

Richard MONToux

EPSI BORDEAUX I5

Manipulation des grands principes de la cryptographie

Utilisation de OpenSSL

Questions:

3.1.1 Rappels théoriques:

- Expliquez moi à quoi peut servir une fonction de hachage ? Dans le cas du téléchargement d'un fichier sur un site internet, quel risque cela permet-il de prévenir ?

Une fonction de hachage peut servir principalement à hacher des données afin de les condenser. Ce condensé est de taille fixe et sa valeur diffère suivant la fonction utilisé.

Lors d'un téléchargement, afin de vérifier si le fichier n'a pas été altéré, il suffit de contrôler la valeur du hash du fichier télécharger et la même que le fichier source.

3.1.2 Contrôle d'intégrité d'un téléchargement:

- Téléchargez depuis le site internet officiel³ la dernière version de GNUPG pour linux.

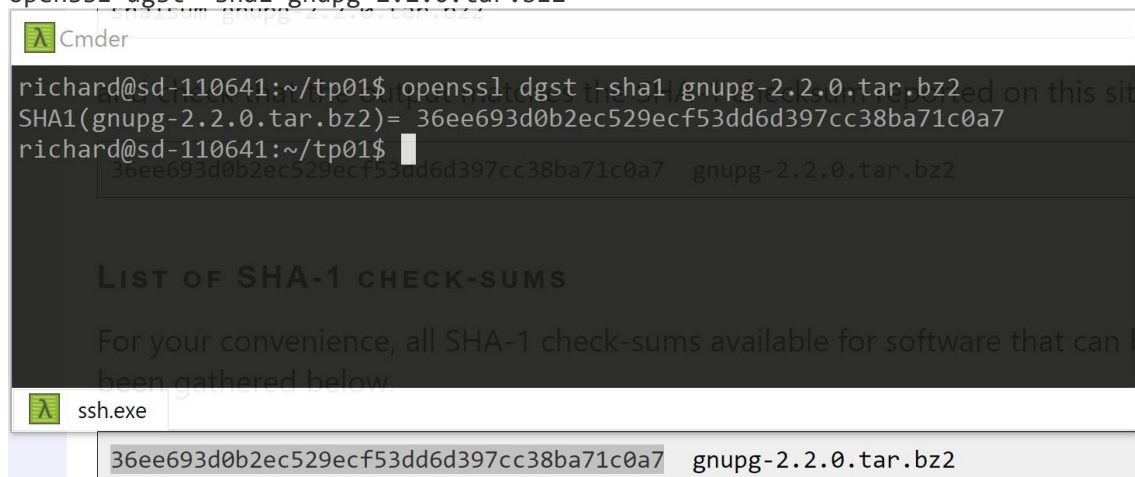
wget https://gnupg.org/ftp/gcrypt/gnupg/gnupg-2.2.0.tar.bz2

- A l'aide de OpenSSL et de l'empreinte SHA1 de l'archive de GNUPG, affichée sur son site officiel⁵, vérifiez que le fichier n'a pas été altéré pendant le téléchargement. Comment avez vous procédé ? Expliquez en détail vos manipulations.

Afin de vérifier que le fichier n'a pas été altéré, je vérifie la clé SHA1 du fichier source avec le fichier téléchargé.

Pour ce faire, j'utilise la commande openssl avec comme option **dgst [digest]** qui permet de définir l'algorithme utilisé, ici on aura sha1.

```
openssl dgst -sha1 gnupg-2.2.0.tar.bz2
```



```
richard@sd-110641:~/tp01$ openssl dgst -sha1 gnupg-2.2.0.tar.bz2
SHA1(gnupg-2.2.0.tar.bz2)= 36ee693d0b2ec529ecf53dd6d397cc38ba71c0a7
richard@sd-110641:~/tp01$
```

LIST OF SHA-1 CHECK-SUMS

For your convenience, all SHA-1 check-sums available for software that can be gathered below

File	SHA-1 Checksum
gnupg-2.2.0.tar.bz2	36ee693d0b2ec529ecf53dd6d397cc38ba71c0a7

3.2.1 Rappels théoriques :

- *Rappelez moi les grands principes du chiffrement symétrique*

Le chiffrement symétrique permet de chiffrer et de déchiffrer un contenu avec la même clé.

Cette clé doit être secrète entre l'émetteur et le destinataire qui se mettent d'accord sur celle-ci ou qu'ils se la transmettent par un autre canal.

3.2.2 Chiffrement de fichier :

- *A partir de votre éditeur de texte favori générez un fichier simple contenant le texte de votre choix*

vim tp01_text.txt

- *A l'aide de OpenSSL chiffrez le contenu de ce fichier. Expliquez en détail votre manipulation. Transmettez ensuite ce fichier à votre voisin, puis la clef utilisée pour le chiffrer*

Afin de chiffrer le fichier avec openssl, j'ai utilisé cette commande:

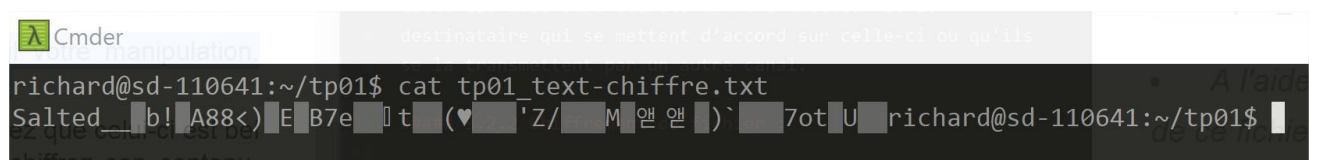
```
openssl enc -e -aes-256-cbc -in tp01_text.txt -out tp01_text-chiffre.txt
```

Cette commande me permet d'encrypter le fichier avec le paramètre **enc** qui permet de dire que l'on va utiliser un algorithme de chiffrement suivi de **-e-** qui définit qu'il s'agit d'un fichier à chiffrer et suivi par l'algorithme d'encrytage.

Ensuite, je récupère le fichier en entré ave **-in** et le nom que je lui donne en sorti **-out**.

- *Votre voisin vous aura transmis un fichier chiffré. Éditez son contenu et constatez que celui-ci est bel et bien chiffré. A l'aide de la clef que vous aurez reçue avec ce fichier, déchiffrez son contenu. Expliquez en détail vos manipulations. Vérifiez avec votre voisin que le contenu que vous avez obtenu est bien celui qu'il a rédigé au départ.*

Cette première image montre bien que le fichier est encrypter.



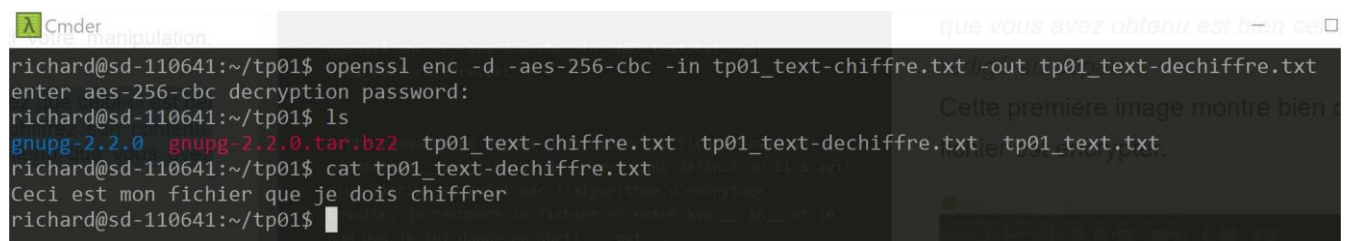
```
richard@sd-110641:~/tp01$ cat tp01_text-chiffre.txt
Salted__b!A88<)E B7e t(♥'Z/M옹 옹)`7otU richard@sd-110641:~/tp01$
```

Afin de pouvoir le lire, je vais utiliser la commande suivante :

```
openssl enc -d -aes-256-cbc -in tp01_text-chiffre -out tp01_text-dechiffre.txt
```

Grace à cette commande je lui dit que l'on va utiliser un algorithme de chiffrement.

Puis on utilise le paramètre **-d** pour lui indiquer que l'on va déchiffrer un fichier.



```
richard@sd-110641:~/tp01$ openssl enc -d -aes-256-cbc -in tp01_text-chiffre.txt -out tp01_text-dechiffre.txt
enter aes-256-cbc decryption password:
richard@sd-110641:~/tp01$ ls
gnupg-2.2.0  gnupg-2.2.0.tar.bz2  tp01_text-chiffre.txt  tp01_text-dechiffre.txt  tp01_text.txt
richard@sd-110641:~/tp01$ cat tp01_text-dechiffre.txt
Ceci est mon fichier que je dois chiffrer
richard@sd-110641:~/tp01$
```

Le fichier est bel et bien déchiffré.

3.2.2 Manipulation des protocoles de chiffrement asymétrique :

3.2.2 Rappels théoriques :

- *Rappelez moi les deux applications principales du chiffrement asymétrique. Expliquez leurs principes de fonctionnement respectifs.*

Chiffrement:

Opération qui consiste à transformer un message à transmettre, dit « message clair », en un autre message, inintelligible pour un tiers, dit « message chiffré », en vue d'assurer le secret de sa transmission

Signature numérique:

Signature reposant sur un système de chiffrement à clé publique et clé privée permettant d'authentifier l'émetteur d'un document. La clé privée sert à signer, la clé publique sert à vérifier cette signature. La signature électronique est l'équivalent numérique de la signature manuscrite.

3.2.2 Préparation de la biclef :


- *Générez pour chacun d'entre vous une paire de clefs RSA 2048bits. Repérez le répertoire où sont stockées vos clefs, puis sauvegardez les.*

Génération de la clé privée RSA 2048:

```
openssl genrsa -out rsa.key 2048
```

Génération de la clé publique RSA 2048

```
openssl rsa -in rsa.key -pubout > rsa.pub
```



```
Cmder
richard@sd-110641:~$ openssl genrsa -out rsa.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
richard@sd-110641:~$ ls
rsa.key
richard@sd-110641:~$ openssl rsa -in rsa.key -pubout > rsa.pub
writing RSA key
richard@sd-110641:~$ ls
rsa.key  rsa.pub
richard@sd-110641:~$
```

- *Pour communiquer avec un interlocuteur vous allez devoir transmettre une de vos clefs. De laquelle s'agit-il ? Échangez alors cette clef avec toutes les personnes avec qui vous allez vouloir communiquer.*

Afin de communiquer avec un interlocuteur, je vais transmettre la clé publique. C'est grâce à celle-ci que l'interlocuteur va pouvoir déchiffrer mon message que j'aurais chiffré avec ma clé privée.

3.3.3 Chiffrement et déchiffrement de fichiers :

- A l'aide de la clef adéquat, chiffrez un fichier texte, à destination d'un interlocuteur précis. Vérifiez à l'aide d'un éditeur le contenu du fichier obtenu. Transmettez le à l'un de vos interlocuteur.

Je vais chiffrer mon fichier avec la clé publique de mon interlocuteur grâce à la commande suivante:

Je commence par utiliser un utilitaire RSA **rsautl** suivi du paramètre **pubin** qui définit le fichier d'entrée de type RSA avec **encrypt** pour dire que l'on va chiffrer le message qui est **in** dans le fichier source et qu'en sortie, nous voulons le nom du fichier suivant.

```
openssl rsautl -pubin -inkey ../rsa.pub -encrypt -in tp01_text.txt -out tp01_text_rsa.txt.crypted
```

```
richard@sd-110641:~/tp01$ openssl rsautl -pubin -inkey ../rsa.pub -encrypt -in tp01_text.txt -out tp01_text_rsa.txt.crypted
richard@sd-110641:~/tp01$ ls
gnupg-2.2.0      tp01_text-chiffre.txt  tp01_text_rsa.txt.crypted
gnupg-2.2.0.tar.bz2 tp01_text-dechiffre.txt tp01_text.txt
richard@sd-110641:~/tp01$
```

Si j'effectue un cat cela me donne :

```
richard@sd-110641:~/tp01$ cat tp01_text_rsa.txt.crypted
U#Qg1 {r4y~YH/"sy=H M-E+KBHJYnk<C...
^Sng
TPV[ [宽宽]-r (richard@sd-110641:~/tp01$ 理理A2h
```

- Récupérez un fichier chiffré par l'un de vos interlocuteur, puis déchiffrez le à l'aide de la clef adéquat.

Lorsque mon interlocuteur va recevoir mon fichier chiffré avec sa clé publique, lui seul pourra le déchiffrer avec sa clé privée. Pour cela, il va utiliser la commande suivante:

En revanche, pour qu'il puisse déchiffrer le fichier, il doit remplacer **encrypt** par **decrypt**.

```
openssl rsautl -decrypt -inkey ../rsa.key -in tp01_text_rsa.txt.crypted -out tp01_text_rsa.txt.uncrypted
```

```
richard@sd-110641:~/tp01$ openssl rsautl -decrypt -inkey ../rsa.key -in tp01_text_rsa.txt.crypted -out tp01_text_rsa.txt.uncrypted
richard@sd-110641:~/tp01$ ls
gnupg-2.2.0      tp01_text-chiffre.txt  tp01_text_rsa.txt.crypted  tp01_text.txt
gnupg-2.2.0.tar.bz2 tp01_text-dechiffre.txt tp01_text_rsa.txt.uncrypted
richard@sd-110641:~/tp01$
```

3.3.4 Signature électronique :

- A l'aide de la clef adéquat, signez un fichier texte. Transmettez ce fichier et sa signature à l'un de vos interlocuteur.

Tout d'abord nous allons calculer l'empreinte avec la commande suivante:

```
openssl dgst -sha256 -out file_sign.txt.sha256 file_sign.txt
```

