TP1

Chiffrement, signature

Compte rendu

Le TP donnera lieu à la rédaction d'un petit fichier texte contenant les réponses aux questions ainsi que d'éventuels résultats de commandes. Pour cela utiliser un éditeur quelconque et classer les réponses par numéro et utilisez le copier-coller si besoin. Le but est d'obtenir un texte court permettant de vérifier vos acquis.

Le fichier portera votre nom suivi du numéro de TP en l'occurrence 1, exemple : martin1.txt

Le fichier sera envoyé par mail ou sur formation.u-psud.fr (dokeos).

Environnement de travail

Pour ce premier tp qui ne necessite pas de droits root vous pouvez lancer toutes les commandes depuis un terminal sous votre session linux normale.

1. Cryptographie asymétrique

1.1 Génération de clé rsa

On utilisera la commande openssl pour faire quelques essais de chiffrement.

Consulter le manuel man genrsa pour créer une bi –clé rsa de 1024 bits, le fichier sera de type .pem

Afficher le contenu de votre bi-clé à l'aide de la commande :

```
$ openssl rsa –in f.pem –text -noout
```

Question 1

Donner la commande de génération de la clé.

Question 2

Quelle est la longueur des 2 nombres premiers ?

Chiffrer cette clé avec l'algorithme DES3, la commande est :

```
$ openssl rsa –in f.pem –des3 –out f.pem
```

Récupérer un fichier séparé pour la partie publique de la clé :

```
$ openssl rsa –in f.pem –pubout –out f.pub
```

1.2 Chiffrement

Le chiffrement cryptographique permet d'assurer la propriété de confidentialité des données. Par exemple, un fichier transmis sur un réseau est chiffré avec la clé publique de son destinataire ; ainsi, seul le destinataire qui possède la clé privée associée peut déchiffrer le fichier original.

Chiffrer un document avec votre clé publique, par exemple l'entrée standard. Attention, le message est d'abord formaté par une fonction de padding avant d'être chiffré par RSA. Nous utilisons ici le padding OAEP (Optimal

Asymmetric Encryption Padding) qui est actuellement recommandé ; OAEP remplace le padding PKCS contre lequel des attaques sont connues.

Attention, les commandes rsautl ne peuvent chiffrer une taille de message plus grande que la clé. Ceci est présenté à titre d'exemple, on utilisera plutôt gpg dans la pratique pour s'échanger des fichiers chiffrés.

Chiffrer un document avec votre clé publique, par exemple l'entrée standard:

```
$ echo essai|openssl rsautl –pubin -inkey f.pub –encrypt –out secret1 -oaep
```

Refaire la même opération dans un second fichier.

Question 3

Quelle est la taille du fichier chiffré et pourquoi ?

Question 4

Les deux fichiers sont-ils identiques ? Pourquoi ?

Vérifier que le déchiffrement redonne la chaîne de départ, la commande est :

```
$ openssl rsautl –inkey f.pem –in secret –decrypt -oaep
```

Question 5

Pourquoi la commande demande-elle un mot de passe ?

1.3 Signature

La signature cryptographique permet d'assurer la propriété d'intégrité des données. Par exemple, un fichier transmis sur un réseau est signé avec la clé privée de son expéditeur ; ainsi, les destinataires peuvent vérifier la source du message en vérifiant la signature avec la clé publique associée.

On signera un fichier quelconque, par exemple /etc/passwd. A noter que l'on signe en réalité un hash du fichier de départ, il existe plusieurs options pour spécifier le hash, par exemple –sha ou –sha1

Signer le fichier /etc/passwd à l'aide de votre clé privée.

Pour cela utiliser la commande suivante :

```
$ openssl dgst –sha256 –out passwd.sig –sign f.pem /etc/passwd
```

A noter que le choix de hash est ici –sha.

Vérifier la signature à l'aide de la commande suivante:

```
$ openssl dgst –sha256 –signature passwd.sig –verify f_pub /etc/passwd
```

Question 6

Pourquoi la première commande demand-t-elle un mot de passe et pas la seconde ?

Faire une copie du fichier passwd dans votre compte. Modifier ce fichier passwd (par exemple un champ commentaire) et vérifier que la modification est bien détectée.

Question 7

Quel est le message d'erreur ?

Question 8

Peut-on savoir ce qui a été modifié ?

1.4 Echange de fichiers chiffrés et signés

Le but de l'exercice est de simuler un échange de fichier chiffré ainsi que la vérification que c'est le bon fichier avec une signature.

On pourra créer deux répertoires « source » et « destination ». On mettra les clés utilisées pour le chiffrement et la signature dans ces répertoires. Les transferts seront simulés avec la commande de copie cp entre les deux répertoires.

On pourra prendre un petit fichier quelconque comme contenu à transférer.

Les clés privées et publiques peuvent être reprise de l'exercice précédent et les clés manquantes seront créés.

On pourra transférer les clés par la commande cp également, mais il ne faut pas oublier que la partie privée de la clé ne doit jamais être échangée. Donc pas de cp sur les clés privées. Si vous avez besoin de copier la clé privée c'est que votre solution n'est pas correcte.

Question 9

Pour le transfert donner toutes les commandes utilisées pour préparer le fichier à envoyer et pour le recevoir, avec l'explication de la fonction de chaque commande.

Question 10

Pour la vérification de la signature donner toutes les commandes utilisées pour signer le fichier, envoyer la signature puis vérifier cette signature avec l'explication de la fonction de chaque commande.

2. Chiffrement symétrique

Utiliser la commande openssl pour chiffrer avec un algorithme symétrique. Pour avoir la liste des possibilités :

```
#openssl enc –help
```

Pour chiffrer, par exemple avec aes-256-cbc :

```
$ openssl enc –aes-256-cbc –salt –in /etc/passwd –out passwd.c
```

Pour déchiffrer:

```
$ openssl enc –d –aes-256-cbc –in passwd.c
```

Question 11

Pourquoi 2 chiffrements avec le même mot de passe donnent deux fichiers différents ?

Question 12

Remplacer l'option salt par nosalt, quel est le résultat sur deux chiffrement du même fichier ?

Note : Ceci est présenté à titre d'exemple, on utilisera gpg dans la pratique.

Préparer deux fichiers contenant exactement les caractères suivant :

F1 : 12345678abcdefgh

F2 : abcdefgh12345678

Chiffrer ces deux fichiers en des-ecb et nosalt.

Question 13

Comparer les deux fichiers chiffrés.Que constatez-vous ?

Question 14

Donner une explication, en particulier pour le choix des options de chiffrement.