

c/ Décryptage des phrases : données chiffrées avec AES256 – le résultat de l'exécution décrypte les messages

```
nathan@nathan-VirtualBox: ~/Bureau/TP1
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --decrypt SSIR.txt.asc
gpg: données chiffrées avec AES256
gpg: chiffré avec 1 phrase secrète
Bonjour les gars!
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --decrypt SSIR.txt.gpg
gpg: données chiffrées avec AES256
gpg: chiffré avec 1 phrase secrète
Bonjour les gars!
nathan@nathan-VirtualBox:~/Bureau/TP1$
```

d/ Chiffre aes

Mdp : Bonjour

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ openssl aes-256-cbc -in SSIR.txt -out SSIR.txt.aes
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
nathan@nathan-VirtualBox:~/Bureau/TP1$ cat SSIR.txt.aes
Salted__B00000
..$0$:@nXB00000$}e0
00000D0nathan@nathan-VirtualBox:~/Bureau/TP1$
```

```
openssl -aes-256-cbc -in SSIR.txt.aes -out déchiffre.aes
```

e/i/ Hachage du fichier avec SHA256

```
nathan@nathan-VirtualBox: ~/Bureau/TP1
```

```
nathan@nathan-VirtualBox:~$ ls
Bureau Documents Images Modèles Musique Public Téléchargements Vidéos
nathan@nathan-VirtualBox:~$ cd Bureau/TP1/
nathan@nathan-VirtualBox:~/Bureau/TP1$ sha256sum
sha224sum sha256sum
nathan@nathan-VirtualBox:~/Bureau/TP1$ sha256sum
sha224sum sha256sum
nathan@nathan-VirtualBox:~/Bureau/TP1$ sha256sum SSIR.txt > Hash.SSIR
nathan@nathan-VirtualBox:~/Bureau/TP1$ cat Hash.SSIR
415994abd21ef159d76643674a4fbf00f7ef07fbe619179ae7aff66f485836f5  SSIR.txt
nathan@nathan-VirtualBox:~/Bureau/TP1$
```

Installer openssh

```
sudo apt install openssh-server
```

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ sudo apt install openssl-server
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  ncurses-term openssl-sftp-server ssh-import-id
Paquets suggérés :
  molly-guard monkeysphere ssh-askpass
Les NOUVEAUX paquets suivants seront installés :
  ncurses-term openssl-server openssl-sftp-server ssh-import-id
0 mis à jour, 4 nouvellement installés, 0 à enlever et 173 non mis à jour.
Il est nécessaire de prendre 688 ko dans les archives.
Après cette opération, 6 010 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://tn.archive.ubuntu.com/ubuntu focal/main amd64 ncurses-term all 6.2-0ubuntu2 [249 kB]
Réception de :2 http://tn.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssl-sftp-server amd64 1:8.2p1-4ubuntu0.1 [51,5 kB]
Réception de :3 http://tn.archive.ubuntu.com/ubuntu focal-updates/main amd64 openssl-server amd64 1:8.2p1-4ubuntu0.1 [377 kB]
Réception de :4 http://tn.archive.ubuntu.com/ubuntu focal/main amd64 ssh-import-id all 5.10-0ubuntu1 [10,0 kB]
688 ko réceptionnés en 5s (141 ko/s)
Préconfiguration des paquets...
Sélection du paquet ncurses-term précédemment désélectionné.
(Lecture de la base de données... 197680 fichiers et répertoires déjà installés.)
```

Creation du tuple

```
nathan@nathan-VirtualBox: ~/Bureau/TP1$ cat tuple.txt
-----BEGIN PGP MESSAGE-----

jA0ECQMcd3qFMNtULKD/0k8BC8b7JGJmAA2c8nmF72SuoPaUUX+9+7uR5qWnjgY
jG9dQaNOPRLyqrDUY9lw4A/uZkikLeICTsdrU81aAJ/LP5nk+UPdhSAEkREdFets
=k9cV
-----END PGP MESSAGE----- ,415994abd21ef159d76643674a4fbf00f7ef07f7be619179ae7aff66f485836f5  SSIR.txt,S
HA256,11428
nathan@nathan-VirtualBox:~/Bureau/TP1$
```

Envoie du tuple

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ sudo scp tuple.txt nathan@192.168.56.102:/home/nathan/Bureau/E
change
[sudo] Mot de passe de nathan :
nathan@192.168.56.102's password:
tuple.txt 100% 106 51.6KB/s 00:00
nathan@nathan-VirtualBox:~/Bureau/TP1$
```

e/ii/ déchiffrement du hash : utilisation de cut avec « , » comme séparateur

```
nathan@nathan-VirtualBox:~$ cut -d, -f1 Bureau/Echange/tuple.txt > SSIR.txt.asc
nathan@nathan-VirtualBox:~$ cat SSIR.txt.asc
-----BEGIN PGP MESSAGE-----

jA0ECQMcd3qFMNtULKD/0k8BC8b7JGJmAA2c8nmF72SuoPaUUX+9+7uR5qWnjgY
jG9dQaNOPRLyqrDUY9lw4A/uZkikLeICTsdrU81aAJ/LP5nk+UPdhSAEkREdFets
=k9cV
-----END PGP MESSAGE-----
```

```
nathan@nathan-VirtualBox:~$ gpg -d SSIR.txt.asc
gpg: données chiffrées avec AES256
gpg: chiffré avec 1 phrase secrète
Bonjour les gars!
```

Vérification de l'intégrité, pour ce faire il faut vérifier que le hash de « Bonjour les gars ! » avec la fonction de hachage SHA256 soit le même que celui reçu. (toutes ces infos sont dans le tuple envoyé)

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ cut -d, -f2 tuple.txt > SSIR2.txt
nathan@nathan-VirtualBox:~/Bureau/Echange$ echo "Bonjour les gars!" > SSIR.txt
nathan@nathan-VirtualBox:~/Bureau/Echange$ sha256sum SSIR.txt > SSIR.txt
nathan@nathan-VirtualBox:~/Bureau/Echange$ cat SSIR.txt SSIR2.txt
415994abd21ef159d76643674a4fbf00f7ef07f7be619179ae7aff66f485836f5  SSIR.txt
415994abd21ef159d76643674a4fbf00f7ef07f7be619179ae7aff66f485836f5  SSIR.txt
```

Le hash est le même donc l'intégrité est vérifiée.

e/iii/ Réponse à l'expéditeur en incrémentant le random envoyé et en écrivant notre nom

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ cat tuple.txt
-----BEGIN PGP MESSAGE-----

jA0ECQMcd3qFMNtULKD/0k8BC8b7JGJmAA2c8nmF72SuoPaUUX+9+7uR5qWnjgY
jG9dQaNOPRLyqrDUY9lw4A/uZkikLeICTsdrU81aAJ/LP5nk+UPdhSAEkREdFets
=k9cV
-----END PGP MESSAGE----- ,415994abd21ef159d76643674a4fbf00f7ef07f7be619179ae7aff6
6f485836f5  SSIR.txt,SHA256,11428
nathan@nathan-VirtualBox:~/Bureau/Echange$ echo "11429,sancho" > rep
11429,sancho
nathan@nathan-VirtualBox:~/Bureau/Echange$ sudo scp rep nathan@192.168.56.101:
/home/nathan/Bureau/TP1
nathan@192.168.56.101's password:
rep 100% 13 11.3KB/s 00:00
nathan@nathan-VirtualBox:~/Bureau/Echange$
```

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ ls
Hash.SSIR rep SSIR.txt SSIR.txt.aes SSIR.txt.asc SSIR.txt.gpg tuple.txt
nathan@nathan-VirtualBox:~/Bureau/TP1$
```

e/iv/ On va légèrement modifier le déchiffré « Bonjour les gars ! » et comparer les hash

Le « B » est remplacé par un « b » et on remarque que les hash sont totalement différents.

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ echo "bonjour les gars!"
bonjour les gars!
nathan@nathan-VirtualBox:~/Bureau/Echange$ echo "bonjour les gars!" > SSIR.txt
nathan@nathan-VirtualBox:~/Bureau/Echange$ sha256sum SSIR.txt
7b7f09ccab0c3fbd98a2aa529c50a2ae2dd8ab642dee659270bd151202ec1bf  SSIR.txt
nathan@nathan-VirtualBox:~/Bureau/Echange$ ^C
nathan@nathan-VirtualBox:~/Bureau/Echange$ shasum SSIR.txt > SSIR.txt
nathan@nathan-VirtualBox:~/Bureau/Echange$ cat SSIR.txt SSIR2.txt
7b7f09ccab0c3fbd98a2aa529c50a2ae2dd8ab642dee659270bd151202ec1bf  SSIR.txt
415994abd21ef159d76643674a4fbf00f7ef07f7be619179ae7aff66f485836f5  SSIR.txt
```

e/v/ Le chiffrement symétrique n'assure que la confidentialité des données, contrairement au chiffrement asymétrique qui permet d'assurer des principes de sécurité supplémentaire.

Une clé symétrique correspond à un échange entre 2 personnes, pour communiquer avec d'autres personnes il faudra une autre clé symétrique. Soit un grand nombre de clé selon le nombre de personnes avec qui on communique

L'utilisation d'une clé unique présente un problème : Communiquer la clé de manière sûre à la personne avec laquelle on souhaite dialoguer. Il est nécessaire de garantir la confidentialité de cette clé. Les échanges qui suivront reposent sur celle-ci. Si une tierce personne accède à la clé, elle pourra lire, modifier, altérer tous les échanges qui s'effectueront entre les 2 protagonistes de départ.

La solution est d'utiliser le chiffrement asymétrique : procédé qui intègre deux clés de chiffrement, une clé publique et une clé privée. La clé de chiffrement du message est appelée clé publique, et la clé de déchiffrement du message est appelée clé privée. Avec une clé publique, l'expéditeur code dans un algorithme de chiffrement un message ne pourra être, décodé que par le destinataire détenteur d'une clé privée.

PS : je précise que lorsque je note « > », en réalité j'effectue un copier collé du contenu du fichier initiale vers celui de destination. Par exemple : sha256 sum SSIR.txt > SSIR.hash ; je copie le hash de SSIR.txt et je le colle dans SSIR.hash car la commande « > » altère l'intégrité des données à cause du caractère invisible de retour à la ligne.

EXERCICE 1 PARTIE 2

a/ Génération du premier jeu de clé

Phrase secrète : CLEF_A

```
nathan@nathan-VirtualBox:~$ gpg --gen-key
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
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 clé complète.

GnuPG doit construire une identité pour identifier la clé.

Nom réel : CLEF_A
Adresse électronique : nat.sanchodelarosa@gmail.com
Vous avez sélectionné cette identité :
  = CLEF_A <nat.sanchodelarosa@gmail.com> =

Changer le (N)on, l'(A)ddresse 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: clé DBE33B1479FDA258 marque de confiance ultime.
gpg: répertoire « /home/nathan/.gnupg/openpgp-revocs.d » créé
gpg: revocation certificate stored as '/home/nathan/.gnupg/openpgp-revocs.d/FC4B
6C1B396EA38C20738ABBDBE33B1479FDA258.rev'
les clés publique et secrète ont été créées et signées.

pub   rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
      FC406C1B396EA38C20738ABBDBE33B1479FDA258
uid    [  ultime ] CLEF_A <nat.sanchodelarosa@gmail.com>
sub    rsa3072 2020-10-11 [E] [expire : 2022-10-11]

nathan@nathan-VirtualBox:~$
```

b/ Affichage de la clé créée

```
nathan@nathan-VirtualBox:~$ gpg --list-keys
/home/nathan/.gnupg/pubring.kbx
-----
pub   rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
      FC406C1B396EA38C20738ABBDBE33B1479FDA258
uid    [  ultime ] CLEF_A <nat.sanchodelarosa@gmail.com>
sub    rsa3072 2020-10-11 [E] [expire : 2022-10-11]

nathan@nathan-VirtualBox:~$
```

c/ Voici le contenu de .gnupg

```
nathan@nathan-VirtualBox:~/.gnupg$ ls
openpgp-revocs.d  pubring.kbx  random_seed
private-keys-v1.d  pubring.kbx~  trustdb.gpg
nathan@nathan-VirtualBox:~/.gnupg$
```

pubring.gpg est notre trousseau de clés publiques (appelé maintenant pubring.kbx) et le rôle de secring.gpg est un trousseau de clé (privées) utilisé dans les versions inférieures de gpg.

d/ Génération d'un certificat de révocation pour CLEF_A

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --output Rev_A --gen-revoke CLEF_A
sec  rsa3072/DBE33B1479FDA258 2020-10-11 CLEF_A <nat.sanchodelarosa@gmail.com>

Faut-il créer un certificat de révocation pour cette clé ? (o/N) o
choisissez la cause de la révocation :
  0 = Aucune cause indiquée
  1 = La clé a été compromise
  2 = La clé a été remplacée
  3 = La clé n'est plus utilisée
  Q = Annuler
(Vous devriez sûrement sélectionner 1 ici)
Quelle est votre décision ? 0
Entrez une description facultative, en terminant par une ligne vide :
> génération pour un tp
>
Cause de révocation : Aucune cause indiquée
génération pour un tp
Est-ce d'accord ? (o/N) o
sortie forcée avec armure ASCII.
Certificat de révocation créé.

Veuillez le déplacer sur un support que vous pouvez cacher ; toute personne
accédant à ce certificat peut l'utiliser pour rendre votre clé inutilisable.
Imprimer ce certificat et le stocker ailleurs est une bonne idée, au cas où le
support devienne illisible. Attention tout de même : le système d'impression
utilisé pourrait stocker ces données et les rendre accessibles à d'autres.

nathan@nathan-VirtualBox:~/Bureau/TP1$
```

e/ Génération d'une paire de clé sur une autre vm

```
nathan@nathan-VirtualBox:~$ gpg --gen-key
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
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.

Non réel : CLEF_B
Adresse électronique : nat.sanchodelarosa@gmail.com
Vous avez sélectionné cette identité :
« CLEF_B <nat.sanchodelarosa@gmail.com> »

Changer le (N)om, l'(A)ddresse électronique ou (O)ui/(O)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: clef 7566AEFA74502939 marquée de confiance ultime.
gpg: répertoire « /home/nathan/.gnupg/openpgp-revocs.d » créé
gpg: révocation certificate stored as '/home/nathan/.gnupg/openpgp-revocs.d/F3B06652F1E4406803
8BC3727566AEFA74502939.rev'
les clefs publique et secrète ont été créées et signées.

pub  rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
     F3B06652F1E44068038BC3727566AEFA74502939
uid      CLEF_B <nat.sanchodelarosa@gmail.com>
sub  rsa3072 2020-10-11 [E] [expire : 2022-10-11]

nathan@nathan-VirtualBox:~$
```

f/ Génération d'un certificat de révocation pour cette paire de clés

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ gpg --output Rev_B --gen-revoke CLEF_B

sec  rsa3072/7566AEFA74502939 2020-10-11 CLEF_B <nat.sanchodelarosa@gmail.com>

Faut-il créer un certificat de révocation pour cette clef ? (o/N) o
choisissez la cause de la révocation :
  0 = Aucune cause indiquée
  1 = La clef a été compromise
  2 = La clef a été remplacée
  3 = La clef n'est plus utilisée
  Q = Annuler
(Vous devriez sûrement sélectionner 1 ici)
Quelle est votre décision ? 0
Entrez une description facultative, en terminant par une ligne vide :
> génération pour tp
>
Cause de révocation : Aucune cause indiquée
génération pour tp
Est-ce d'accord ? (o/N) o
sortie forcée avec armure ASCII.
Certificat de révocation créé.

Veuillez le déplacer sur un support que vous pouvez cacher ; toute personne
accédant à ce certificat peut l'utiliser pour rendre votre clef inutilisable.
Imprimer ce certificat et le stocker ailleurs est une bonne idée, au cas où le
support devienne illisible. Attention tout de même : le système d'impression
utilisé pourrait stocker ces données et les rendre accessibles à d'autres.

nathan@nathan-VirtualBox:~/Bureau/Echange$
```

g/ Export des clés publiques entre les machines ;

De A vers B

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --export -r FC406C1B396EA3BC20738A8B08E33B1479FDA258 > CLEF_A_PUB
nathan@nathan-VirtualBox:~/Bureau/TP1$ scp CLEF_A_PUB nathan@192.168.56.102:/home/nathan/Bureau/Echange
CLEF_A_PUB
100% 1757  864.5KB/s  00:00
```

De B vers A

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ gpg --export -r F3B06652F1E44068038BC3727566AEFA74502939 > CLEF_B_PUB
nathan@nathan-VirtualBox:~/Bureau/Echange$ scp CLEF_B_PUB nathan@192.168.56.101:/home/nathan/Bureau/TP1
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ECDSA key fingerprint is SHA256:bnTU4WwY/Ly0vE0SUWk1z4x+fer/Y9XlHpc22n8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.101' (ECDSA) to the list of known hosts.
nathan@192.168.56.101's password:
CLEF_B_PUB
100% 1757  709.5KB/s  00:00

nathan@nathan-VirtualBox:~/Bureau/Echange$
```

Import des clés publiques

De A et de B

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ gpg --import CLEF_A_PUB
gpg: clef DBE33B1479FDA258 : clef publique « CLEF_A <nat.sanchodelarosa@gmail.com> » importée
gpg: Quantité totale traitée : 1
gpg:   Importées : 1
nathan@nathan-VirtualBox:~/Bureau/Echange$ gpg --list-keys
/home/nathan/.gnupg/pubring.kbx
-----
pub  rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
     F3B06652F1E44068038BC3727566AEFA74502939
uid      [  ultime ] CLEF_B <nat.sanchodelarosa@gmail.com>
sub  rsa3072 2020-10-11 [E] [expire : 2022-10-11]

pub  rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
     FC406C1B396EA3BC20738A8B08E33B1479FDA258
uid      [  inconnue ] CLEF_A <nat.sanchodelarosa@gmail.com>
sub  rsa3072 2020-10-11 [E] [expire : 2022-10-11]
```

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --import CLEF_B_PUB
gpg: clef 7566AEFA74502939 : clef publique « CLEF_B <nat.sanchodelarosa@gmail.com> » importée
gpg: Quantité totale traitée : 1
gpg:   Importées : 1
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --list-keys
/home/nathan/.gnupg/pubring.kbx
-----
pub  rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
     FC406C1B396EA3BC20738A8B08E33B1479FDA258
uid      [  ultime ] CLEF_A <nat.sanchodelarosa@gmail.com>
sub  rsa3072 2020-10-11 [E] [expire : 2022-10-11]

pub  rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
     F3B06652F1E44068038BC3727566AEFA74502939
uid      [  inconnue ] CLEF_B <nat.sanchodelarosa@gmail.com>
sub  rsa3072 2020-10-11 [E] [expire : 2022-10-11]
```


h/ J'attribue une confiance totale aux clés importées et les signes pour certifier leur authenticité.

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --edit-key F3BD6652F1E4406B03BBC3727566AEFA74502939
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

pub rsa3072/7566AEFA74502939
   créé : 2020-10-11  expire : 2022-10-11  utilisation : SC
   confiance : inconnu  validité : inconnu
sub rsa3072/0E0D860BBF897206
   créé : 2020-10-11  expire : 2022-10-11  utilisation : E
   [ inconnue ] (1). CLEF_B <nat.sanchodelarosa@gmail.com>

gpg> sign

pub rsa3072/7566AEFA74502939
   créé : 2020-10-11  expire : 2022-10-11  utilisation : SC
   confiance : inconnu  validité : inconnu
   Empreinte clef princip. : F3BD 6652 F1E4 406B 03BB C372 7566 AEFA 7450 2939
   CLEF_B <nat.sanchodelarosa@gmail.com>

Cette clef va expirer le 2022-10-11.
Voulez-vous vraiment signer cette clef avec votre
clef « CLEF_A <nat.sanchodelarosa@gmail.com> » (D0E33B1479FDA258)

Voulez-vous vraiment signer ? (o/N) o

gpg> trust
pub rsa3072/7566AEFA74502939
   créé : 2020-10-11  expire : 2022-10-11  utilisation : SC
   confiance : inconnu  validité : inconnu
sub rsa3072/0E0D860BBF897206
   créé : 2020-10-11  expire : 2022-10-11  utilisation : E
   [ inconnue ] (1). CLEF_B <nat.sanchodelarosa@gmail.com>

Décidez maintenant de la confiance que vous portez en cet utilisateur pour
vérifier les clefs des autres utilisateurs (en regardant les passeports, en
vérifiant les empreintes depuis diverses sources, etc.)

1 = je ne sais pas ou n'ai pas d'avis
2 = je ne fais PAS confiance
3 = je fais très légèrement confiance
4 = je fais entièrement confiance
5 = j'attribue une confiance ultime
n = retour au menu principal

Quelle est votre décision ? 5
Voulez-vous vraiment attribuer une confiance ultime à cette clef ? (o/N) o

pub rsa3072/7566AEFA74502939
   créé : 2020-10-11  expire : 2022-10-11  utilisation : SC
   confiance : ultime  validité : inconnu
sub rsa3072/0E0D860BBF897206
   créé : 2020-10-11  expire : 2022-10-11  utilisation : E
   [ inconnue ] (1). CLEF_B <nat.sanchodelarosa@gmail.com>
Veuillez remarquer que la validité affichée pour la clef n'est pas
forcément correcte avant d'avoir relancé le programme.

gpg>
gpg: signal interrupt caught ... exiting
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --list-keys
```

Chiffrage d'un fichier depuis la machine A avec la clé publique de B

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --encrypt --armor -r F3BD6652F1E4406B03BBC3727566AEFA74502939 SSIR.txt
Le fichier « SSIR.txt.asc » existe. Faut-il réécrire par-dessus ? (o/N) o
nathan@nathan-VirtualBox:~/Bureau/TP1$ scp SSIR.txt.asc nathan@192.168.56.102:/home/nathan/Bureau/Echange
nathan@192.168.56.102's password:
SSIR.txt.asc
100% 720 206.4KB/s 00:00
```

i/ Déchiffrage du fichier depuis la machine B

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ gpg -d SSIR.txt.asc
gpg: chiffré avec une clef RSA de 3072 bits, identifiant 0E0D860BBF897206, créée le 2020-10-11
« CLEF_B <nat.sanchodelarosa@gmail.com> »
Bonjour les gars!
nathan@nathan-VirtualBox:~/Bureau/Echange$
```

Envoi de message de confirmation à A

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ echo "Bien reçu" > rep_CLEA
nathan@nathan-VirtualBox:~/Bureau/Echange$ gpg --encrypt --armor -r FC406C1B396EA38C20738ABDBE33B1479FDA258 rep_CLEA
Le fichier « rep_CLEA.asc » existe. Faut-il réécrire par-dessus ? (o/N) o
nathan@nathan-VirtualBox:~/Bureau/Echange$ scp rep_CLEA.asc nathan@192.168.56.101:/home/nathan/Bureau/TP1
nathan@192.168.56.101's password:
rep_CLEA.asc
100% 711 221.2KB/s 00:00
nathan@nathan-VirtualBox:~/Bureau/Echange$
```

Déchiffrage de la réponse

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg -d rep_CLEA.asc
gpg: chiffré avec une clef RSA de 3072 bits, identifiant 9A15720E02734F0C, créée le 2020-10-11
« CLEF_A <nat.sanchodelarosa@gmail.com> »
Bien reçu
nathan@nathan-VirtualBox:~/Bureau/TP1$
```

Le message de confirmation est déchiffré avec succès

j/ On commence par supprimer les clés de l'autre machine du système car si on l'a déjà on a pas besoin de la récupérer à nouveau

```
nathan@nathan-VirtualBox:~$ gpg --delete-keys F3BD6652F1E4406B03BBC3727566AEFA74502939
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

pub rsa3072/7566AEFA74502939 2020-10-11 CLEF_B <nat.sanchodelarosa@gmail.com>

Faut-il supprimer cette clef du porte-clefs ? (o/N) o
```

Ensuite, on upload la clé sur le serveur pgp.mit.edu

```
nathan@nathan-VirtualBox:~$ gpg --keyserver pgp.mit.edu --send-keys FC406C1B396EA38C20738ABDBE33B1479FDA258
gpg: envoi de la clef DBE33B1479FDA258 à hkp://pgp.mit.edu
nathan@nathan-VirtualBox:~$
```

```
nathan@nathan-VirtualBox:~$ gpg --keyserver pgp.mit.edu --send-keys F3BD6652F1E4406B03BBC3727566AEFA74502939
gpg: envoi de la clef 7566AEFA74502939 à hkp://pgp.mit.edu
nathan@nathan-VirtualBox:~$
```

Maintenant, on récupère les clés publiques

```
nathan@nathan-VirtualBox:~$ gpg --keyserver pgp.mit.edu --recv-keys 7566AEFA74502939
gpg: clef 7566AEFA74502939 : clef publique « CLEF_B <nat.sanchodelarosa@gmail.com> » importée
gpg: Quantité totale traitée : 1
gpg:      importées : 1
nathan@nathan-VirtualBox:~$
```

```
nathan@nathan-VirtualBox:~$ gpg --keyserver pgp.mit.edu --recv-keys DBE33B1479FDA258
gpg: clef DBE33B1479FDA258 : clef publique « CLEF_A <nat.sanchodelarosa@gmail.com> » importée
gpg: Quantité totale traitée : 1
gpg:      importées : 1
nathan@nathan-VirtualBox:~$
```

On peut vérifier que l'importation est réussie en tapant gpg --list-key

```
nathan@nathan-VirtualBox:~$ gpg --list-key
/home/nathan/.gnupg/pubring.kbx
-----
pub   rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
      FC406C1B396EA38C20738ABDBE33B1479FDA258
uid   [  ultime ] CLEF_A <nat.sanchodelarosa@gmail.com>
sub   rsa3072 2020-10-11 [E] [expire : 2022-10-11]

pub   rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
      F3BD6652F1E4406B03BBC3727566AEFA74502939
uid   [ inconnue ] CLEF_B <nat.sanchodelarosa@gmail.com>
sub   rsa3072 2020-10-11 [E] [expire : 2022-10-11]
```

```
nathan@nathan-VirtualBox:~$ gpg --list-key
/home/nathan/.gnupg/pubring.kbx
-----
pub   rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
      F3BD6652F1E4406B03BBC3727566AEFA74502939
uid   [  ultime ] CLEF_B <nat.sanchodelarosa@gmail.com>
sub   rsa3072 2020-10-11 [E] [expire : 2022-10-11]

pub   rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
      FC406C1B396EA38C20738ABDBE33B1479FDA258
uid   [ inconnue ] CLEF_A <nat.sanchodelarosa@gmail.com>
sub   rsa3072 2020-10-11 [E] [expire : 2022-10-11]
```

C'est validé.

k/ génération d'un nombre aléatoire que l'on signe avec la clé privé

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ echo $RANDOM > SSIR.txt
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --sign SSIR.txt
Le fichier « SSIR.txt.gpg » existe. Faut-il réécrire par-dessus ? (o/N) o
nathan@nathan-VirtualBox:~/Bureau/TP1$
```

Chiffage de ce fichier avec la clé publique de B

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --encrypt --armor -r F3BD6652F1E4406B03BBC3727566AEFA74502939 SSIR.txt
gpg: 0E0D8608BF897206 : aucune assurance que la clef appartienne vraiment à l'utilisateur nommé.

sub   rsa3072/0E0D8608BF897206 2020-10-11 CLEF_B <nat.sanchodelarosa@gmail.com>
      Empreinte clef princip. : F3BD 6652 F1E4 406B 03BB C372 7566 AEFA 7450 2939
      Empreinte de sous-clef : 0FFE 2F81 5E7A E14F 072F 95E0 0E0D 8608 BF89 7206

La clef n'appartient PAS forcément à la personne nommée
dans l'identité. Si vous savez "vraiment" ce que vous
faites, vous pouvez répondre oui à la prochaine question.

Faut-il quand même utiliser cette clef ? (o/N) o
Le fichier « SSIR.txt.asc » existe. Faut-il réécrire par-dessus ? (o/N) o
nathan@nathan-VirtualBox:~/Bureau/TP1$
```


Envoi du fichier à B

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ scp SSIR.txt.asc nathan@192.168.56.102:/home/nathan/Bureau/Echange
nathan@192.168.56.102's password:
SSIR.txt.asc 100% 703 165.8KB/s 00:00
```

I/ Déchiffrement du fichier reçu par B

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ gpg --decrypt SSIR.txt.asc
gpg: chiffré avec une clef RSA de 3072 bits, identifiant 0E0D860BBF897206, créée le 2020-10-11
« CLEF_B <nat.sanchodelarosa@gmail.com> »
20455
```

On va signer et certifier la signature sur les deux machines pour ne plus avoir d'avertissements

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ gpg --edit-key FC406C1B396EA38C20738AB80BE33B1479FDA258
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

pub rsa3072/DBE33B1479FDA258
   créé : 2020-10-11 expire : 2022-10-11 utilisation : SC
   confiance : inconnu validité : inconnu
sub rsa3072/9A15720E02734F0C
   créé : 2020-10-11 expire : 2022-10-11 utilisation : E
   [ inconnue ] (1). CLEF_A <nat.sanchodelarosa@gmail.com>

gpg> sign

pub rsa3072/DBE33B1479FDA258
   créé : 2020-10-11 expire : 2022-10-11 utilisation : SC
   confiance : inconnu validité : inconnu
   Empreinte clef princip. : FC40 6C1B 396E A38C 2073 8ABB 0BE3 3B14 79FD A258
   CLEF_A <nat.sanchodelarosa@gmail.com>

Cette clef va expirer le 2022-10-11.
Voulez-vous vraiment signer cette clef avec votre
clef « CLEF_B <nat.sanchodelarosa@gmail.com> » (7566AEFA74502939)

Voulez-vous vraiment signer ? (o/N) o
gpg> q
Faut-il enregistrer les modifications ? (o/N) o

nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --edit-key F3BD6652F1E4406803B8C3727566AEFA74502939
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

pub rsa3072/7566AEFA74502939
   créé : 2020-10-11 expire : 2022-10-11 utilisation : SC
   confiance : inconnu validité : inconnu
sub rsa3072/0E0D860BBF897206
   créé : 2020-10-11 expire : 2022-10-11 utilisation : E
   [ inconnue ] (1). CLEF_B <nat.sanchodelarosa@gmail.com>

gpg> dign

Commande incorrecte (essayez « help »)

gpg> sign

pub rsa3072/7566AEFA74502939
   créé : 2020-10-11 expire : 2022-10-11 utilisation : SC
   confiance : inconnu validité : inconnu
   Empreinte clef princip. : F3BD 6652 F1E4 4068 03B8 C372 7566 AEFA 7450 2939
   CLEF_B <nat.sanchodelarosa@gmail.com>

Cette clef va expirer le 2022-10-11.
Voulez-vous vraiment signer cette clef avec votre
clef « CLEF_A <nat.sanchodelarosa@gmail.com> » (DBE33B1479FDA258)

Voulez-vous vraiment signer ? (o/N) o
gpg> q
Faut-il enregistrer les modifications ? (o/N) o
```

Vérifier que la clé est signée

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --list-sigs
gpg: vérification de la base de confiance
gpg: marginals needed: 3 completes needed: 1 trust model: gpg
gpg: profondeur : 0 variables : 1 signées : 1
   confiance : 0 i., 0 n.d., 0 j., 0 m., 0 t., 1 u.
gpg: profondeur : 1 variables : 1 signées : 0
   confiance : 1 i., 0 n.d., 0 j., 0 m., 0 t., 0 u.
gpg: la prochaine vérification de la base de confiance aura lieu le 2022-10-11
/home/nathan/.gnupg/pubring.kbx
-----
pub rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
   FC406C1B396EA38C20738AB80BE33B1479FDA258
uid [ ultime ] CLEF_A <nat.sanchodelarosa@gmail.com>
sig 3 DBE33B1479FDA258 2020-10-11 CLEF_A <nat.sanchodelarosa@gmail.com>
sub rsa3072 2020-10-11 [E] [expire : 2022-10-11]
sig DBE33B1479FDA258 2020-10-11 CLEF_A <nat.sanchodelarosa@gmail.com>

pub rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
   F3BD6652F1E4406803B8C3727566AEFA74502939
uid [ totale ] CLEF_B <nat.sanchodelarosa@gmail.com>
sig 3 7566AEFA74502939 2020-10-11 CLEF_B <nat.sanchodelarosa@gmail.com>
sig DBE33B1479FDA258 2020-10-13 CLEF_A <nat.sanchodelarosa@gmail.com>
sub rsa3072 2020-10-11 [E] [expire : 2022-10-11]
sig 7566AEFA74502939 2020-10-11 CLEF_B <nat.sanchodelarosa@gmail.com>
```

On remarque que tout est ok

m/ On peut automatiser ces vérifications de signature avec un script shell

```
#!/bin/sh
echo "Récupération des nouvelles signatures des clefs hébergé dans le serveur qui ont partie de notre trousseau"
for i in $(/usr/bin/gpg --list-keys | grep 'pub' | cut -c 13-20); \
do /usr/bin/gpg --keyserver pgp.mit.edu --recv-key $i; done
~
~
~
```

n/ On vérifie à nouveau la signature

```
nathan@nathan-VirtualBox:~$ gpg --check-sig FC406C1B396EA38C20738ABDBE33B1479FDA258
pub rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
    FC406C1B396EA38C20738ABDBE33B1479FDA258
uid [ totale ] CLEF_A <nat.sanchodelarosa@gmail.com>
sig!3 DBE33B1479FDA258 2020-10-11 CLEF_A <nat.sanchodelarosa@gmail.com>
sig! 7566AEFA74502939 2020-10-13 CLEF_B <nat.sanchodelarosa@gmail.com>
sub rsa3072 2020-10-11 [E] [expire : 2022-10-11]
sig! DBE33B1479FDA258 2020-10-11 CLEF_A <nat.sanchodelarosa@gmail.com>

gpg: 3 good signatures
```

C'est good

o/ Envoie de la réponse signé à A

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ echo "20456" > SSIR.txt
nathan@nathan-VirtualBox:~/Bureau/Echange$ gpg --sign SSIR.txt
```

```
nathan@nathan-VirtualBox:~/Bureau/Echange$ gpg --encrypt --armor -r FC406C1B396EA38C20738ABDBE33B1479FDA258
SSIR.txt
Le fichier « SSIR.txt.asc » existe. Faut-il réécrire par-dessus ? (o/N) o
nathan@nathan-VirtualBox:~/Bureau/Echange$ scp SSIR.txt.asc nathan@192.168.56.101:/home/nathan/Bureau/TP1
nathan@192.168.56.101's password:
SSIR.txt.asc
100% 703 427.6KB/s 00:00
nathan@nathan-VirtualBox:~/Bureau/Echange$
```

La machine A a déjà les signatures de certifiées je ne le refais donc pas

p/Différentes versions de vérifications des signatures :

gpg --detach-sign

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --detach-sign SSIR.txt.asc
```

Cette commande détache la signature du message et crée un autre fichier signature SSIR.txt.asc.sig

```
SSIR.txt.asc.sig
```

gpg --clearsign

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --clearsign SSIR.txt.asc
```

Cette commande crée un nouveau fichier dans lequel on voit une séparation entre le message et la clé

```
SSIR.txt.asc.asc
```

gpg --sign

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --sign SSIR.txt.asc
```

Cette commande crée un nouveau fichier dans lequel on a un mélange entre la clé et le message crypté.

La différence entre les trois est qu'avec detach-sign, la signature du message est détachée dans un autre fichier ; avec clearsign, la signature est détachée mais dans le même fichier et --sign, c'est un mélange entre message et signature.

```
1 -----BEGIN PGP SIGNED MESSAGE-----
2 Hash: SHA512
3
4 -----BEGIN PGP MESSAGE-----
5
6 hQKAS0Vcgc4Cc08MAQw4Cpwcenq0fW083014f/y+/B908thdLk010PfoYV606pt
7 nLNP5uqy9EgUtuFz2jW4++u1y5ahnJ5huEh0Z+e5Jvz50VJ/c0d585kto
8 Ap03htNMCTBwocGp8ndxXpogPKxr4LzHZUqnbFYy3IFncP8z+/wqxUBSpCC0hr
9 65C1pKVPPptMen4eUARnolHcExe68FASzqDumJHVZFOYIXbJL0nak7lp7PT8/du
10 0+ahdwyIp31+f4p/+0A2eVUYtwnFLdIA+JzfyV01e4DQ0yJX05SHBCpLZLuKA
11 L/unchofT2Z1q5Tdepggk1tqzELhvvWtDemp/4qdKsc4/g0D0b3wGulY0Idp
12 IqUlyXNqENlpx+9wagTc0nssSDHS/APuCANBBYfRXSgcKQ3Uw+UHSZ3FRxmfUz
13 uLFK9X50Y4ADYUyKNIY/HwelnBamCA3hyL4AAAL52nI0tCKnGICU7G7wVR16
14 INhLdb20VEJLD/SLGUL0k83EF7YTHSND05Shdkc4LhYhNlRdkZyDQnsJfQ5zFL
15 3fagzA4jIdr7NYtxB8C1vby10BUJ/3x6UR/3kdVFP5vVxkhy7mFg
16 =GR3B
17 - - - - -END PGP MESSAGE-----
18 -----BEGIN PGP SIGNATURE-----
19
20 1QgzBAEBCgAdFLEE/EBsGzLuo4wgc4q72+H7Fhn90lgFAl+PhRMACgkQ2+H7Fhn9
21 01JTfGwApYauvneIFQ5b3Bnd1TCpDatG0EBvBVBt:3roz5YqY8r96gltqdaranL
22 TtrQKjC5qz40nz5Dz3cy4Dszew04b4+9sBwqz/FlswJ5y9b00UeLmly2QFp0px
23 TqGwc02ueI/ax+g5tspvKfSEvrvhaxUE/Q3T435WjpmYwtS05C96NCY2wD5+B40
24 QKZ/Zen5LYMwvncExd5zqHfEble5xyltezpnIGdITGAljy7JlmpZ7/kuz3JuenL
25 bnhLXNqZ7ACDNLvz2AyH2KCLyAurVuelVlpKCS005Zm5L2018Juf+xyfYff
26 VEJ50uKusG1wv/Bhhy4bfrxsUHL2DrddcLV3Re01kqNj431o4cP06Yq3PvgJk
27 0qtXb4sx7LZtbUUVXoFJAfKjV178n/umnFbG0m3Dmf0161Sorwt9BdrKK3p
28 USnBQhokB5AhVeqGuT8DhbyV2Fbq5NZcbuzgZF50nU8+wdhL9HcbEwLhptZotg
29 K5SzC06
30 =ZAGH
31 -----END PGP SIGNATURE-----
```

```
nathan@nathan-VirtualBox: ~/Bureau/TP1$ gpg --verify SSIR.txt.asc.sig
gpg: les données signées sont supposées être dans « SSIR.txt.asc »
gpg: Signature faite le mar. 13 oct. 2020 11:41:07 CET
gpg:          avec la clef RSA FC406C1B396EA38C20738ABDBE33B1479FDA258
gpg: Bonne signature de « CLEF_A <nat.sanchodelarosa@gmail.com> » [ultime]
nathan@nathan-VirtualBox: ~/Bureau/TP1$
```

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ gpg --decrypt SSIR.txt.asc
gpg: chiffré avec une clef RSA de 3072 bits, identifiant 9A15720E02734F0C, créée le 2020-10-11
« CLEF_A <nat.sanchodelarosa@gmail.com> »
20456
```

```

install -pgo $(cat /dev/urandom) -m0555 /usr/bin/$(hostname)
nsh #rm /usr/bin/$(hostname) && gpg --keyid F8076028744840808387477506AF474A72939
gpg (GnuPG) 2.0.18: warning: This program has been flagged as not a security program.
This is free software; you are free to change and redistribute it.
There is NO warranty, to the extent permitted by law.

nsh #rm057 /usr/bin/F8076028744840808387477506AF474A72939
SC
confname : /usr/bin/$(hostname)
valite : totale
nsh #rm057 /usr/bin/F8076028744840808387477506AF474A72939
crsf : 2020-10-11 00:00:00 : 2022-10-11 00:00:00 : 1
t totale : (1). ClsF a met: sanchoukine@saugui.com

gpg> true
nsh #rm057 /usr/bin/F8076028744840808387477506AF474A72939
crsf : 2020-10-11 00:00:00 : 2022-10-11 00:00:00 : 1
confname : /usr/bin/$(hostname)
valite : totale
nsh #rm057 /usr/bin/F8076028744840808387477506AF474A72939
crsf : 2020-10-11 00:00:00 : 2022-10-11 00:00:00 : 1
t totale : (1). ClsF a met: sanchoukine@saugui.com

Déclarer maintenant la confiance aux deux portés en cet utilisateur pour
vérifier les clés des autres utilisateurs.
Cela signifie que vous acceptez les données des utilisateurs, en
vérifiant les empreintes données diverses sources, etc.)

1 = Je me fiais pas un p'tin peu d'avis
2 = Je me fiais pas du tout
3 = Je fais très légèrement confiance
4 = Je fais entièrement confiance
1 = j'attribue une confiance illite
n = retour au menu principal

Quelle est votre décision ?
Veuillez maintenant attribuer une confiance illite à cette clé ? (n/0) n

nsh #rm057 /usr/bin/F8076028744840808387477506AF474A72939
crsf : 2020-10-11 00:00:00 : 2022-10-11 00:00:00 : 1
confname : /usr/bin/$(hostname)
valite : totale
nsh #rm057 /usr/bin/F8076028744840808387477506AF474A72939
crsf : 2020-10-11 00:00:00 : 2022-10-11 00:00:00 : 1
t totale : (1). ClsF a met: sanchoukine@saugui.com
voilà ! remarquez que la validité d'attribuer la confiance à cet us
est forcément correcte avant d'avoir lancé le programme.

```

Au final, je me suis uniquement servi de trust

ii/ installation des packages graphviz et sig2dot

```

nath@ubuntu:~$ curl -fsSL https://raw.githubusercontent.com/rafaelbarros/bureau7775 -o /dev/null && sudo apt-get install -y graphviz
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  libbonobo libcairo libgraphite libgts-0.7-5 libgts-bin libgvc libgvpr2 liblab-gamuti libpathplan4
Paquets suggérés :
  gsfonts graphviz-doc
Les NOUVEAUX paquets suivants seront installés :
  graphviz libbonobo libcairo libgraphite libgts-0.7-5 libgts-bin libgvc libgvpr2 liblab-gamuti libpathplan4
0 à 2 Ko de nouveaux fichiers temporaires seront créés.
Il est nécessaire de redéfinir 1 888 ko dans les archives.

```

[illegible]

Pour ce faire, il faut installer le logiciel sig2dot. Il y a un tutoriel d'installation de l'environnement en suivant ce lien que j'illustre par les images suivantes. [Sig2dot](#)

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ git clone https://github.com/bnmh/sig2dot2
2.git
Clonage dans 'sig2dot2'...
remote: Enumerating objects: 199, done.
remote: Total 199 (delta 0), reused 0 (delta 0), pack-reused 199
Réception d'objets: 100% (199/199), 51.19 Kio | 200.00 Kio/s, fait.
Résolution des deltas: 100% (107/107), fait.
nathan@nathan-VirtualBox:~/Bureau/TP1$
```

```
nathan@nathan-VirtualBox:~/Bureau/TP1$ ls
CLEF_A_PUB rev_CLEA sig2dot2 SSIR.txt.asc SSIR.txt.gpg
CLEF_B_PUB rev_CLEA.asc SSIR_CLEB SSIR.txt.asc.asc tuple.txt
Hash_SSIR Rev_A SSIR.txt SSIR.txt.asc.gpg
rev script SSIR.txt.asc SSIR.txt.asc.sig
nathan@nathan-VirtualBox:~/Bureau/TP1$ cd sig2dot2/
nathan@nathan-VirtualBox:~/Bureau/TP1/sig2dot2$ ls
COPYING README.md setup.py sig2dot2 tests TODO tox.ini
nathan@nathan-VirtualBox:~/Bureau/TP1/sig2dot2$ cd sig2dot2/
nathan@nathan-VirtualBox:~/Bureau/TP1/sig2dot2$ ls
__init__.py doc sig2dot2
nathan@nathan-VirtualBox:~/Bureau/TP1/sig2dot2$ mv sig2dot2.py ./
nathan@nathan-VirtualBox:~/Bureau/TP1/sig2dot2$ cd ..
nathan@nathan-VirtualBox:~/Bureau/TP1$ cd sig2dot2/
nathan@nathan-VirtualBox:~/Bureau/TP1/sig2dot2$ ls
__init__.py gpg __init__.py sig2dot2
nathan@nathan-VirtualBox:~/Bureau/TP1/sig2dot2$ cd ..
nathan@nathan-VirtualBox:~/Bureau/TP1$ cd sig2dot2/
nathan@nathan-VirtualBox:~/Bureau/TP1/sig2dot2$ ls
COPYING README.md setup.py sig2dot2 sig2dot2.py tests TODO tox.ini
```

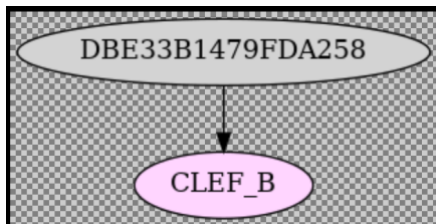
```

jathan@hacker:~/src/FreeBSD$ -Bureau/TP3/sig2dot25 pip3 install .
Processing /home/jathan/Bureau/TP3/sig2dot25
Collecting sig08601
  Downloading sig08601-0.1.13-py2.py3-none-any.whl (9.3 kB)
Building wheels for collected packages: sig2dot
  Building wheel for sig2dot: python setup.py bdist_wheel -s . done
  Created wheel for sig2dot: filename=sig2dot-0.1.2-py3-none-any.whl size=25449
sha256=8f29f9ee1fcec6b4338b411714236539e2486f722b65db474b4d83b5d1
Stored in directory: /tmp/pip-ephem-wheel-cache-q4lfnfsc/wheels/ef/3c/f9/c04e
240c34c3bee2a37c7c24c0c03b79f3f03f67e27
Successfully built sig2dot
Installing collected packages: sig08601, sig2dot
WARNING: The script sig2dot is installed in '/home/jathan/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning,
use --no-manifest-script-location.
Successfully installed sig08601-0.1.13 sig2dot-0.1.2
jathan@hacker:~/src/FreeBSD$ cd /Bureau/TP3/sig2dot25
jathan@hacker:~/src/FreeBSD$ cd /Bureau/TP3/sig2dot25

```

```
nathan@nathan-VirtualBox:~/bureau/TP4/sig2dot25 cd ..  
nathan@nathan-VirtualBox:~/bureau/TP4$ pgp -o -options -with-colons -fixed-l  
st-mode -list-sets | ./sig2dot2/sig2dot.py > test.dot  
2020-10-19 20:58:21,210 sig2dot.exporter.dot.writer INFO Renderdate: 2020-10-19T  
19:58:21.209717+00:00 (1603137501)  
nathan@nathan-VirtualBox:~/bureau/TP4$ sigdot.exporter.dot.writer INFO Max_Ratio: 1.0  
nathan@nathan-VirtualBox:~/bureau/TP4$ ls  
CLEF_A_PUB      rep_CLEA      sig2dot2      SSIR_t.txt.asc      SSIR_t.txt.gpg  
CLEF_B_PUB      rep_CLEA.asc  SSIR_CLEB     SSIR_t.txt.asc.asc  SSIR_t.txt.dot  
hash_SSIR       Rev_A         SSIR_txt      SSIR_t.txt.asc.gpg  tuple.txt  
rep_CLEA        SSIR          SSIR_t.asc    SSIR_t.asc.asc      SSIR_t.asc.gpg  
nathan@nathan-VirtualBox:~/bureau/TP4$ dot -Tpng test.dot -o test.png
```

La dernière commande : `dot -Tpng test.dot -o test.png` permet de générer le graphe



```
nathan@nathan-VirtualBox: ~/Bureau/TP1$ gpg --list-sig
/home/nathan/.gnupg/pubring.kbx
-----
pub  rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
    FC406C1B396EA38C20738AB8DBE33B1479FDA258
uid  [ ultimate ] CLEF_A <nat.sanchodelarosa@gmail.com>
sig 3  DBE33B1479FDA258 2020-10-11  CLEF_A <nat.sanchodelarosa@gmail.com>
sub  rsa3072 2020-10-11 [E] [expire : 2022-10-11]
sig  DBE33B1479FDA258 2020-10-11  CLEF_A <nat.sanchodelarosa@gmail.com>

pub  rsa3072 2020-10-11 [SC] [expire : 2022-10-11]
    F3B06652F1E4406B03B8C3727566AEFA74502939
uid  [ ultimate ] CLEF_B <nat.sanchodelarosa@gmail.com>
sig 3  7566AEFA74502939 2020-10-11  CLEF_B <nat.sanchodelarosa@gmail.com>
sig  DBE33B1479FDA258 2020-10-13  CLEF_A <nat.sanchodelarosa@gmail.com>
sub  rsa3072 2020-10-11 [E] [expire : 2022-10-11]
sig  7566AEFA74502939 2020-10-11  CLEF_B <nat.sanchodelarosa@gmail.com>
```

On remarque donc qu'il y a un lien entre la clé B et la clé A (DBE33B1479FDA258).

EXERCICE 2

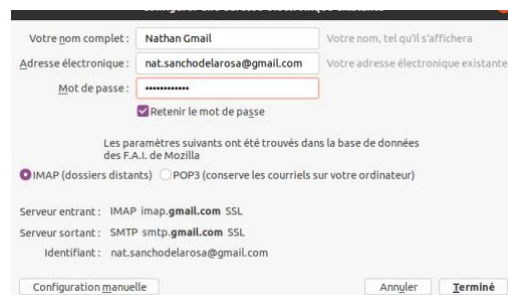
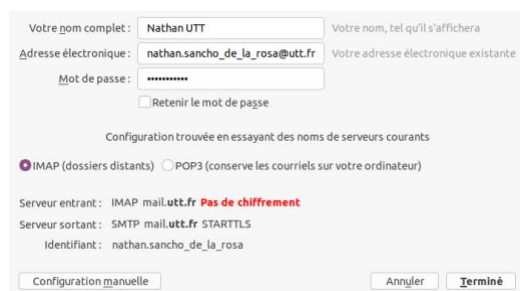
Installation de thunderbird

```
nathan@nathan-VirtualBox:~$ sudo apt install thunderbird
[sudo] Mot de passe de nathan :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
thunderbird est déjà la version la plus récente (1:68.10.0+build1-0ubuntu0.20.04.1).
thunderbird passé en « installé manuellement ».
0 mis à jour, 0 nouvellement installés, 0 à enlever et 207 non mis à jour.
```

Lancement de thunderbird

```
nathan@nathan-VirtualBox:~$ thunderbird &
[1] 2636
nathan@nathan-VirtualBox:~$ 1603138722759 addons.xpi WARN Not conv
erting unknown addon type undefined
```

Connexion avec une adresse email : un compte gmail et un compte utt



Aller dans la section module supplémentaire et installer enigmail

Filtrer les résultats

CATÉGORIE »

Tous les modules

FONCTIONNE AVEC »

Quel que soit Thunderbird

Tous les systèmes

Résultats de recherche pour « enigmail »

Trier par : Pertinence | Plus d'utilisateurs | Les mieux notés | Les plus récents | Plus ▼

**Enigmail**
Chiffrement des courriels et authentification OpenPGP pour Thunderbird.
★★★★★ (255) · 99 255 utilisateurs

[+ Ajouter à Thunderbird](#)

Configuration d'enigmail :

Il suffit de générer une paire de clés publique/privé pour chaque adresse mail. Ceci se fait automatiquement

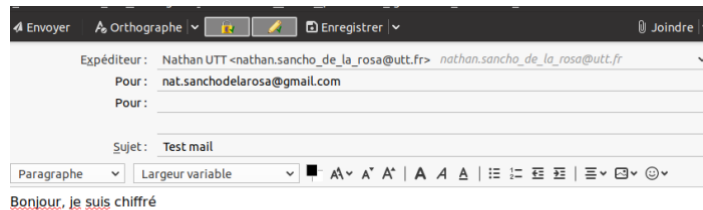
Voici les différentes clés

Nom	ID de clé
> CLEF_A <nat.sanchodelarosa@gmail....	DBE33B1479FDA258
> Nathan Gmail <nat.sanchodelarosa@...	B4DB3189100088E2
> Nathan UTT <nathan.sancho_de_la_r...	982D8DC7358F83D0
> CLEF_B <nat.sanchodelarosa@gmail.co...	7566AEFA74502939

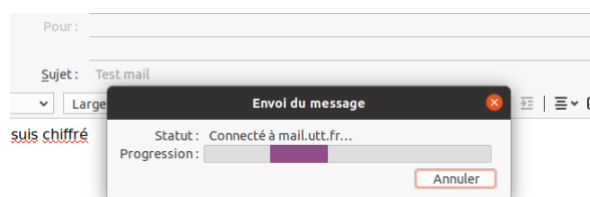
Ces clés ont été générés avec leur propre certificat et un mot de passe pour les protéger

Maintenant, je vais envoyer un mail de l'adresse utt vers l'adresse gmail

Ce mail sera chiffré et signé car j'ai coché les deux options



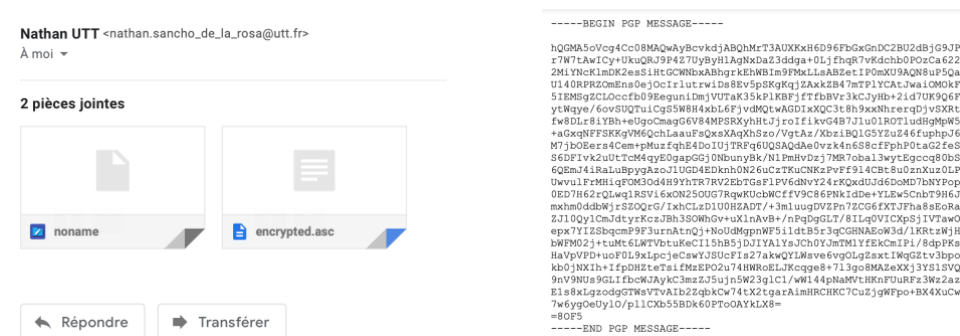
Envoie de l'email



Maintenant, on va voir la boîte de réception de la seconde messagerie sur son pc

On remarque qu'il y a deux fichiers qui sont la signature et le message chiffré

Voici le contenu du message



Pour déchiffrer le contenu de ce message, on peut se rendre sur la messagerie gmail depuis thunderbird

Ils vont nous demander le mot de passe de notre clé secrète afin de déchiffrer le message

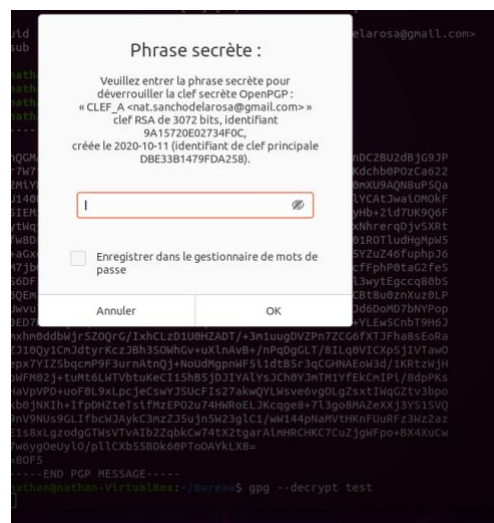


Où alors, on peut copier-coller le contenu du fichier asc

```
nathan@nathan-VirtualBox:~/Bureau$ touch test
nathan@nathan-VirtualBox:~/Bureau$ nano test
nathan@nathan-VirtualBox:~/Bureau$ cat test
-----BEGIN PGP MESSAGE-----

hQGMAS0Vcg4Cc08MAQWYBcvkdjABQhMrT3AUXKxH6D96FbGxGnDC2BU2dBjG9JP
r7W7tAwICy+UkuQRJ9P4Z7UyByH1AgNxDaZ3ddga+0LjfhqR7vKdchb0P0ZCa622
2M1YNcLMDK2es5iHtGCWNbXABhgrkEhWBI9FMxLLsABZetIP0mXU9AQN8uP5Qa
U140RPRZ0mEns0eJ0Cirlutrwids8Ev5pSKgKqjZAxkZB47mTPLYCatJwaiOM0kF
5IEMsgZCLOccfb09EeguniDmjVUTaK35kPLKBFjFTfbBVR3kCJyHb+2id7UK9Q6F
ytWqye/6ovSUQTuicgS5W8H4xbl6FjvdMQtwAGDIxXQC3t8h9xxNhrerqDjvSXRt
fw8DLr8iYBh+eUgoCmagG6V84MPSRXyhHtJjroIfikvG4B7J1u01ROTludHgMpw5
+aGxqNFFSKKgVM6QchLaauFsQxsXAqXhSzo/VgtAz/XbzibQLG5YzuZ46fuphpJ6
M7jb0Eers4Cem+pMuzfqhE4DoIUjTRFq6UQSAQdAe0vzk4n6S8cFFphP0taG2fes
S6DFivk2uUtCm4qyE0gapGGj0NbunyBk/N1PMHvDzj7MR7oba13wytEgcccQ80bS
6QEmJ4RaLubpygAzo31UG04EDknh0N26uCzTKuCNKzPvFf9L4CBt8u0znXuz0LP
UwvulFrMhiqFOM30d4H9YhTR7RV2EbTGSFLPV6dNvY24rKQxdUJd6DoMD7bNYPop
0ED7H62rQLwq1R5Vi6x0N250UG7RqwkucbWcFfV9C86PNkIdDe+YLEw5CnbT9H6J
mxhm0ddbWjrsZOQrG/IxhCLZD1U0HZAOT/+3m1uugDVZPn7ZCG6fXTJFha8sEoRa
ZJ10Qy1CmJdtyrkczJBh350WhGv+uXlnAvB+/nPqDgGLT/8ILq0VICXpSjIVTaw0
epx7YIZSbqcmP9F3urnAtnQj+NoUdMgpnWF5i1dtB5r3qCGHNAEow3d/1KrtzWjH
bWFM02j+tuMt6LWTVbtuKecI15hB5jDJIYALysJCh0YJmTM1YfEkCmIPi/8dpPKs
HaVpVPD+uof0L9xLpcJecswYJSUCFis27akwQYLvsve6vGOLgZsxtIWqGZtv3bpo
kb0jNXIh+IfpDHZet5IfMzEPO2u74HWRoELJKcqe8+7L3go8MAZEXXj3Y51SVQ
9nV9NUs9GLfIbCWJAYkC3mZJ5ujn5W23glC1/ww144pNamVTHKnFUuRFz3Wz2az
E1s8xLgzodgGTW5VTvAIbZ2qbkCw74tX2tgarAimHRCHKC7CuZjgWfpo+BX4XuCW
7w6ygoeUylo/pllCxb55Bdk60PT00AYKLX8=
=80F5
-----END PGP MESSAGE-----
nathan@nathan-VirtualBox:~/Bureau$
```

Et le décrypter avec gpg



```
nathan@nathan-VirtualBox:~/Bureau$ gpg --decrypt test
gpg: chiffré avec une clef ECDH de 256 bits, identifiant A085234D116AE944, créée
le 2020-10-19
« Nathan UTT <nathan.sancho_de_la_rosa@utt.fr> »
gpg: chiffré avec une clef RSA de 3072 bits, identifiant 9A15720E02734F0C, créée
le 2020-10-11
« CLEF_A <nat.sanchodelarosa@gmail.com> »
Content-Type: multipart/mixed; boundary="86UdJUb0edMPLQEV30y6cJaKIw4UPrNd";
protected-headers="v1"
From: Nathan UTT <nathan.sancho_de_la_rosa@utt.fr>
To: nat.sanchodelarosa@gmail.com
Message-ID: <fd59b203-109a-720b-699e-7252da7a64c2@utt.fr>
Subject: Test mail

--86UdJUb0edMPLQEV30y6cJaKIw4UPrNd
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: quoted-printable
Content-Language: en-US

Bonjour, je suis chiffr=C3=A9

--86UdJUb0edMPLQEV30y6cJaKIw4UPrNd--
gpg: Signature faite le lun. 19 oct. 2020 22:31:12 CET
gpg: avec la clef EDDSA 18305A6925A0C022EBCE9497982D8DC7358F83D0
gpg: Bonne signature de « Nathan UTT <nathan.sancho_de_la_rosa@utt.fr> » [ultime]
nathan@nathan-VirtualBox:~/Bureau$
```

Ainsi, on me demande le mdp de la clé et une fois rentré je vois le message en clair (le caractère spéciale é a bug). La signature de utt est vérifiée et bonne.

Ces deux techniques fonctionnent car tant sur la machine que sur thunderbird les deux paires de clés sont présentes. Il y a un petit problème avec la clé attribuée à gmail car j'utilise cette adresse mail pour 3 paires de clés.

Pour résumer, chaque adresse mail possède une paire de clé privé/publique. Chaque compte possède les clés publiques de ses contacts. Ainsi, chaque mail peut-être signé, chiffrés avec la clé publique du contact ou les 2.