Travaux pratiques –TP 1
Services de sécurité, mécanismes, algorithmes avec *API Openssl*

Partie 1 : OpenSSL

OpenSSL est une boîte à outils cryptographiques implémentant les protocoles SSL et TLS. Il offre :

- Une bibliothèque de programmation en C permettant de réaliser des applications client/serveur sécurisées s'appuyant sur SSL/TLS
- Une commande en ligne (OpenSSL) permettant
  - la création de clés RSA, DSA (signature)
  - la création de certificats X509
  - le calcul d'empreintes (MD5, SHA, RIPEMD160, ...)
  - le chiffrement et déchiffrement (RSA, DES, IDEA, RC2, RC4, Blowfish, ...)
  - la réalisation de tests de clients et serveurs SSL/TLS
  - la signature et le chiffrement de courriers (S/MIME)

La syntaxe générale de la commande openssl est

```
$ Openssl  <commande>  <option>
```

On trouvera toutes les informations la concernant à l'adresse : http://www.openssl.org
L'objectif de ce TP est de vous familiariser avec les services de base de la sécurité, chiffrement, hachage et signature.

a) Télécharger l'outil openssl avec la commande *apt-get install openssl*
b) Vérifier la bonne installation de l'outil
c) Répondez aux questions suivantes :
   1. Quelle est la version d'openssl installée ?
   2. Lister tous les algorithmes de cryptographie présents dans l'outil ?
d) Tapez la commande suivante et expliquer le résultat : *openssl ciphers –v*
e) Pour connaître toutes les fonctionnalités de openSSL : *man openssl*
f) Pour connaître les paramètres d'une fonction, taper la commande openssl enc –help
   enc est la fonction de chiffrement et déchiffrement symétrique.

Partie 2 : Fonctions de hachage

a) Créer le fichier Plain.txt contenant la phrase « travaux pratiques inf944 »

Générer le digest (l'empreinte) en SHA1 et MD5 pour votre fichier. (openssl dgst --help)

b) Modifier le contenu du fichier en remplacer seulement le caractère t par T du mot travaux. Générer de nouveau le digest du fichier. Comparer le résultat avec la question b). Conclure.
c) Télécharger le fichier usa.rar dans la rubrique travaux pratiques de la page du cours INF944 - Master Réseaux (RES). Décompresser le fichier et générer le digest en MD5 pour chaque fichier dans usa.rar. Qu'en pensez-vous ?

Partie 3 : Chiffrement symétrique

La commande *openssl enc* permet de chiffrer et déchiffrer des messages. Plus d'informations sur cette commande peuvent être trouvées en tapant openssl enc -h

a) Créer un fichier texte Plain.txt et y écrire : « Travaux Pratiques SECRES ». Chiffrer ce fichier avec l'algorithme DES-CBC et sauvegarder le fichier sous le nom Cipher.txt. Utiliser l'option –k pour saisir le mot de passe symétrique.

   La clé de déchiffrement est dérivée à partir du mot de passe

-k password, the password to derive the key from. k small letter
-kfile filename, read the password to derive the key from the first line of filename
-K key, the actual key to use: this must be represented as a string comprised only of hex digits. In Capital Letter

b) Ouvrir le fichier chiffré. Qu'observez-vous au niveau des premiers caractères ? À quoi cela sert-il ?
c) Pour la lisibilité ajouter l'option -base64 et re-chiffrer le fichier plain.txt. Ouvrir le fichier cipher.txt. Information : il faut ajouter –base64 pour déchiffrer le fichier plain.txt, sinon openssl affichera une erreur.

d) Déchiffrer le fichier chiffré en lui donnant le nom NewPlain.txt et vérifier qu'on retombe bien sur le fichier de départ. Pour vérifier : diff Plain.txt NewPlain.txt –q

e) Reprendre la question a) et c) mais cette fois ci en indiquant l'option –p pour afficher certaines informations ? Expliquer le résultat.

f) Chiffrer le plain.txt encore une fois (NewCipher.txt). Comparer les fichiers Cipher.txt et NewCipher.txt. Justifiez.

g) Refaire ce test deux nouvelles fois en ajoutant l'option -nosalt. Comparer et expliquer les résultats obtenus.

h) Grâce à la commande rand, créer un fichier GrandFichier.txt avec des données aléatoires, d'une taille d'environ 1000 Mo (1 Go). Utiliser la commande time pour calculer le temps de chiffrement de votre fichier en utilisant RC2, DES, 3DES, AES en mode CBC (Prendre le temps user). Comparer les résultats ?

i) Re-chiffrer le GrandFichier.txt avec RC2, DES, 3DES et AES en mode ECB. Comparer les temps déchiffrement avec les résultats obtenus dans la question h. Justifier.

j) Chiffrer le fichier GrandFichier.txt en -des-cbc en donnant la clé 36D1456C26A3670D et le vecteur d'initialisation FB22881684E1864D. Utiliser l'option –p. Justifier la taille de la clé et le vecteur d'initialisation. Déchiffrer le *cipher* obtenu.

k) Chiffrer le fichier GranFichier.txt en –aes-128-cbc en donnant les mêmes paramètres de la question précédente (clé et IV). Utiliser l'option –p. Justifier le résultat. Déchiffrer le *cipher* obtenu.

Pour connaitre les paramètres des commandes :
openssl enc --help
openssl rand --help

Partie 4 : Chiffrement asymétrique

Dans cet exercice, l'intérêt est de chiffrer un document avec une clé publique que seul son créateur pourra déchiffrer.

a) Générer une paire de clés RSA 2048 bits que vous nommerez PrivateKeyMyName.priv. Ex : PrivateKeyDupont.priv. Utilisez l'option –p.
b) Extraire ensuite la clé publique que vous nommerez PublicKeyDupont.pub
c) Envoyer à votre voisin votre clé publique (pas ssh ou clé usb)
d) Chiffrer le fichier Plain.txt avec la clé publique de votre collègue
e) Déchiffrer le fichier que vous avez reçu avec votre clé privée
f) Chiffrer le fichier GrandFichier.txt avec la clé publique de votre collègue. Vous devez remarquer un problème. Comment l'expliquez-vous ?