TP1 : Chiffrer et déchiffrer des données

Objectifs :

- Chiffrement et déchiffrement des données avec OpenSSL

Ressources requises :

- Poste de travail Kali linux

Contexte :

- ### Protocole SSL ?

  Le terme SSL est un acronyme pour Secure Socket Layer qui est un protocole (en fait un ensemble de protocoles) qui a été développé par la société Netscape Communication Corporation pour permettre de la communication sécurisée en mode client/serveur pour des application réseaux utilisant TCP/IP.

  Le principe général d'un protocole de type SSL est qu'il se passe en deux temps :

  Une poignée de mains : c'est une étape durant laquelle le client et le serveur s'identifient, se mettent d'accord sur le type du système de chiffrement et les clefs qui seront utilisés lors du reste de la communication.
  La phase de communication : les données sont alors échangées en format compressées et chiffrées et signées.

- ### OpenSSL ?

  OpenSSL est un projet open source qui fournit une boîte à outils robuste, de qualité commerciale et complète pour les protocoles TLS (Transport Layer Security) et SSL (Secure Sockets Layer). C'est aussi une bibliothèque de cryptographie à usage général. Au cours de ces travaux pratiques, vous allez utiliser OpenSSL pour chiffrer et déchiffrer des messages texte.

  La bibliothèque OpenSSL est une implantation libre des protocoles SSL et TLS qui donne accès à :

  - une bibliothèque de fonctionnalité écrite en C permettant de réaliser des applications client/serveur sécurisées s'appuyant sur SSL/TLS,
  - un ensemble d'exécutables en commande en ligne permettant :
    - la création de clef RSA, DSA (pour les signature)
    - la création de certificat X509 (identification)
    - le calcul d'empreinte (MD5, SHA, RIPEMD160, ...)
    - le chiffrement et le déchiffrement (RSA, DES, AES,IDEA, RC4, Blowfish ....)
    - la signature et le chiffrement de courriers (S/MIME).

A tout instant vous pouvez avoir une vue sur l'ensemble des fonctionnalités de OpenSSL à l'aide des pages de manuel (man openssl).
La syntaxe générale pour l'utilisation en mode shell des fonctionnalités OpenSSL est la suivante :

```
$ openssl <commande> <options>
```

Vous pouvez utiliser les fonctionnalités suivantes :

- openssl genrsa -out <fichier_rsa.priv> <size> : génére la clé privé RSA de taille size. les valeurs possible pour size sont : 512, 1024, etc.
- $ openssl rsa -in <fichier_rsa.priv> -des3 -out <fichier.pem> : chiffre la clef privé RSA avec l'algorithme DES3. Vous pouvez utiliser DES, 3DES, IDEA, etc.
- $ openssl rsa -in <fichier_rsa.priv> -pubout -out <fichier_rsa.pub> : stocke la partie publique dans un fichier à part (création de de la clé publique associée à la clef privée dans le fichier fichier.pem).
- $ openssl enc <-algo> -in <claire.txt> -out <chiffre.enc> : pour le chiffrement de claire.txt avec l'algorithme spécifié (openssl enc --help pour avoir la liste des possibilités ou bien openssl list-cipher-commands) dans un fichier chiffre.enc.
- $ openssl enc <-algo> -in <chiffre> -d -out <claire> : pour le déchiffrement.
- $ openssl rand -out <clé.key> <nombre_bits> : pour générer un nombre aléatoire de taille nombre_bits (utiliser l'option -base 64 pour la lisibilité).
- openssl aes-256-cbc -in <claire.txt> -out <chiffre.enc> -e -k <clé.key>: pour chiffrer un fichier avec l'AES.
- $ openssl pkeyutl -encrypt -pubin -inkey <rsa.pub> -in <clair.txt> -out <chiffre.enc> : chiffrer fichier.txt avec la RSA en utilisant la clef publique rsa.pub.
- $ openssl pkeyutl -decrypt -inkey <rsa.priv> -in <chiffre.enc> -out <fihcier.txt> : pour déchiffrer le fichier fic.dec.

Instructions :

A. Chiffrement symétrique :

1. Générer une clé symétrique AES 192 bits de manière aléatoire et enregistrez-la dans un fichier nommé "secret.key".
2. Chiffrer un fichier texte (message.txt) avec l'algorithme AES-128 puis enregistrer la version cryptée sous le nom "message.enc". Le contenu du fichier message.enc s'est-il affiché correctement ? À quoi ressemble-t-il ? Expliquez votre réponse. Pour rendre le fichier.
3. Créez un fichier texte contenant le message "Master Informatique FST" et nommez-le "fst.txt". Ensuite, utilisez la clé symétrique (générée aléatoirement (Question 1)) pour chiffrer le message, puis utilisez la même clé pour déchiffrer le fichier chiffré.
4. Au lieu d'un fichier, utilisez un mot de passe pour générer la clé et chiffrez le fichier "fst.txt" en utilisant l'algorithme AES-128.
5. Utilisez l'algorithme AES-256 pour chiffrer le fichier PDF du TP, puis déchiffrez-le.
6. Placez les fichiers "message.txt" et le fichier PDF du TP dans le même répertoire. Chiffrez ce répertoire en utilisant DES3

B. Chiffrement asymétrique :

1. Créez une paire de clés RSA 4096 ; vous devez chiffrer la clé privée avec l'algorithme DES3
2. Chiffrer un petit fichier avec votre clé RSA.
3. Envoyez votre clé publique à un voisin. Celui-ci vous enverra la sienne. Générer un petit fichier texte et envoyez-le à votre voisin chiffré avec sa clef publique. Lui vous enverra un fichier chiffré avec sa clef. Renvoyez-lui le message qu'il vous a envoyé mais en clair.
4. Essayer de chiffrer un gros fichier avec votre clef RSA.
5. Toujours en binôme : ***A*** génère une clef AES 256 qu'il chiffre avec la clef publique RSA de ***B*** et il lui envoi le chiffré. A partir de là, ***B*** récupère la clef (en clair), et il chiffre un gros fichier avec la clef AES puis Il envoie le gros fichier chiffré à ***A*** qui doit le déchiffré.