TP 1 : OpenSSL usages de base

1 Qu'est-ce-qu'OpenSSL

1.1 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 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 générale 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.

1.2 OpenSSL

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

- 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 signature)
  - la création de certificat X509 (identification)
  - le calcul d'empreinte (MD5, SHA, RIPEMD160, ...)
  - le chiffrement et le déchiffrement (RSA, DES, IDEA, RC2, RC4, Blowfish, ...)
  - la réalisation de de tests de clients et serveurs SSL/TSL
  - 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'utilsation en mode shell des fonctionnalités OpenSSL est la suivante :

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

le `$` représente le prompt du shell.

2 Opérations de base avec OpenSSL

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 dgst <-algo> -out <sortie> <entrée>` : pour hacher un fichier. L'option `<-algo>` est le choix de l'algorithme de hachage (sha, sha1, dss1, md2,md4, md5, ripemd160).
- `$ 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 rsautl -encrypt -inkey <rsa.pub> -in <clair.txt> -out <chiffre.enc>` : chiffrer `fichier.txt` avec la RSA en utilisant la clef publique `rsa.pub`.
- `$ openssl rsautl -decrypt -inkey <rsa.priv> -in <chiffre.enc> -out <fihcier.txt>` : pour déchiffrer le fichier `fic.dec`.
- `$ openssl rsautl -sign -inkey <ras.priv> -in <fichier.txt> -out <fic.sig>` : pour générer une signature.
- `$ openssl rsautl -verify -pubin -inkey <rsa.pub> -in fic fic.sig` : pour vérifier une signature.

3 Que faut-il faire ?

Se familiariser avec les fonctionalités et la bibliothèque openSSL : Dans ce premier TP, on va utiliser les fonctionalités `openSSL` comme vous l'avez peut-être déjà un peu fait dans le court d'usages sécurisés des TIC. Nous ne ferons pas la même chose. Le but est de se préparer à l'utilisation de la bibliothèque. Dans un premier temps, vous allez faire les choses suivantes en utilisant les fonctionalités OpenSSL et pas encore la bibliothèque :

1. Forgez vos clefs RSA 512.
2. Chiffrer le petit fichier en 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. Générez une clef AES 256. Chiffrez le fichier pdf du TP avec puis déchiffrez-le.
5. Toujours en binome : A génére une clef AES 256 qu'il chiffre avec la clef publique RSA de B et il lui envoye le chiffré. A partir de là, B récupère la clef (en clair), et il chiffre un gros fichier avec l'AES et la clef AES. Il envoie le gros fichier chiffré à A qui doit le déchiffré.
6. Mettre en place une poignet de mains avec les fonctionalités déjà utilisées.