Sécurité et Cryptographie :
TP RSA

L'objectif du TP est de manipuler des clés RSA. Vous devrez être capable de générer une paire de clefs RSA l'aide de openssl, et de l'utiliser pour signer et chiffrer des données.

1 Génération d'une paire de clefs RSA

La commande genrsa de openssl permet de générer une paire de clefs RSA :
`$ openssl genrsa -out < fichier> <taille>` Les fichiers produits lors de la création des clés sont au format PEM (Privacy-Enhanced Message).

Question 1 *Générez une bi-clé RSA de 2048 bits dans un fichier .pem*

2 Visualisation des clés RSA

La commande rsa permet de visualiser le contenu d'un fichier au format PEM contenant une bi-clé RSA :
`$ openssl rsa -in < fichier> -text -noout` L'option -text demande l'affichage décodé de la bi-clé. L'option -noout supprime la sortie normalement produite par la commande rsa. L'option -pubout permet d'obtenir en sortie une clé publique (distribuable à tous) au lieu de la clé privée obtenue par défaut.

Question 2 *Afficher le contenu du fichier* `.pem` *avec la commande cat sous Linux, puis avec la commande rsa de openssl. Comparer les différences ?*

Question 3 *Que vaut votre exposant de chiffrement ? comparez avec ceux de vos voisins.*

Question 4 *Utilisez l'option -pubout pour exporter clé publique dans un fichier* `.pub.pem`.

3 Chèrement d'un fichier de clés RSA

Il n'est pas souhaitable de laisser une bi-clé en clair D'habitude on chiffre la clé privée avec un algorithme symétrique (des, aes)

– Soit lors de la création de la bi-clé :

```
$ openssl genrsa -des3 -out ficher.pem 1024
```

– Soit en chiffrant après coup une bi-clé existante avec la commande rsa : `$ openssl rsa -in ficher.pem -des3 -out ficher.pem`

Question 5 *Chiffré votre clé RSA avec un algorithme symétrique ; Afficher le contenu du fichier* `.pem` *puis avec la commande rsa. Essayer différents algorithmes symétriques*

4 Chiffrement, déchiffrement avec RSA

La commande rsautl permet de chiffrer et déchiffrer des données.

```
$ openssl rsautl -encrypt -in <fichier entree> -inkey <cle> -out <fichier sortie>
```

– *fichier entree* est le fichier des données à chiffrer.
– *cle* est le fichier contenant la clé RSA. Si ce fichier ne contient que la clé publique, il faut ajouter l'option -pubin.
– *fichier sortie* est le fichier des données chiffrées.

Pour déchiffrer, on remplace l'option `-encrypt` par `-decrypt`.

Question 6 *Echanger entre vous vos clés publiques et chiffrez de petit message. Envoyez-lez à vos collègues.*

5 Signature avec RSA

Pour signer, on utilise l'option `-sign` de la commande rsautl :

```
$ openssl rsautl -sign -in <fichier a signer > -inkey <cle > -
out <signature >
```

et pour vérifier la signature :

```
$ openssl rsautl -verify -in <signature > -pubin -inkey <cle > -out
  <fichier signe>
```

Question 7 *Signez un petit fichier et échanger le avec vos collègues ?*

6 Empreinte d'un document

Il est possible de calculer une empreinte d'un document avec la commande dgst :

```
$ openssl dgst <hachage > -out <empreinte > <fichier entree >
```

où hachage est une fonction de hachage, comme MD5 (option -md5) qui calcule des empreintes de 128 bits, ou SHA1 (-sha1) de 160 bits.

Question 8 *Signez un gros fichier en utilisant son empreinte*

7 Utilisation des libs openssl

Dans un langage de votre choix utiliser la lib openssl pour

1. Générer des clés et les exporter
2. Chiffrer des messages
3. Déchiffrer des messages