Bootcamp
Openssl API

Part 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>
```

Pour plus d'informations: http://www.openssl.org
RTFM :

```
$ man openssl
```

L'objectif de ce Bootcamp est de se familiariser avec le service de chiffrement en utilisant l'outil OpenSSL.

1. Lancer votre machine virtuelle Linux (ou votre MAC OS)
2. Si OpenSSl n'est pas installé, installez le :

```
# apt-get install openssl
```

3. Vérifiez l'installation de l'outil :

```
$ openssl
OpenSSL>
```

4. Répondez aux questions suivantes:
    a. Quelle est la version d'openssl installée ?
    b. Lister tous les algorithmes de cryptographie présents dans l'outil
5. Pour voir les paramètres d'une fonction donnée, vous pouvez utiliser la commande **help**

```
$ openssl enc -help
```

Part 2: Chiffrement Symétrique

La commande openssl enc permet de chiffrer et déchiffrer des messages. Plus d'informations sur cette commande peuvent être trouvées en tapant openssl enc -h

1. Créez un fichier nommé plain.txt contenant un texte.
2. Chiffrez ce fichier à l'aide de l'algorithme DES-CBC et enregistrez le fichier sous le nom Cipher.txt. Utilisez l'option –k pour saisir le mot de passe symétrique. Quelles sont les autres options pour la spécification de clé
3. Ouvrez le fichier chiffré à l'aide d'un éditeur de texte. Qu'observez-vous au niveau des premiers caractères ? A quoi cela sert-il ?
4. Déchiffrez le fichier ciffré. Utilisez le nom newPlain.txt pour le déchiffrement. Vérifiez que vous récupérez le fichier initial. Pour le vérifier, vous pouvez utiliser la commande diff:

```
$ diff plain.txt newPlain.txt -q
```

5. Pour plus de lisibilité, ajoutez l'option -base64 et rechiffrez le fichier plain.txt. Ouvrez le fichier cipher.txt.
    a. vous devez ajouter -base64 pour déchiffrer le fichier cipher.txt, sinon openssl affichera une erreur.
6. Répétez les questions 2 et 3, mais cette fois, utilisez l'option -p. expliquer les informations obtenues.
7. Chiffrez à nouveau plain.txt (newCipher.txt). Comparez Cipher.txt et newCipher.txt. Justifiez.
8. Répétez ce test encore deux fois en ajoutant l'option -nosalt. Comparez et expliquez les résultats obtenus.
9. À l'aide de la commande rand, créez un fichier de grande taille hugeFile.txt avec des données aléatoires d'environ 1 Go.
10. Utilisez la commande time pour calculer le temps de chiffrement de votre fichier en utilisant RC2, DES, 3DES, AES en mode CBC (prendre le temps user). Comparez les résultats?
11. Comparez le temps de chiffrement avec le temps de déchiffrement des dernières opérations
12. Répétez les questions 10 et 11, mais utilisez des modes d'opération différents (ECB, CBC et CTR). Comparez les résultats.
13. À l'aide de DES-CBC et AES-128-CBC (utilisez l'option -p), chiffrez hugeFile.txt en utilisant la clé 36D1456C26A3670D et l'IV FB22881684E1864D (option -K). Justifiez la taille de la clé et le vecteur d'initialisation dans les deux cas. Essayez avec l'option k, quelle est la différence?

Part 3: Chiffrement Asymétrique

La commande openssl rsautl est utilisée pour chiffrer et déchiffrer les messages. Cette commande vous permet également de signer un fichier et de vérifier sa signature. Elle est également utilisé pour générer des paires de clés. Pour plus d'informations sur cette commande : openssl rsautl –h.

1. Générez une paire de clés RSA 2048 bits que vous nommerez PrivateKeyVotreNom.priv.
   Ex: PrivateKeyAlice.priv.
   a. Du texte est généré lorsque vous créez la clé, pouvez-vous l'expliquer?
   b. Lisez le contenue de la clé à l'aide d'un éditeur de texte
      cat PrivateKeyYourName.priv
   c. Utilisez les options -text -noout pour voir chaque partie de la clé

2. Créez la clé publique associée que vous nommerez PublicKeyVotreNom.pub
   a. Lisez le contenue de la clé à l'aide d'un éditeur de texte
3. Envoyez la clé publique à votre collègue (en utilisant une clé USB, un email ou via une connexion SSH). Ainsi chaque étudiant recevra au moins une clé publique et enverra sa clé publique à au moins un collègue.
   scp user@serverSource:path/to/source user@serverDest:path/to/destination
4. Créez un fichier texte appelé plainVotreNom.txt et insérez-y quelques lignes de texte
   a. Chiffrez le fichier plainVotreNom.txt avec la clé publique de votre collègue et envoyez-le lui
   b. Demandez à votre collègue de chiffrer son fichier texte avec votre clé publique et de vous l'envoyer
5. Déchiffrez le fichier que vous avez reçu avec votre clé privée
6. À l'aide de la commande rand, créez un fichier hugeFile.txt avec des données aléatoires d'environ 1 000 Mo (1 Go).
   a. Chiffrez le fichier avec la clé publique de votre collègue. Vous devriez remarquer un problème. Comment l'expliquez-vous?

Qu'est-ce que le chiffrement hybride?