TP1 401 : chiffrement :

1 : Fonction de hachage

Le condensé, de par ses propriétés mathématiques, est un outil de choix pour la vérifica- tion de l'intégrité des données. En effet, si le moindre octet est modifié dans le fichier de départ, son condensé change radicalement. En pratique, le condensé permet de vérifier qu'un fichier n'a pas été altéré lors d'une transmission. On s'en sert également pour le stockage des mots de passe sous Linux

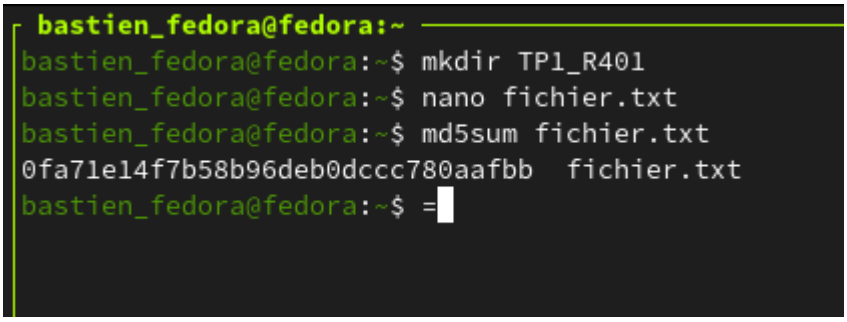
1.1 : Calcul d'un condensé à l'aide de MD5 :

A l'aide d'un éditeur quelconque, créez un fichier contenant un texte quelconque. Sauvez- le, puis calculez son condensé à l'aide de la fonction md5sum :

```
md5sum fichier.txt
```

Exercice 1 : Quel hash obtenez vous ?

```
echo "fichier quelconque" > fichier.txt
md5sum fichier.txt
```



Exercice 2 : Combien de condensés différents sont possibles (Indice : c'est de l'hexadecimal)?

Il y a 128 bits de condensé possible car il y a 16^32 combinaisons possibles.

Exercice 3 : La dernière propriété mathématique énoncée dans Wikipédia est-elle possible en pratique ?

La robustesse de la fonction de hash MD5 est possible en pratique. Néanmoins il est possible de trouver des collisions en pratique.

1.2 : Verification des propriétes du condensé

Exercice 4 Reprenez le fichier précédent et renommez-le. Calculez son hash. Que remarquez vous ?

```
mv fichier.txt fichier2.txt
md5sum fichier2.txt
```

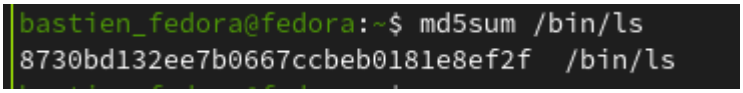
Exercice 5 Modifiez maintenant le contenu du fichier. Recalculez le hash. Que remarquez-vous ?

```
echo "fichier modifié" > fichier2.txt
md5sum fichier2.txt
```

Le code MD5 n'a pas changer, il est toujours le même.

Exercice 6 Peut-on calculer le hash d'un fichier binaire (un executable par exemple) ? Vérifiez votre réponse.

```
md5sum /bin/ls
```



Oui on peut calculer le hash d'un fichier binaire.

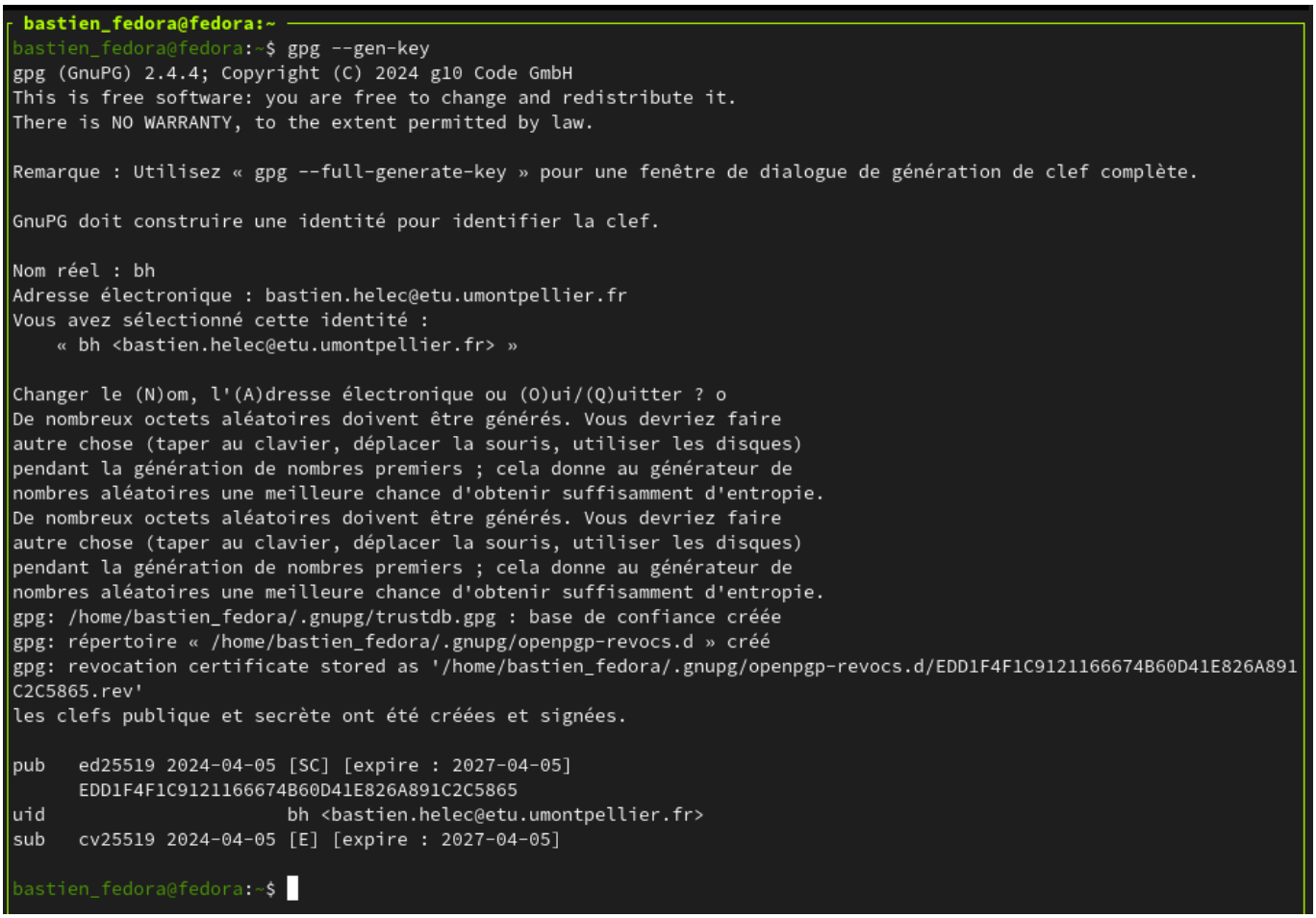
2 Clefs de chiffrement :

2.1 : Génération de clefs :

Nous allons commencer par créer un couple de clés publique/privée associées à votre Email. Pour ce faire, nous allons utiliser gnupg.

Vous allez suivre la procédure suivante : Tapez la commande suivante pour générer la paire de clefs.

```
gpg --gen-key # version simplifié
gpg --full-generate-key # version avec toutes les options
```


bastien_fedora@fedora:~
bastien_fedora@fedora:~$ gpg --gen-key
gpg (GnuPG) 2.4.4; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Remarque : Utilisez « gpg --full-generate-key » pour une fenêtre de dialogue de génération de clef complète.
GnuPG doit construire une identité pour identifier la clef.
Nom réel : bh
Adresse électronique : bastien.helec@etu.umontpellier.fr
Vous avez sélectionné cette identité :
« bh <bastien.helec@etu.umontpellier.fr> »
Changer le (N)om, l'(A)dresse électronique ou (O)ui/(Q)uitter ? o
De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
gpg: /home/bastien_fedora/.gnupg/trustdb.gpg : base de confiance créée
gpg: répertoire « /home/bastien_fedora/.gnupg/openpgp-revocs.d » créé
gpg: revocation certificate stored as '/home/bastien_fedora/.gnupg/openpgp-revocs.d/EDD1F4F1C9121166674B60D41E826A891
C2C5865.rev'
les clefs publique et secrète ont été créées et signées.
pub ed25519 2024-04-05 [SC] [expire : 2027-04-05]
EDD1F4F1C9121166674B60D41E826A891C2C5865
uid bh <bastien.helec@etu.umontpellier.fr>
sub cv25519 2024-04-05 [E] [expire : 2027-04-05]
bastien_fedora@fedora:~$

```
bastien_fedora@fedora:~
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
   (9) ECC (sign and encrypt) *default*
  (10) ECC (signature seule)
  (14) Existing key from card
Quel est votre choix ? 10
Sélectionnez le type de courbe elliptique désiré :
   (1) Curve 25519 *default*
   (4) NIST P-384
   (6) Brainpool P-256
Quel est votre choix ? 4
Veuillez indiquer le temps pendant lequel cette clef devrait être valable.
         0 = la clef n'expire pas
      <n>  = la clef expire dans n jours
      <n>w = la clef expire dans n semaines
      <n>m = la clef expire dans n mois
      <n>y = la clef expire dans n ans
Pendant combien de temps la clef est-elle valable ? (0)
La clef n'expire pas du tout
Est-ce correct ? (o/N) n
Pendant combien de temps la clef est-elle valable ? (0) 1
La clef expire le sam. 06 avril 2024 16:49:54 CEST
Est-ce correct ? (o/N) o

GnuPG doit construire une identité pour identifier la clef.

Nom réel : bh401
Adresse électronique : bastien.helec@etu.umontpellier.fr
Commentaire : gpg full gen key
Vous avez sélectionné cette identité :
    « bh401 (gpg full gen key) <bastien.helec@etu.umontpellier.fr> »

Changer le (N)om, le (C)ommentaire, l'(A)dresse électronique
ou (O)ui/(Q)uitter ? o
De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
gpg: revocation certificate stored as '/home/bastien_fedora/.gnupg/openpgp-revocs.d/56D0FC892FF21D8A01DF7BA298FF28017
386B5DA.rev'
les clefs publique et secrète ont été créées et signées.

pub   nistp384 2024-04-05 [SC] [expire : 2024-04-06]
      56D0FC892FF21D8A01DF7BA298FF28017386B5DA
uid                      bh401 (gpg full gen key) <bastien.helec@etu.umontpellier.fr>
```

passphrase : TP1401

Exercice 7 : Vérifiez que votre trousseau numérique contient bien votre nouvelle clé.

```
gpg --list-keys
```

```
bastien_fedora@fedora:~$ gpg --list-KEY
gpg: invalid option "--list-KEY"
bastien_fedora@fedora:~$ gpg --list-key
gpg: vérification de la base de confiance
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: profondeur : 0  valables :   2  signées :   0
     confiance : 0 i., 0 n.d., 0 j., 0 m., 0 t., 2 u.
gpg: la prochaine vérification de la base de confiance aura lieu le 2024-04-06
[keyboxd]
---------
pub   nistp384 2024-04-05 [SC] [expire : 2024-04-06]
      56D0FC892FF21D8A01DF7BA298FF28017386B5DA
uid           [  ultime ] bh401 (gpg full gen key) <bastien.helec@etu.umontpellier.fr>

pub   ed25519 2024-04-05 [SC] [expire : 2027-04-05]
      EDD1F4F1C9121166674B60D41E826A891C2C5865
uid           [  ultime ] bh <bastien.helec@etu.umontpellier.fr>
sub   cv25519 2024-04-05 [E] [expire : 2027-04-05]
```

quels sont les différents champs ?

pub : clé publique uid : utilisateur sub : clé privée

Exercice 8 : Quelle commande permet de liste les clefs privées présentes sur votre machine ? Comment peut-on savoir comment relier la clef publique et la clef privée ?

```
gpg --list-secret-keys
```

```
bastien_fedora@fedora:~$ gpg --list-secret-key
[keyboxd]
---------
sec   nistp384 2024-04-05 [SC] [expire : 2024-04-06]
      56D0FC892FF21D8A01DF7BA298FF28017386B5DA
uid           [  ultime ] bh401 (gpg full gen key) <bastien.helec@etu.umontpellier.fr>

sec   ed25519 2024-04-05 [SC] [expire : 2027-04-05]
      EDD1F4F1C9121166674B60D41E826A891C2C5865
uid           [  ultime ] bh <bastien.helec@etu.umontpellier.fr>
ssb   cv25519 2024-04-05 [E] [expire : 2027-04-05]
```

On peut relier la clef publique et la clef privée grâce à l'ID de la clef.

2.2 : Diffusion de la clef publique Pour permettre aux gens de vous envoyer des messages chiffrés, il est nécessaire de leur distribuer votre clef publique. Cela pourrait se faire par Email, mais deux questions se poseraient : ▹ Êtes vous sur que l'expéditeur du mail est bien la bonne personne ? ▹ Ne peut-on pas faire cela de manière plus compliquée, donc plus rigolote ?

Des serveurs de clés sont mis gratuitement à disposition des internautes pour la distribution des clefs publiques. Nous allons donc exporter votre clef par ce biais :

```
gpg --keyserver pgp.mit.edu --send-keys 0xXXXXXXXX
```

Il se peut que les serveurs soient très longs en terme de temps d'accès Pour éviter d'être bloqué trop longtemps nous allons exporter la clef et l'envoyer à votre binôme. Pour l'exporter on va utiliser la commande suivante (l'option – armor permet de la sauvegarder au format 7 bits donc en hexadécimal) :

```
gpg --armor --export > clef.pub
```

Exercice 9 : Quel est le contenu de votre fichier ?

```
cat clef.pub
```

Exercice 10 : Il est également possible d'exporter sa clef privée pour la mettre sur son ordinateur (si on ne l'a pas créée sur son ordinateur). Pour celà on utilise la commande suivante :

```
gpg --armor --export-secret-keys > clef.priv
```

3 Chiffrement d'un fichier :

Exercice 11 Pour envoyer un message chiffré à votre binôme, quelle clef faut-il utiliser pour chiffrer le message ?

Pour cela soit on la récupère sur un site de dépôt de clef soit on l'importe à partir d'un fichier reçu.

```
gpg --keyserver pgp.mit.edu --recv-keys <ID_Clef>
```

Ou bien à partir d'un fichier :

```
gpg --import <NomFichierClefPublique.key>
```

Exercice 12 Vérifier que la clef est importée sur votre trousseau numérique.

```
gpg --list-keys
```

 gpg_list

 gpg_list

Exercice 13 Générer un fichier texte "toto.txt" et chiffrez le avec la commande suivante :

```
gpg --armor --recipient <@mail_du_destinataire> --encrypt
↪→ toto.txt
```

Quel est le fichier généré qui correspond à la version chiffrée de toto.txt ?

```
toto.txt.asc
```

Le déchiffrage du fichier se fera avec la clef privée du destinataire. La commande utilisée pour faire cela est :

```
gpg --decrypt toto.txt.asc > toto.txt
```

Exercice 14 Vérifier que le fichier déchiffré est équivalent au fichier initial.

5 Utilisation d'un certificat :

exercice 18 : Effectuer la configuration sur le serveur. Faut-il prévoir quelque chose sur le client ? Vérifier si le trafic est chiffré lors de l'envoi de la page web.

```
curl -v https://localhost
```

 curl

Le trafic est bien chiffré.

exercice 19 : Quels sont d'après vous les risques pour un administrateur réseau d'avoir des utilisateurs qui utilisent principalement des sites en https ?

Les risques sont que les utilisateurs pensent que les sites sont sécurisés alors qu'ils ne le sont pas forcément. Il est possible de faire du phishing en utilisant des sites en https.