M1 2016 - 2017

SECU **TP**

Exercice 1 -

1. Calculez le haché d'un même message avec les fonctions de hachage SHA-1, RIPE-MD160 et MD5.
2. Calculez le haché du même message à l'aide de la commande *md5sum*.
3. Chiffrez le message de la question 1. avec les algorithmes Blowfish, 3DES et CAST5.
4. Déchiffrez les messages et comparez les à l'original en utilisant SHA-256.
5. Téléchargez une image en .jpg sur internet, chiffrez la avec AES.
6. Générez une paire de clés RSA.
7. Demandez à un autre binôme sa clé publique.
8. Envoyez à ce binôme la clé secrète utilisée à la question 5. chiffrée avec la clé publique et le chiffré de l'image.
9. Déchiffrez les fichiers reçu et affichez l'image.

Exercice 2 -

Connectez vous à google en https.

1. Combien de certificats sont présent dans la chaîne de certificats ?
2. Quelle est l'autorité de certification racine ?
3. Quelle suite cryptographique est utilisée pour cette session ?
4. Quel algorithme est utilisé pour garantir l'intégrité ?
5. Quel algorithme est utilisé pour garantir la confidentialité ?
6. Décrire les autres algorithmes de la suite cryptographique et leurs fonctions.
7. Mêmes questions pour le site : `https://wwwd.caf.fr`
8. Mêmes questions pour le site : `https://outlook.com`
9. Utiliser la commande *speed* de openssl pour comparer la vitesse de ces différents algorithmes.

Exercice 3 - La certification avec OpenSSL

1. Générez une paire de clés RSA de 2048 bits dans un fichier **Keys.pem**. Vérifiez la longueur de p, q, et du module $N = pq$ dans la clé publique. Chiffrez votre fichier **Keys.pem** avec 3Des.
2. Exportez la clé publique dans un fichier **PKey.pem**.
3. Vous allez jouer le rôle d'une autorité de certification (AC) et, au même temps, celui d'une personne qui veut certifier sa clé publique. Du côté AC : créez un répertoire qui s'appelle **exampleCA**. Dans ce répertoire-ci, créez deux répertoires : un qui s'appelle **certs**, l'autre qui s'appelle **private**. Pour ceci vous pouvez utiliser la commande :

   ```
   mkdir certs private
   ```

   Puis changez les permissions pour le répertoire private :

   ```
   chmod 700 private
   ```

   Vous devez également créer un compteur pour les numéros de certificats. On fait cela avec la commande :

   ```
   echo '01' > serial
   ```

   Vous pouvez visualiser l'effet en utilisant les commandes *ls* et *cat*.
4. Créez deux fichiers : **index.txt** et **openssl.cnf**.

   ```
   touch index.txt
   touch openssl.cnf
   ```

   Le dernier de ces deux fichiers est un fichier de configuration pour les certificats.
5. Exécutez les commandes suivantes :

   ```
   OPENSSL_CONF=./openssl.cnf
   export OPENSSL_CONF
   echo $OPENSSL_CONF
   ```

   Vous devez voir s'afficher le chemin du fichier **openssl.cnf**.
6. Télécharger le fichier **openssl.cnf** à l'adresse :

   `https://seafile.cifex-dedibox.ovh/d/5a5ca91281/`

   Utiliser le pour compléter votre fichier **openssl.cnf**. Remplacer l'utilisation de md5 par sha1.
7. Utilisez la commande :

   ```
   openssl req -x509 -newkey rsa:2048 -out cacert.pem -outform PEM
   ```

   Décrire l'effet de cette commande.
8. Vérifiez l'existence des fichiers **cacert.pem** et **privkey.pem**. Sinon corriger votre fichier de configuration.
9. Visualisez le certificat en utilisant la commande :

```
openssl x509 -in cacert.pem -text -noout
```

10. Maintenant on joue le rôle d'une personne qui veut certifier sa clé publique (la clé dans **PKey.pem**). On doit créer une requête de certification (que l'on appelle *Certreq*) en utilisant la commande :

    ```
    openssl req -new -key Keys.pem -out Certreq
    ```

    Après la création du fichier, veuillez visualiser le fichier que vous avez créé en utilisant la commande :

    ```
    openssl req -in Certreq -text -noout
    ```

11. Normalement, l'usager va envoyer le fichier Certreq à l'autorité de certification. Pour l'instant, comme vous jouez le rôle de l'AC, vous devez signer cette requête en utilisant la clé de l'AC. Quelle clé est-ce que vous devez utiliser – la clé publique où la clé privée de l'AC ? Où se trouve-t-elle ?

12. Signez la requête en utilisant la commande :

    ```
    openssl x509 -days <durée de validité en jours> -CAserial
    serial -CA cacert.pem -CAkey <chemin au fichier nécessaire pour
    la clé dans la question 11> -in ../req -req -out ../cert
    ```

13. Visualisez votre certificat en utilisant la commande :

    ```
    openssl req -in Certreq -text -noout
    ```

    Lisez maintenant le contenu du fichier **serial**. Pourquoi a-t-il changé ?

14. Maintenant vous avez une paire de clés privée/publique et un certificat pour elle. Utilisez les pour signer et chiffrer un email.