, donnez la définition d'un nombre aléatoire cryptographiquement sûr. Selon le langage de programmation choisi, donnez le nom de la bibliothèque qui va vous permettre de générer ces nombres aléatoires.

Réponse :

Un nombre aléatoire cryptographiquement sûr est :
- un nombre généré aléatoirement qui est difficile à prédire ou à reproduire, même en utilisant des informations sur la façon dont il a été généré
- un nombre aléatoire indépendant des autres nombres aléatoires générés précédemment.
- Il doit également être très difficile, étant donné les k premiers bits d'une séquence, de trouver le $(k+1)$-ème bit à l'aide d'un algorithme polynomial avec un taux de succès de plus de 50%.
Ces nombres sont souvent utilisés pour sécuriser les communications cryptographiques en générant des clés de chiffrement ou en ajoutant du bruit à des communications pour empêcher qu'elles ne soient déchiffrées.
En Java, nous pouvons utiliser la bibliothèque "SecureRandom" pour générer des nombres aléatoires cryptographiquement sûrs. Voici comment on peut utiliser cette bibliothèque pour générer un nombre aléatoire cryptographiquement sûr :

```java
import java.security.SecureRandom;

SecureRandom random = new SecureRandom();
int randomInt = random.nextInt();
```

- https://miashs-www.u-ga.fr/prevert/Prog/Java/CoursJava/lesBig.html
- https://stackoverflow.com/questions/11051205/difference-between-java-util-rando