Securité Informatique: Projet 1

2-DES

Groupe 15

Pham Quang Vinh

Balon-Perin Alexandre

Rukundo Patrick

Année Académique 2010-2011

Table of Contents

Introduction

Dans le cadre du cours « Computer Security », Il nous a été demandé de réaliser un programme permettant de crypter et décrypter des messages grâce à l'algorithme de chiffrement DES. Cet algorithme devra être utilisé deux fois de suite, cette construction est appelée 2-DES. Nous avons décidé d'implémenter le code en java en utilisant l'API standard « crypto ».

DES

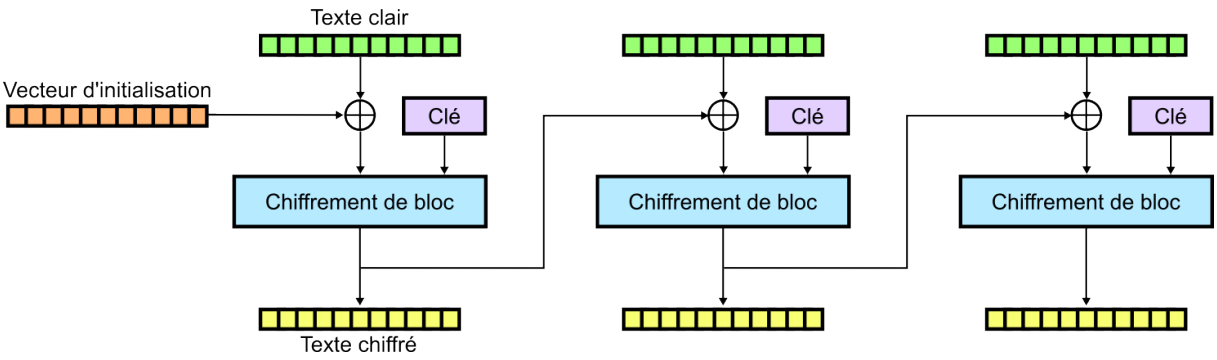
Le Data Encryption Standard est l'algorithme de chiffrement que nous allons utiliser dans ce projet. Plus précisément, nous allons implémenter 2-DES comme un outil d'une classe cryptographique. Il disposera des méthodes init(k1,k2,IV) où k1, k2 sont les clés de chiffrement, encrypt(fileName) et decrypt(fileName). Les clés utilisées dans cet algorithme sont constituées de 64 bits, 56 bits de codage et 8 bits de parité. Il faut remarquer que ces clés sont générées de façon pseudo aléatoire. En effet, les ordinateurs ne sont pas capables de générer des nombres totalement aléatoires. Cependant, cela sera suffisant dans le cadre de ce projet.

L'algorithme 2-DES est simplement la succession de deux algorithmes DES. On passe un message clair par un premier algorithme DES avec la première clé k1, et le résultat sera une nouvelle fois traité par un algorithme DES avec cette fois-ci la clé k2, afin d'obtenir le résultat d'un chiffrement par 2-DES.

DES-CBC

Le mode de chiffrement choisi est le CBC (Cipher Block Chaining).

Il consiste à traiter le fichier à crypter par bloc de 64 bits, chaque bloc traité étant réutilisé pour traiter le suivant. Cela évite d'avoir deux fois le même chiffrement dans le cas ou deux blocs seraient constitués des même caractères et donc une attaque par comparaison statistique.



Listing complet des implantations

Classe DES2

Méthodes :

Public void init (SecretKey k1, SecretKey k2, String IVstr)

Cette méthode instancie des objets de la classe Cipher, des outils de cryptographie. Lors de l'instanciation on passe en paramètre le String "algorithm/mode/padding". Comme demandé dans l'énoncé l'algorithme DES et le mode CBC (Cipher Block Chaining) sont utilisés.

Une fois instanciés, on va les initialiser pour qu'ils deviennent des outils d'encryptage ou de décryptage auxquels sont assignées la première ou la seconde clé

Public void streams (String addressI, String addressO, Cipher cipher)

Cette méthode sert à gérer les flux d'entrée et de sortie. C'est aussi dans cette méthode que se déroule le processus d'encryptage et de décryptage. En effet grâce à une instance de la classe CipherInputStream, on peut lire un fichier en le passant directement par l'algorithme d'encryptage/décryptage désigné par le Cipher qu'on passe en paramètre du constructeur.[…]

Le résultat du décryptage ou de l'encryptage se retrouve écrit dans un autre fichier.

En appliquant successivement cette méthode, on obtient l'algorithme 2DES demandé.

Public void encrypt (String addressI, String addressO)

Cette méthode permet l'encryptage selon l'algorithme 2DES. Elle fonctionne en appelant deux fois la méthode *streams* avec en lui passant cipher(k1) la première fois et le cipher(k2) la seconde.

On passe le fichier contenant le message clair avec le cipher travaillant avec la première clé à la méthode *streams* décrite ci-dessus. On en récupère un fichier intermédiaire, contenant un message codé. On passe ce message codé avec le cipher travaillant avec la deuxième et on récupère le résultat final.

Public void decrypt (String addressI, String addressO)

Cette méthode fonctionne sur le même principe que la méthode encrypt, elle fonctionne seulement dans l'autre sens. On décrypte avec la clé 2 ensuite avec la clé 1.

Méthode d'attaque contre 2-DES

Connaissant l'algorithme 2-DES, nous allons procéder à une attaque à texte connu « meet-in-the-middle ». Cette attaque est une attaque exhaustive, c 'est-à-dire qu'on va générer toutes les paires de clés possibles et les tester. Connaissant le message clair et message codé, cette attaque consiste à crypter le message clair par l'algorithme DES avec la première clé d'une paire et décrypter le message codé avec la deuxième clé de la paire. Si les résultats correspondent, il y a de fortes chances que l'on ait trouvé la bonne paire de clés. On confirme le résultat en appliquant l'algorithme 2-DES avec les deux clés trouvées sur un autre message clair dont le message codé est connu et on vérifie.

Dans le cadre de cet exercice, seuls les 16 bits de poids les plus faibles varient, les 40 bits restants sont tous à zéro. Ceci évite d'avoir un nombre de calcul difficilement réalisable avec un seul ordinateur moderne en un temps raisonnable. Cependant, il y a donc tout de même 2^16 * 2^16, autrement dit plus de 4 milliards de calculs à effectuer dans le pire des cas.

Pour réaliser cette attaque, nous avons procédé de la manière suivante : on lit les fichiers X1 et Y1, on génère 2^16 clés différentes, on code le fichier X1 et décode le fichier Y1 avec chacune de ces clés. Les résultats sont gardés en mémoire dans un tableau recueillant tous les messages codés et un tableau recueillant tous les messages décodés. On enregistre également les clés dans une table de clés.

Ensuite, on va parcourir les deux tableaux afin de trouver les correspondances entre les résultats. Pour chaque entrée d'un tableau, il faudra vérifier avec chacune des entrées de l'autre table. On se retrouve donc avec un calcul de 2^16 * 2^16, soit plus de 4 milliards d'opérations.

Une fois qu'on trouve une équivalence dans les résultats, on va vérifier que la paire de clés qu'on trouve est la bonne, en cryptant avec 2DES un autre message clair dont le message codé est connu, et comparer le résultat avec le message codé connu.

Si la vérification est satisfaisante, on a trouvé la paire de clés.

Remarques

Nous avons constaté un phénomène étrange. En effet, lors de l'attaque exhaustive, il nous arrive que l'algorithme de cryptanalyse nous donne un total de 16 clés. Nous avons remarqué que l'algorithme acceptait des clés qui différaient à 1 bit près et ce, pour chaque bit variable. On se retrouve donc avec 2^4 clés différentes qui satisfont l'algorithme d'attaque.

Conclusion

Ce premier projet en sécurité informatique nous a permis de faire une première approche dans ce monde. Nous avons ainsi pu découvrir comment utiliser les libraires cryptographiques mises à notre disposition. Nous avons également pu nous essayer à la cryptanalyse avec une attaque exhaustive de l'algorithme 2DES.