

Travaux Pratiques – TP

Chiffrement Asymétrique

Partie 1 : OpenSSL

OpenSSL est une boîte à outils cryptographiques implémentant les protocoles SSL et TLS. Il offre :

- Une bibliothèque de programmation en C permettant de réaliser des applications client/serveur sécurisées s'appuyant sur SSL/TLS
- Une commande en ligne (OpenSSL) permettant
 - la création de clés RSA, DSA (signature)
 - la création de certificats X509
 - le calcul d'empreintes (MD5, SHA, RIPEMD160, ...)
 - le chiffrement et déchiffrement (RSA, DES, IDEA, RC2, RC4, Blowfish, ...)
 - la réalisation de tests de clients et serveurs SSL/TLS
 - la signature et le chiffrement de courriers (S/MIME)

La syntaxe générale de la commande `openssl` est

`$ openssl <commande> <option>`

On trouvera toutes les informations la concernant à l'adresse : <http://www.openssl.org>

- a) Lancer une machine virtuelle *Debian* et taper la commande `su -s` (mot de passe : *root*)
- b) Vérifier l'installation de l'outil en tapant la commande `openssl`
- c) Répondre aux questions suivantes :
 1. Quelle est la version d'openssl installée ? taper la commande `openssl`, ensuite la commande `version -a`
 2. Lister tous les algorithmes de cryptographie présents dans l'outil ? taper la commande `list-cipher-algorithms`. Analyser le résultat.
- d) Pour connaître toutes les fonctionnalités de openssl : `man openssl`
- e) Pour connaître les paramètres d'une fonction, taper la commande `openssl rsautl -h`
`rsautl` est la fonction de chiffrement et déchiffrement asymétrique.

Partie 2 : Chiffrement asymétrique

Dans cet exercice, l'intérêt est de chiffrer un document avec une clé publique que seul son créateur pourra déchiffrer.

La commande `openssl rsautl` permet de chiffrer et déchiffrer des messages. Cette commande permet aussi de signer un fichier et vérifier la signature. Plus d'informations sur cette commande peuvent être trouvées en tapant `openssl rsautl -h`.

La commande `openssl rsa` permet de générer une paire de clés. Plus d'informations sur cette commande peuvent être trouvées en tapant `openssl rsa -h`.

- a) Générer une paire de clés RSA 1024 bits que vous nommerez `PrivateKeyMyName.priv`.
Ex : `PrivateKeyDupont.priv`. Utilisez l'option `-p`.
- b) Extraire ensuite la clé publique que vous nommerez `PublicKeyDupont.pub`
- c) Envoyer à votre voisin votre clé publique (pas ssh ou clé usb)
- d) Créer un fichier texte `Plain.txt` et y écrire : « Travaux Pratiques SECRES ». Chiffrer le fichier `Plain.txt` avec la clé publique de votre collègue

- e) Demander à votre collègue de chiffrer son fichier `Plain.txt` avec votre clé publique. Déchiffrer le fichier que vous avez reçu avec votre clé privée.
- f) Grâce à la commande `rand`, créer un fichier `GrandFichier.txt` avec des données aléatoires, d'une taille d'environ 1000 Mo (1 Go). Chiffrer le fichier `GrandFichier.txt` avec la clé publique de votre collègue. Vous devez remarquer un problème. Comment l'expliquez-vous ?

Partie 3 : Chiffrement symétrique/asymétrique et signature numérique

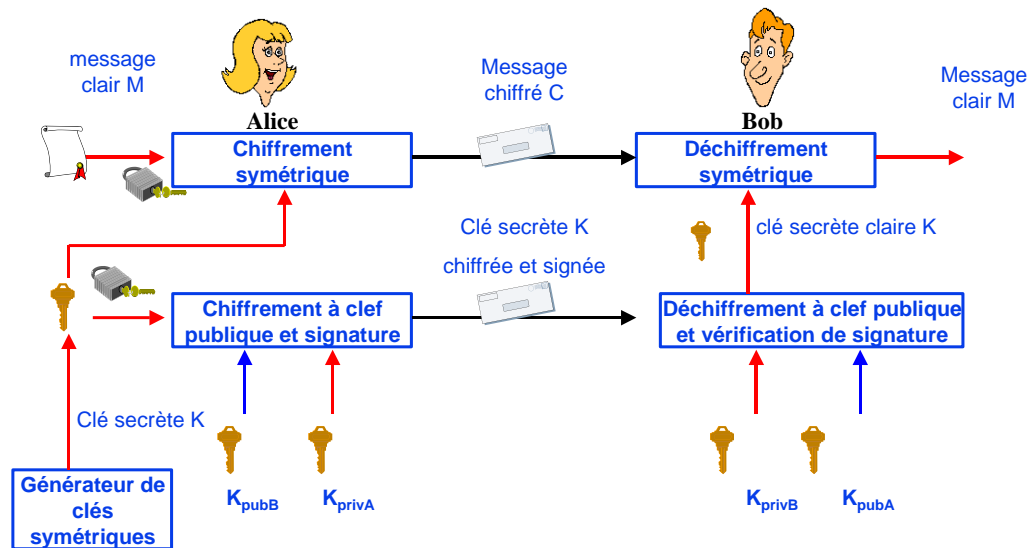


Figure 1. Cryptographie hybride (symétrique et asymétrique)

- a) Expliquer le scénario ci-dessus.
- b) Quels sont les différents services de sécurité réalisés dans ce scénario.
- c) Générer avec la commande `rand` une clé symétrique `SymKey.key` de taille 128 bits. Encoder la clé en hex.
- d) Visualiser la clé symétrique en binaire en utilisant la commande linux : `xxd`
`xxd -b -c 8 SymKey.key`
- e) Réaliser le scénario ci-dessus en échangeant d'une manière sûre la clé symétrique avec votre collègue. Pour information, on signe le hash d'un fichier et non pas le fichier lui-même.

Pour connaître les paramètres des commandes :

```
openssl enc -h
```

```
openssl rand -h
```

```
openssl rsautl -h
```

```
openssl rsa -h
```