
TP (n° 1) de sécurité informatique

Matthias GRADAIVE - Antoine FOUCAULT

6 novembre 2014

©2014 ISTIC

Sommaire

1	Functionalités de nos programmes	1
1.1	Vigenère	1
1.2	Casseur	1
2	Tests de la qualité de nos programmes	1

Introduction

L'objectif de ce TP est d'étudier le code de Vigenère, afin de créer deux programmes :

- Un chiffreur/déchiffreur de message (nommé vigenere)
- Un casseur de message chiffré, chargé de retrouver la/les clé(s) probable(s) d'un message chiffré (nommé casseur)

1 Fonctionnalités de nos programmes

1.1 Vigenère

Le programme vigenère répond aux demandes du TP, il est capable de chiffrer/déchiffrer des fichiers sur les 256 valeurs possibles d'un char à partir d'une clé fournie par l'utilisateur.

De plus, il est possible de spécifier le chemin vers un fichier contenant la clé de chiffrement/déchiffrement à utiliser afin de ne pas limiter la valeur des caractères utilisables pour la clé à celles du shell.

1.2 Casseur

Le programme casseur répond aux demandes du TP, améliorations comprises.

Il est possible de simplement entrer le fichier chiffré et, dans le cas d'une cryptanalyse réussie, le programme retourne les 256 clés possibles, sous forme de chaîne de caractère ou sous forme hexadécimale (pour les caractères non inscriptibles).

Comme demandé dans la partie amélioration du sujet, il est possible de renseigner un fichier "modèle", afin de retourner seulement la clé la plus probable.

Enfin, dans le cas où on spécifie un fichier "modèle", il est possible d'exporter la clé trouvée ainsi que le déchiffré du fichier d'entrée avec cette clé dans des fichiers. Cela facilite l'utilisation de clés non-inscriptibles.

2 Tests de la qualité de nos programmes

2.1 Vigenère

Afin d'assurer l'intégrité des fichiers lors des opérations de chiffrement/déchiffrement avec le programme vigenere, nous avons créé un fuzzer.

Pour renforcer le caractère aléatoire de notre fuzzer, nous avons utilisé le générateur de nombres pseudo-aléatoires MRG32K3A, considéré comme un générateur fiable et non-redondant (pour l'étendue de nos tests).

Voici l'algorithme utilisé (N et M choisis par l'utilisateur) :

- Pour chaque longueur de clé i de 1 à N
- Pour chaque itération de 1 à M

- Générer une clé aléatoire de longueur i
- Chiffrer le fichier spécifié à l'aide de la clé et du programme vigenère
- Déchiffrer le fichier spécifié à l'aide de la clé et du programme vigenère
- Comparer le fichier original et le fichier déchiffré

Ainsi, à l'aide du pourcentage de réussite totale et des pourcentages de réussite par longueur de clé, nous avons pu corriger les erreurs présentes dans notre code et nous assurer du fonctionnement de vigenere.

2.2 Casseur

Conclusion

Ce TP nous a permis de mieux comprendre le code de Vigenère ainsi que sa cryptanalyse.

Le passage de la partie théorique au développement des programmes vigenère et casseur a permis de comprendre comment mettre en œuvre ces méthodes au sein d'un programme.