TP N°1 Réseaux 2A - Cryptographie
Module TR-C8

Un compte rendu est à rendre à la fin du TP. Le compte rendu (concernant uniquement le point 1.4) doit faire apparaître les commandes tapées, leurs résultats (lorsque c'est possible), des explications et des commentaires.

1. Cryptage

Openssl est un logiciel permettant la gestion des paramètres et des clés pour SSL.

1.1. Clé secrète

`openssl enc -h` : liste les options et les différents algorithmes de cryptage
`openssl enc -algo -P -nosalt` : génère une clé secrète à partir seulement d'un mot de passe.
`openssl enc -algo -kfile fichierclé -in texteclair -out textechiffré -e`: crypte, la clé secrète est dans un fichier.
`openssl enc -algo -kfile fichierclé -in textechiffré -out texteclair -d:` décrypte.

a) Générer une clé secrète. Stocker la clé dans un fichier.
b) Générer un texte (par exemple avec `emacs`) puis crypter (avec Triple DES) ce texte avec la clé.
c) Décrypter le texte, vérifier.

1.2. Clés asymétriques

Génération de clés (dans `openssl` la clé privée comporte la clé publique)
`openssl genrsa -h` : liste les options et les différents algorithmes de cryptage.
`openssl genrsa -out fichierclé` : génère la clé privé.
`openssl genrsa -out fichierclé -algo` : génère la clé privé et la crypte.

a) Générer une clé RSA de 2048 bits non cryptée. La visualiser.
b) Générer une clé RSA cryptée par une passphrase. La visualiser.

Administration des clés
`openssl rsa -h` : aide
`openssl rsa -in key.pem -pubout` : extrait la clé publique de la clé privée.
`openssl rsa -in key.pem -pubout -out pubkey.pem` : affiche la partie publique de la clé privée.
`openssl rsa -in key.pem -algo -out nkey.pem` : change la passphrase ou l'algorithme de cryptage.
`openssl rsa -in key.pem -text` : affiche les composants de la clé

c) A partir de la clé privée précédente extraire la clé publique. Vous devez maintenant posséder deux fichiers Apub, Apriv.

d) Afficher les composants de votre clé privée et expliquer le format d'archivage de la clé

Utilisation des clés : signature, vérification, encryption, et décryption
`openssl rsautl -h` : aide
`openssl rsautl -in entrée -out sortie -inkey pubkey.pem -pubin -encrypt`: crypte en utilisant la clé publique
`openssl rsautl -in entrée -out sortie -inkey key.pem -decrypt` : décrypte en utilisant la clé privée
`openssl rsautl -in entrée -out sortie -inkey key.pem -sign` : signe le fichier en utilisant la clé privée

`openssl rsautl -in entrée -out sortie -inkey pubkey.pem -verify` : contrôle la signature (décrypte avec la clé publique)

e) Utiliser la clé publique pour crypter un message. Vérifier que la clé privée le décrypte.

1.3. Message digest

`openssl dgst -h`
`openssl dgst fichier` : calcule l'empreinte du fichier
`openssl dgst -algo -sign fichierclépriv file` : calcule et signe l'empreinte du fichier
`openssl dgst -algo -verify fichierclépublic -signature empreintesignée file` : vérifie l'empreinte signé avec la clé public.

a) Calculer l'empreinte d'un fichier texte.
b) Modifier un seul bit dans ce fichier, par exemple un A (code 65) devient C (code 67) et recalculer l'empreinte.
c) Calculer la signature de l'empreinte d'un fichier.
d) Vérifier cette signature.

1.4. Scénario

On souhaite transmettre un message "**message confidentiel**" de l'émetteur A vers le destinataire B avec les services de sécurité suivants :

- Authentification des parties,
- Intégrité et confidentialité des données transmises.

Pour cela, on décide de réaliser dans l'ordre les opérations suivantes :

1. chiffrement du message en **DES** (**mode CBC**) avec la clé de session "**secret**",
2. calcul de l'empreinte du message chiffré avec l'algorithme **md5**, le tout chiffré avec la clé RSA privée de l'émetteur (c'est la signature du message).
3. transport de la clé de session par chiffrement du mot de passe "secret" avec la clé publique du destinataire.

Il est demandé de préciser dans vos réponses si les clés utilisées sont celles de A et/ou de B.

a) Calculer la signature du message à transmettre et le crypte de la clé de session.
b) Quel est le message final qui sera transmis au destinataire ?
c) Vérifier l'authenticité (Authentification de l'émetteur + Intégrité des données) du message reçu à la destination puis le déchiffrer.