Auteur : Imbert Victor
Date : 19/01/2020

Compte-rendu du TP A : Authentification de courriels

Auto-évaluation

Je pense avoir réussi les exercices avec un degré de 9/10 car les fonctions sont moyennement volumineuse. Une classe « Utils » propose des utilitaires permettant la conversion String ↔ byte[], l'opération XOR, et générateur de ipad et opad.
Les algorithmes utilisés pour parcourir le mail ne sont pas gourmands.

Pédagogie

L'intérêt de cet exercice repose dans la lecture correcte d'un fichier et de sa réécriture avec tous les éléments (y compris caractères invisibles tel le retour chariot) et l'utilisation de buffer pour utiliser une fonction de hachage au lieu d'une longue chaîne de byte. Il y a également la possibilité d'utiliser MD5 d'une façon à rendre une attaque plus difficile (respect d'une RFC).

Test

La valeur du HMAC pour « email1.txt » : f59a29d4e91b06ca297c93f4784c7313

Codage (modularité)

Tout est détaillé dans les commentaires « JavaDoc » des fonctions. Nous avons la fonction generateRFC2104Email qui s'occuppe d'appeler les fonctions « insertAppendice » et « calculateAppendice » qui respectivement insère un appendice dans un mail et le calcule. La fonction « checkEmailAuthencity » vérifie l'authenticité du mail en se servant du secret et de l'appendice qu'il récupère dans le champs X-AUTH du mail qu'il vérifie, il affiche dans la console en fin d'éxécution si le mail est authentique ou non.

La classe « Utils » possède une fonction « XORTwoByteArrays » qui fait l'opération XOR de 2 tableaux de byte.

Compilation et exécution

Il suffit de faire « make » pour compiler tous les programmes (Resume.java et Check.java)
On peut ensuite taper la commande pour générer un mail avec son appendice :

```
java Resume <nom Fichier mail> <nom Fichier mail produit>
```

Ainsi que la commande pour vérifier l'authenticité d'un mail :

```
java Check <nom Fichier mail à vérifier>
```

Commande pour tester avec l'exemple :

```
java Resume email1.txt emailOutput.txt
java Check emailOutput.txt
```