TP 3
Cryptographie

Exercice 1 : Empreintes numériques MD5, SHA-1 et SHA-2

a) Créez un fichier texte contenant un petit paragraphe, et calculez l'empreinte numérique de ce document avec les commandes `md5sum`, `sha1sum` et `sha256sum`. Donnez l'empreinte obtenue avec chaque commande.

b) Donnez la taille de chaque empreinte (en bits).

c) Modifiez un caractère dans le fichier texte, et calculez à nouveau les empreintes numériques MD5, SHA-1 et SHA-2. Que constatez-vous ?

d) Quelle est l'utilité de l'empreinte numérique ?

e) Laquelle des méthodes d'empreinte numérique précédentes (MD5, SHA-1, SHA-2) est recommandée au jour d'aujourd'hui, et pourquoi ?

Exercice 2 : Trousseau de clés et signature numérique GPG

a) Qu'est-ce que GPG ?

b) Générez une paire de clé avec la commande `gpg --gen-key` en utilisant les paramètres par défaut.

c) Listez votre trousseau de clés avec la commande `gpg --list-keys` et validez la présence de vos clés.

d) Exportez votre clé publique avec la commande `gpg --armor --output ma-clé.asc --export UserID` et donnez le résultat.

e) Créez un fichier texte contenant un petit paragraphe, et chiffrez ce document avec la commande `gpg -er UserID document.txt`. Validez le résultat en visualisant le fichier `document.txt.gpg`, et supprimez le document non chiffré.

f) Déchiffrez le document chiffré créé précédemment avec la commande `gpg document.txt.gpg`, et validez le résultat.

g) Signez le document texte initial avec la commande `gpg --clearsign document.txt`. Validez le résultat en visualisant le fichier `document.txt.asc`.

h) Vérifiez le document signé avec la commande `gpg --verify document.txt.asc`.

i) Quelle est l'utilité de la signature numérique ?

Exercice 3 : Chiffrement symétrique AES

a) Créez un fichier texte contenant un petit paragraphe, et chiffrez ce document en AES 256 bits avec la commande `openssl enc -aes-256-cbc -in inputfile -out outputfile`. Validez que le document est bien chiffré.

b) Déchiffrez le document chiffré avec la commande `openssl enc -d -aes-256-cbc -in inputfile -out outputfile`. Validez que le document est bien déchiffré.

Exercice 4 : Chiffrement asymétrique RSA

a) Générez une clé privée RSA de 1024 bits avec la commande `openssl genrsa -out server-private.key 1024`, et visualisez le contenu du fichier ainsi généré.

b) Affichez les paramètres de votre clé privée RSA avec la commande `openssl rsa -in server-private.key -text -noout`, et commentez le résultat.

c) Générez une clé publique RSA à partir de votre clé privée avec la commande `openssl rsa -in server-private.key -pubout -out server-public.key`, et visualisez le contenu du fichier ainsi généré.

d) Générez un fichier aléatoire de 32 octets avec la commande `openssl rand 32 -out rand32.txt`. Ensuite chiffrez le fichier avec la commande `openssl rsautl -encrypt -pubin -inkey server-public.key -in rand32.txt -out rand32enc.txt`. Validez que le document est bien chiffré.

e) Déchiffrez le fichier chiffré avec la commande `openssl rsautl -decrypt -inkey server-private.key -in rand32enc.txt -out rand32-dec.txt`. Validez que le document est bien déchiffré.

Exercice 5 : Connexion SSH automatique avec clés RSA

a) Initialisez la configuration SSH du compte `etudiant` de deux machines Linux avec la commande `rm -rf ~/.ssh`. Lancez le service SSH sur une des deux machine Linux avec la commande `service ssh start`, et utilisez l'autre machine en tant que client SSH pour validez le fonctionnement du service SSH avec la commande `ssh etudiant@adresseserver`. Puis déconnectez-vous du serveur.

b) Sur la machine client SSH avec le compte `etudiant`, générez les clés RSA avec la commande `ssh-keygen` en laissant les paramètres par défaut et sans mot passe. Visualisez le contenu des fichiers générés. Ensuite ajoutez votre identité RSA à l'agent d'authentification local SSH avec la commande `ssh-add`.

c) Transférez votre clé sur le serveur SSH avec ma commande `ssh-copy-id etudiant@adresseserver`. Validez la connexion automatique du client avec le serveur en utilisant la commande `ssh etudiant@adresseserver`.

Exercice 6 : Connexion SSL / TLS

Capturez le trafic réseau d'une connexion complète HTTPS, et expliquez le fonctionnement de la négociation SSL / TLS.

Exercice 7 : Autorité de certification

Créez une autorité de certification.