Rapport du travail de session

**Dans le cadre du cours INF1433 : Initiation à la sécurité informatique**

[Company name] | [Company address]

Le chiffrement ELGAMAL

Faivre, Maxime(FAIM06039702) et CARLU, Ludovic (CARL18119709)

2016

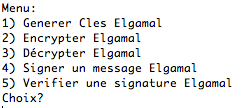
Initiation à la sécurité : rapport du travail de session

Le chiffrement Elgamal (sujet I)

# Exécution du programme :

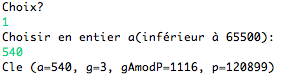
1. Ouvrir le terminal
2. Utiliser la commande « cd » pour se mettre dans le répertoire du projet
3. Lancer la commande : java -jar exec.jar

## Exemples d’exécution :

**Menu principal du programme :**

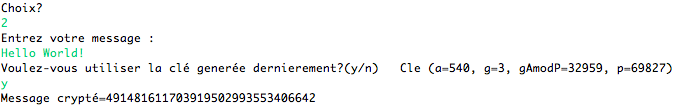
**On choisit maintenant la fonction qu’on veut exécuter en entrant le chiffre correspondant.**

1. **Pour générer une clé Elgamal :**

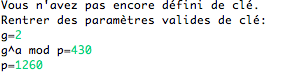


1. **La clé est ensuite stockée dans le programme, on peut donc l’utiliser pour coder un message, ou bien utiliser une autre clé en rentrant les paramètres à la main.**

**Utilisation de la clé stockée (en choisissant « y ») :**

****

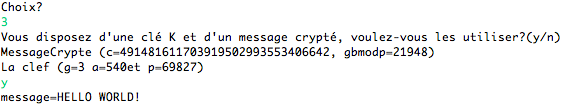
**Utilisation d’une clé extérieure (en choisissant « n ») :**



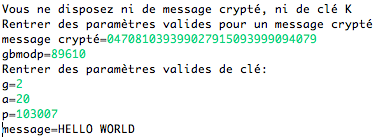
**Vous disposez maintenant d’un message crypté et stocké dans le programme.**

1. **Vous pouvez le décrypter en choisissant le choix 3. Soit en utilisant le message et la clé stockée ou soit en entrant vos propres paramètres.**

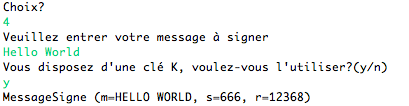
**Utilisation des paramètres stockés :**



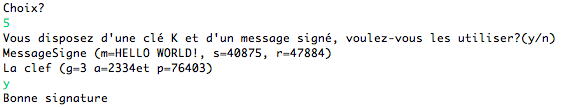
**Utilisation de paramètres extérieurs :**



1. **Vous pouvez aussi signer des messages en utilisant la fonction 4**



1. **Et une fonction pour vérifier la signature.**



## Discussion sur le projet

***Choix du langage de programmation :***

Nous avons choisi le langage Java pour une raison de portabilité, mais aussi de simplicité. En effet, nous savions tous les deux coder et étions à l’aise avec ce langage de programmation. Le langage Java nous offrait aussi la possibilité d’utiliser des classes d’objet déjà faite nous permettant de manipuler de grands nombres entier grâce à sa classe BigInteger et ses nombreuses fonctions.

***Choix de l’interface utilisateur :***

Pour ce qui est de l’interface utilisateur, nous avons opter pour une interface console. Cela nous semblait simple à implémenter et nous n’étions pas à l’aise avec les interfaces graphiques. De plus, notre interface permet à l’utilisateur d’utiliser des données déjà stocké pour qu’il puisse les réutiliser dans les autres fonctions. Par exemple, lorsqu’il génère une clé Elgamal avec la fonction 1, l’utilisateur peut réutiliser cette clé dans la fonction 2 pour crypter un message, cela permet d’éviter les copier-collers. De plus si l’utilisateur ne veut pas utiliser ses données, il est tout à fait en mesure d’utiliser ses propres arguments, il lui suffit juste d’indiquer au programme qu’il ne souhaite pas utiliser la clé qu’il a générer dernièrement et de rentrer ses propres paramètres dans la console.

***Difficultés rencontrées :***

Nous avons dû faire face à 2 principales difficultés.

La première était de comprendre les concepts de mathématique de générateur. En effet, nous ne savions pas ce qu’était un générateur, ni même sa notation. Nous avons travaillé sur ce concept pour acquérir cette notion avant même de commencer à coder.

Par la suite, notre 2ème difficulté a été de savoir comment manipuler de grands nombres entiers. Nous avons fait des recherches et avons trouvé qu’il existait une classe Java. La classe BigInteger nous a permis d’utiliser de grand entiers et de pouvoir faire de lourds calculs sans avoir de problème, par exemple leur appliquer des exposants ou des modulos.

Réalisé par Maxime FAIVRE(FAIM06039702) et Ludovic CARLU(CARL18118709).