Lycée Voltaire

T.P : Sécurité des communications

N.S.I

1 Présentation de openSSL

1.1 Protocole SSL

Le protocole SSL (Secure Socket Layer) a été développé par la société Netscape Communications Corporation pour permettre aux applications client/serveur de communiquer de façon sécurisée. SSL a été universellement adopté sur le World Wide Web pour authentifier et chiffrer les communications entre clients et serveurs.

Une session SSL se déroule en deux temps :

1. une phase de poignée de mains (handshake) durant laquelle le client et le serveur s'identifient, conviennent du syst ?me de chiffrement et d'une clé qu'ils utiliseront par la suite.
2. la phase de communication proprement dite durant laquelle les données échangées sont compressées, chiffrées et signées.

L'identification durant la poignée de mains est assurée à l'aide de certificats X509.

1.2 openSSL

openSSL (http://www. openssl .org) est une boîte à outils cryptographiques qui va nous permettre :
— la création de clés RSA, DSA (signature) ;
— la création de certificats X509 ;
— le calcul d'empreintes (MD5, SHA, RIPEMD160, ...) ;
— le chiffrement et déchiffrement (DES, IDEA, RC2, RC4, Blowfish, ...) ;
— la réalisation de tests de clients et serveurs SSL/TLS ;
— la signature et le chiffrement de courriers (S/MIME).
Pour connaître toutes les fonctionnalités de **openSSL** taper dans un terminal : `man openssl`

2 Chiffrement symétrique

openssl permet d'utiliser plusieurs algorithmes de chiffrement symétrique.
Pour obtenir une liste des syst ?mes disponibles taper dans un terminal : `openssl enc -ciphers`

Exercice nº 1

1. Créer un fichier `toto`. Contenant un texte de votre choix.
2. Ouvrir un terminal (lien) et importer (petite fl ?che en bas) le fichier toto. La commande `ls` vous permettra de tester que le fichier est bien importé.
3. Pour chiffrer un fichier en utilisant l'algorithme "blowfish" en mode CBC, on utilise :

   ```
   openssl enc -bf-cbc -in toto -out toto.chiffre
   ```

   Taper le code ci-dessus et observer le contenu du fichier avec la commande : `cat toto.chiffre`
4. Pour déchiffrer un message, on utilise :

   ```
   openssl enc -bf-cbc -d -in toto.chiffre -out toto.dechiffre
   ```

5. Y a t-il une différence de taille entre les fichiers `toto` et `toto.dechiffre` ?
6. Comparer le contenu des fichiers `toto` et `toto.dechiffre` (Le faire manuellement et en utilisant la commande : `diff toto toto.dechiffre`)

Exercice nº 2

On a chiffré le fichier top_secret.chiffre avec l'algorithme "blowfish" en mode CBC

1. Télécharger le fichier top_secret.chiffre
2. Dans le terminal, importer le fichier `top_secret.chiffre`
3. Déchiffrer le message à l'aide de la clef : `Password1234`
4. Exporter le fichier déchiffré avec la commande `export_file nomfichier`

3 Chiffrement asymétrique - RSA avec openSSL

Génération d'une paire de clés

Pour générer une paire de clés RSA de 2048 bits, stockée dans le fichier `maCle.pem`, on saisit la commande suivante :

```
openssl genrsa -out maCle.pem 2048
```

Exercice nº 3

Pourquoi parle t-on d'une **PAIRE** de clefs ?

Le fichier `maCle.pem` obtenu est un fichier au format PEM (Privacy Enhanced Mail, format en base 64).
Pour visualiser la clef privée saisir la commande : `cat maCle.pem`
Pour visualiser la clef publique saisir la commande : `openssl rsa -in maCle.pem -pubout`

Exercice nº 4

1. La commande : `openssl rsa -in maCle.pem -pubout -out maClePublique.pem` permet d'enregistrer uniquement la clef publique dans le fichier `maClePublique.pem`.

2. La commande : `cat maClePublique.pem` permet de visualiser la clef publique.

3. Télécharger votre clé publique avec : `export_file maClePublique.pem`.
4. Télécharger votre paire de clés avec : `export_file maCle.pem`.
5. Pourquoi le fichier **meCle.pem** ne doit pas être diffusé ?
6. Renommer votre clef publique : `nom_pub.pem`. Déposer votre clef **PUBLIQUE** via ce lien et **GARDER PRECIEUSEMENT** votre clef privée.

3.1 Chiffrement/Déchiffrement de données avec RSA

Voici la commande pour chiffrer des données avec une clé publique RSA.

```
openssl rsautl -encrypt -in <fichier_entree> -inkey <clePublique> -pubin -out <fichier_sort
```

où :
— **fichier_entree** est le fichier des données à chiffrer. Attention, ce fichier ne doit pas avoir une taille excessive (ne doit pas dépasser 116 octets pour une clé de 1024 bits) ;
— **clePublique** est le fichier contenant la clé RSA publique.
— **fichier_sortie** est le fichier chiffré.

Pour déchiffrer, on remplace l'option `-encrypt` par `-decrypt`. Le fichier contenant la clé doit évidemment contenir la partie privée. Ce qui donne :

```
openssl rsautl -decrypt -in <fichier_entree> -inkey <cle> -out <fichier_sortie>
```

Exercice nº 5

1. Ecrire à mon intention un message **gentil et respectueux**. Ce message que vous signerez devra ?tre au format txt et contenir au maximum 70 caractères.
2. Chiffrer ce fichier avec **MA** clef publique. (récupérable via ce lien).
3. Déposer le fichier chiffré via ce lien
4. Je devrais ?tre capable de le déchiffrer. Patientez un peu ...en passant à la question suivante.
5. Récupérer un message gentil différent du votre ce lien et essayer de le déchiffrer.
6. Comparer la taille du fichier déchiffré avec celui chiffré. (donner le facteur multiplicatif)

4 Utilisation des deux chiffrements

Exercice nº 6

Télécharger le fichier : operation_espadon.chiffre que j'ai chiffré avec une clé de chiffrement symétrique utilisant l'algorithme "blowfish" en mode CBC.
Cette clé de chiffrement symétrique a été chiffée avec votre clef publique puis déposée via ce lien.

Qu'est-ce qui se cache derri ?re **operation_espadon.chiffre** ? (J'esp ?re que vous avez conservé précieusement votre clef privée !)

Exercice nº 7

1. Choisir une clé pour un chiffrement symétrique.
2. Créer un fichier qui décrit et explique vos souhaits d'orientation post-bac.
3. Chiffrer ce fichier avec la clé symétrique en utilisant l'algorithme "blowfish" en mode CBC.
   (Nommé le fichier chiffré : `VotreNom_orientation.chiffre`)
4. Ecrire votre clé de chiffrement symétrique dans un fichier texte et chiffrer le fichier avec ma clef publique.
   Nommer ce fichier chiffré : (`VotreNom_cle.chiffre`)
5. Déposer les deux fichiers chiffrés via ce lien.