2017/2018

TP de sécurité

TP n1 : (compte rendu)

Réalisation d'une boite d'outils de cryptographie classique

1. Cesar à implémenter chiffrement / déchiffrement de n rang ;
2. Mot de passe (0..9)

TP n2 : (Crypter/Decrypter avec les bibliotheques de SSL)

Qu'est-ce-qu'OpenSSL ?

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 applications réseaux utilisant TCP/IP.

Le protocole TLS (Transport Layer Security) est une évolution de SSL réalisé par l'IETF et qui sert de base à HTTPS par exemple. Le protocole SSL est entre la couche TCP/IP et une application utilisant TCP. Le principe général d'un protocole de type SSL est qu'il se passe en deux temps :

1. **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.

2. **La phase de communication** : les données sont alors échangées en format compressées et chiffrées et signées.

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

Dr H. Nacer

- Une bibliothèque de fonctionnalité écrite en C permettant de réaliser des applications client/serveur sécurisées s'appuyant sur SSL/TSL,
- Un ensemble d'exécutables en commande en ligne permettant :

− La forge de clef RSA, DSA (pour les signatures)

− La création de certificat X509 (identification)

− Le calcul d'empreinte (MD5, SHA, RIPEMD160, ...)

− Le chiffrement et le déchiffrement symétrique et asymétrique (exemple : RSA, DES, IDEA, RC2, RC4, Blowfish, ...)

− La réalisation de tests de clients et serveurs SSL/TSL

− La signature et le chiffrement de courriers (S/MIME).

Lorsque vous vous connectez à un site en HTTPS, c'est OpenSSL qui s'occupe du chiffrement de la connexion, si le serveur utilise OpenSSL, comme la quasi-totalité des OS Unix-like. OpenBSD par exemple utilise un fork d'OpenSSL qui se nomme LibreSSL.

OpenSSL peut donc chiffrer des flux mais également des fichiers.

Vous pouvez avoir une vue sur l'ensemble des fonctionnalités de OpenSSL à l'aide des pages de manuel (man openssl).

Openssl –list cipher - commands : affiche la liste des algorithmes de cryptage possible.

Exemple : **/* chiffrement symétrique */**

- Créer un message : nano testmes
- Lire le message : cat testmes
- Lire le repertoire : ls /* retrouver testmes*/
- Chiffrer : openssl enc-aes-256-cbc-base64 - in testmes – out testenc

Explication :

$ openssl <commande> <option >

$ représente le prompt du shell.

A. Cryptographie symétrique

- **$ Openssl enc <-algo> -in <input.txt> -out <output.txt> -kfile <password.txt>** : chiffre le fichier input.txt avec l'algorithme et le mot de passe spécifiés.
- **$ Openssl enc <-algo> -in <output.txt> -d -out <clair.txt> -kfile <password.key>** : chiffre le fichier input.txt avec l'algorithme et le mot de passe spécifiés.
- **$ Openssl rand -out <password.key> <size>** : générer un mot de passe aléatoire sur « *size »* bits
- **Enc –help**
- **list-cipher-commands** afficher la liste des algorithmes de cryptage possible.

B. Cryptographie anti-symétrique

- **$ openssl genrsa -out <clepv.prem > <size>** : génére la clé privé et publique RSA de taille size bits. les valeurs possibles pour size sont : 512, 1024, etc.
- **$ Openssl rsa –in <fichier_rsa.pem> -pubout –out <clepub.pem>** : stocker la clé publique dans un fichier à part.
- **$ Openssl rsa –in <clepv.pem> -text –noout** : afficher le contenu détaillé de la clé privée RSA générée.
- **$ Openssl rsa –pubin –in <clepub.pem> -text –noout:** afficher le contenu détaillé de la clé publique RSA générée.
- **$ Openssl rsa –in <clepv.prem> -des3 –out <clepvchiffre.pem>** : chiffre la clef privé RSA avec l'algorithme DES3. Vous pouvez utiliser DES, 3DES, IDEA, etc.
- **$ Openssl rsautl –encrypt –pubin –inkey clepub.pem –in input.txt –out crypt.txt : chiffrer le fichier input.txt avec la clé clepub.pem**
- **Openssl rsautl –decrypt –inkey clepv.pem –in crypt.txt –out clair.txt : déchiffrer le fichier crypt.txt avec la clé clepv.pem**