SCI 460 – Cryptographie
TP1 : algorithmes de chiffrement

IUT d'Arles – DUT SRC – 2010-2011

Objectifs : Étude des chiffrements par substitution (ou simple), symétriques (ou à clef privée) ou asymétriques (ou à clef publique).

Pour l'ensemble du TP, vous trouverez le sujet ainsi que les fichiers nécessaires pour répondre aux questions sur le site : http://x.heurtebise.free.fr.

1 Cryptographie par substitution ou simple

Nous allons étudier un algorithme de cryptographie par substitution relativement simple : il s'agit du chiffre de César. Le chiffre de César consiste simplement à **décaler les lettres** de l'alphabet de quelques crans vers la droite ou la gauche. Pour cet exercice, nous allons écrire un programme en javascript en répondant à chacune des questions suivantes.

Questions :

Q1. Écrire une fonction **Cesar_crypt(clair, decalage, chiffre)** qui permet de crypter un message 'clair' grâce à l'algorithme de César selon un 'decalage' donné afin de donner le message crypté 'chiffre'.

Q2. Écrire la fonction **Cesar_decrypt(chiffre, decalage, clair)** qui réalise l'opération inverse.

Q3. Écrire un programme en javascript ou un fichier HTML permettant de :
- demander à l'utilisateur le message en clair ainsi que la valeur de décalage ;
- d'afficher le message crypté ;
- d'afficher le message décrypté final.

Q4. Écrire une fonction **symbol_freq(chiffre, freq_table)** permettant de calculer la fréquence (stockée dans **'freq_table'**) d'apparition des lettres d'un message chiffre.

Q5. Écrire une fonction **attaque_cesar(chiffre, clair)** permettant de retrouver le texte en clair à partir du texte chiffre, grâce à une analyse des fréquence (question Q4).

Q6. Écrire un programme en javascript ou un fichier HTML permettant de :
- demander à l'utilisateur le message chiffré ;
- d'afficher le message crypté ;
- d'afficher le message décrypté final en utilisant l'attaque par étude des fréquences d'apparition des lettres du message crypté.

2 Cryptographie symétrique ou à clé privée

Nous allons étudier un algorithme de cryptographie symétrique (ou à clé privée) : il s'agit d'un chiffrement élémentaire par bloc, en 2 étapes :

1. Réalisation d'un ou-exclusif (XOR) bit à bit entre le bloc du message et la clé.
2. Mélange des bits du bloc obtenu à l'aide d'une permutation secrète des bits.

Pour cet exercice, nous allons écrire un programme en javascript en répondant à chacune des questions suivantes.

Questions :

Q7. Écrire une fonction **XOR_crypt(clair, clef, chiffre_temp)** qui permet de crypter un message **'clair'** grâce à la fonction XOR entre chaque bloc (de même taille que la '**clef**') et la '**clef**' afin de donner le message crypté **'chiffre_temp'**.

Q8. Écrire la fonction **XOR_decrypt(chiffre_temp, clef, clair)** qui réalise l'opération inverse.

Q9. Écrire un programme en javascript ou un fichier HTML permettant de :

- demander à l'utilisateur le message en clair ainsi que la clef ;
- d'afficher le message crypté avec l'algorithme XOR seul ;
- d'afficher le message décrypté final avec l'algorithme XOR seul.

Q10. Écrire une fonction **inverse(chiffre_temp, chiffre)** permettant d'inverse l'ordre des bits de chaque bloc du message '**chiffre_temp**' afin de donner le message crypté **'chiffre'**.

Q11. Écrire un programme en javascript ou un fichier HTML permettant de :

- demander à l'utilisateur le message en clair ainsi que la clef ;
- d'afficher le message crypté avec l'algorithme XOR seul et la fonction '**inverse**' ;
- d'afficher le message décrypté final avec l'algorithme XOR seul et la fonction '**inverse**' .

3 Cryptographie asymétrique ou à clé publique

Nous allons étudier un algorithme de cryptographie asymétrique (ou à clé publique) : il s'agit de l'algorithme RSA. Sa sécurité repose sur la difficulté de factoriser un entier (et plus spécifiquement du problème de l'extraction de racines modulaires).

Pour cet exercice, nous allons étudier l'algorithme RSA à l'aide du site d'Herbert Hanewinkel http://www.hanewin.net/encrypt/rsa/rsa-test.htm, écrit en javascript. Si vous désirez voir le contenu de ce site, vous pouvez toujours analyser le code source.

Questions :

Q12. Analyser la durée de création des clefs publiques et privées en fonction de la taille de celles-ci en bits.

Q13. Analyser la durée de chiffrement et de déchiffrement, pour plusieurs tailles de texte et plusieurs tailles de clefs.

Q14. Vous pourrez faire votre propre site avec les fichiers javascript issus de ce site.