TP (n° 1) de sécurité informatique

Matthias Gradaive - Antoine Foucault

11 novembre 2014

Sommaire

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 Fonctionalités 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 de 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 caractère 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 contenant des caractères non-inscriptibles.

2 Limites

Notre programme supporte seulement les caractères ASCII et ne peut donc prendre en entrée des caractères accentués, obligeant l'utilisateur à nettoyer le fichier qu'il veut chiffrer.

3 Tests de la qualité de nos programmes

3.1 Vigenère

Afin d'assurer l'intégrité des fichiers lors des opérations de chiffrement/déchiffrement avec le programme vigenere, nous avons crée un fuzzer.
Pour renforcer le caractère aléatoire de notre fuzzer, nous avons utilisé le générateur

de nombres pseudo-aléatoires MRG32K3A, considéré comme une 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 précédemment crée à l'aide de la clé et du programme vigenère
    - Comparer le fichier original et le fichier déchiffré

Ainsi, à l'aide du taux de réussite totale et des taux de réussite par longueur de clé, nous avons pu corriger les erreurs présentes dans notre code et nous assurer du fonctionnement de vigenere afin d'atteindre 100% de réussite.

4 Difficultés rencontrées

La principale difficulté rencontrée a été le passage d'un alphabet de 26 valeurs à un alphabet de 256 valeurs. Nous sommes d'abord resté fixé sur l'utilisation de l'opération modulo puis avons ensuite compris que cette opération n'était pas nécessaire du fait de la rotation naturelle des nombres dans le type char (127+1 = -128).

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 permit de comprendre comment mettre en œuvre ces méthodes au sein d'un programme.