## GPG

### I) Configuration de GPG

Dans un premier temps, nous allons télécharger sur toutes nos machines le paquet gpg :

```
apt update && apt upgrade
apt install gpg
```

Sur notre routeur, nous allons générer une paire de clés publique et privée en utilisant l'outil GPG à l'aide de la commande suivante :

```
gpg --full-generate-key --expert
```

Nous créons une clé de certification. Pour cela, nous choisissons le type de clé RSA (8) et nous enlevons les options de signature et de chiffrement en sélectionnant successivement (S) puis (C) et enfin (Q) pour quitter. Nous lui choisissons une taille de 4096 bits puis une durée de validité de 1an. Nous pouvons maintenant générer la clé.

```
root@asie-rtr:~# gpg --full-generate-key --expert
gpg (GnuPG) 2.2.40; Copyright (C) 2022 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.

gpg: répertoire « /root/.gnupg » créé
gpg: le trousseau local « /root/.gnupg/pubring.kbx » a été créé
Sélectionnez le type de clef désiré :
   (1) RSA et RSA (par défaut)
   (2) DSA et Elgamal
   (3) DSA (signature seule)
   (4) RSA (signature seule)
   (7) DSA (indiquez vous-même les capacités)
   (8) RSA (indiquez vous-même les capacités)
   (9) ECC et ECC
  (10) ECC (signature seule)
  (11) ECC (indiquez vous-même les capacités)
  (13) Clef existante
  (14) Existing key from card
Quel est votre choix ? 8

Actions possibles pour une clef RSA : Signer Certifier Chiffrer Authentifier
Actions actuellement permises : Signer Certifier Chiffrer

   (S) Inverser la capacité de signature
   (C) Inverser la capacité de chiffrement
   (A) Inverser la capacité d'authentification
   (Q) Terminé

Quel est votre choix ? SC
Choix incorrect.

Actions possibles pour une clef RSA : Signer Certifier Chiffrer Authentifier
Actions actuellement permises : Signer Certifier Chiffrer

   (S) Inverser la capacité de signature
   (C) Inverser la capacité de chiffrement
   (A) Inverser la capacité d'authentification
   (Q) Terminé

Quel est votre choix ? S

Actions possibles pour une clef RSA : Signer Certifier Chiffrer Authentifier
Actions actuellement permises : Certifier Chiffrer

   (S) Inverser la capacité de signature
   (C) Inverser la capacité de chiffrement
   (A) Inverser la capacité d'authentification
   (Q) Terminé

Quel est votre choix ? C

Actions possibles pour une clef RSA : Signer Certifier Chiffrer Authentifier
Actions actuellement permises : Certifier

   (S) Inverser la capacité de signature
   (C) Inverser la capacité de chiffrement
   (A) Inverser la capacité d'authentification
```

```
Actions possibles pour une clef RSA : Signer Certifier Chiffrer Authentifier
Actions actuellement permises : Certifier

   (S) Inverser la capacité de signature
   (C) Inverser la capacité de chiffrement
   (A) Inverser la capacité d'authentification
   (Q) Terminé

Quel est votre choix ? q
les clefs RSA peuvent faire une taille comprise entre 1024 et 4096 bits.
Quelle taille de clef désirez-vous ? (3072) 4096
La taille demandée est 4096 bits
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) 1y
La clef expire le mer. 11 sept. 2024 14:00:43 CEST
Est-ce correct ? (o/N) o

GnuPG doit construire une identité pour identifier la clef.

Nom réel : DSI Asie
Adresse électronique : macron@explosion.gsb.org
Commentaire :
Vous avez sélectionné cette identité :
    « DSI Asie <macron@explosion.gsb.org> »

Changer le (N)om, le (C)ommentaire, l'(A)dresse électronique
ou (O)ui/(Q)uitter ?
```

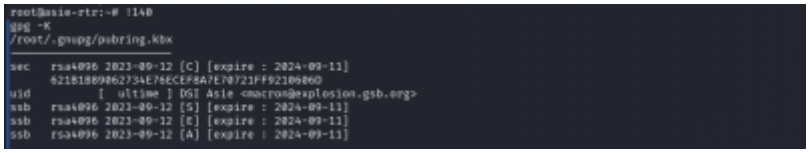
```
De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
gpg: /root/.gnupg/trustdb.gpg : base de confiance créée
gpg: répertoire « /root/.gnupg/openpgp-revocs.d » créé
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/6218188806273 4E76ECEF8A7E70721FF9210606D.rev'
les clefs publique et secrète ont été créées et signées.

pub   rsa4096 2023-09-12 [C] [expire : 2024-09-11]
      6218188806273 4E76ECEF8A7E70721FF9210606D
uid                      DSI Asie <macron@explosion.gsb.org>
```

Nous vérifions ensuite que la clé publique soit bien générée :

```
gpg -k
```


Nous vérifions également que la clé privée soit bien générée :

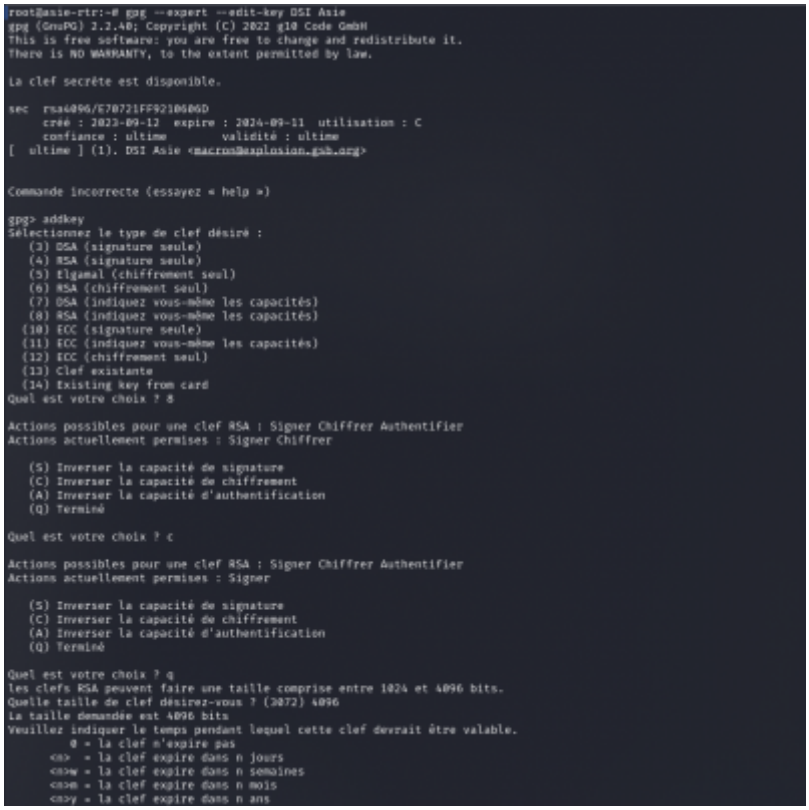
```
gpg -K
```




Nous allons maintenant créer une sous-clé de signature :

```
gpg --expert --edit-key DSI Asie
#pour cette commande il faut préciser --edit-key pour éditer la clé
puis indiquer le nom de la clé que nous souhaitons modifier.
addkey
```

Nous sélectionnons le type de clé RSA (8), puis nous enlevons l'option de chiffrement (C) et enfin (Q) pour quitter. Nous lui choisissons une taille de 4096 bits puis une durée de validité de 1an. Nous pouvons générer la sous-clé.

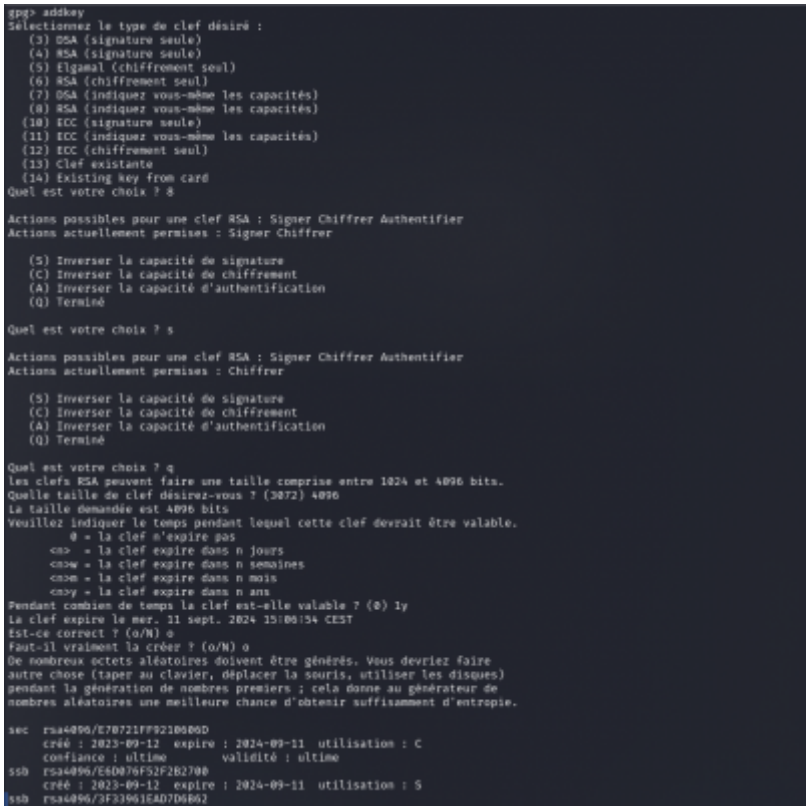




Nous créons une sous-clé de chiffrement :

```
addkey
```

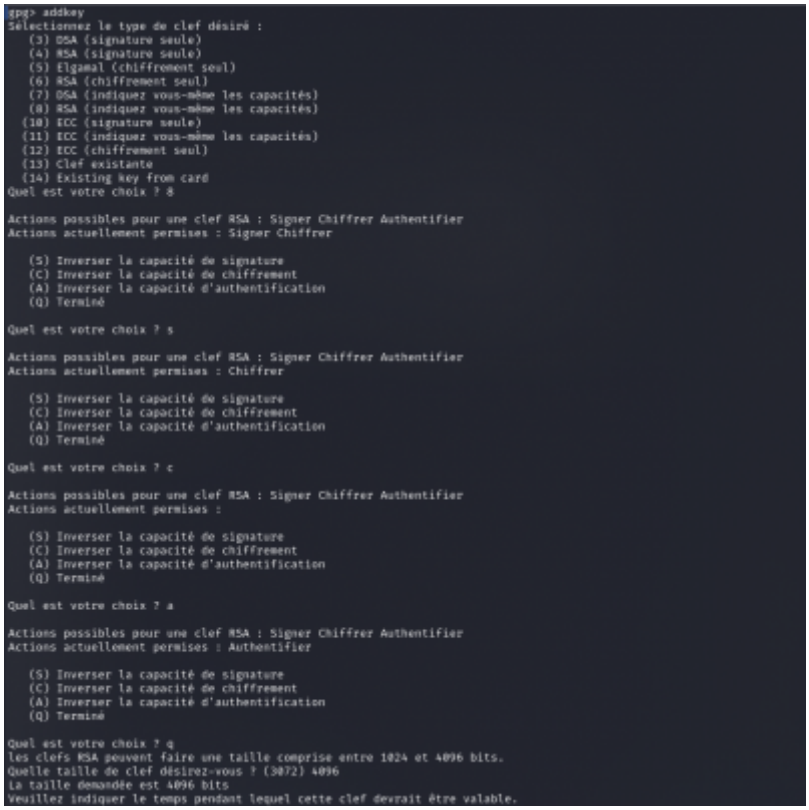
Nous sélectionnons le type de clé RSA (8), puis nous enlevons l'option de signature (S) et enfin (Q) pour quitter. Nous lui choisissons une taille de 4096 bits puis une durée de validité de 1an. Nous pouvons générer la sous-clé.




Nous créons une sous-clé d'authentification :

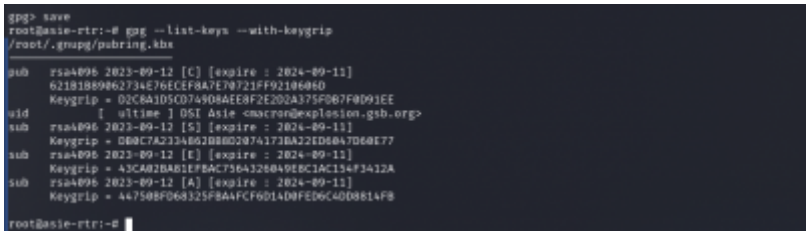
```
addkey
```

Nous sélectionnons le type de clé RSA (8), puis nous enlevons l'option de signature (S), de chiffrement (C), nous ajoutons l'option d'authentification (A) et enfin nous quittons le menu de configuration (Q). Nous lui choisissons une taille de 4096 bits puis une durée de validité de 1an. Nous pouvons générer la sous-clé.




Nous sauvegardons les changements effectués :

```
save
gpg --list-keys --with-keygrip
#l'option --list-keys permet de lister les clé présente
#l'option --with-keygrip permet d'afficher le grip de la clé
d'authentification.
```




Une fois nos clés créées, nous allons copier les clés publiques suivantes dans le fichier ~/.ssh/authorized_keys afin que les machines possédant ces clés publiques puissent se connecter au serveur :

- la clé publique du routeur
- la clé publique du routeur prof

Sur le serveur, nous allons importer la clé publique du routeur ainsi que sa clé privé pour autoriser la connexion du serveur vers le routeur et du routeur vers le serveur. Nous importons la clé publique dans le fichier ~/.ssh/authorized_keys.

Pour qu'une machine autorise une connexion SSH par clé GPG, nous devons importer la clé publique du routeur dans le fichier ~/.ssh/authorized_keys. Pour qu'une machine puisse se connecter à une autre machine autorisant la clé publique du routeur, il faut que cette dernière soit en possession de la clé privée du routeur.

Dans le fichier ~./ssh/sshd_config prohibit password

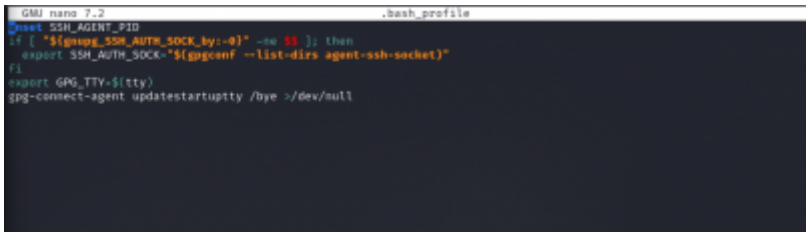
Etant donné que, par défaut, l'agent SSH ne reconnait pas les clés GPG, nous devons activer pour chaque machine (conteneurs et VM inclus) l'agent GPG dans le fichier ~/.gnupg/gpg-agent.conf afin que ce dernier puisse prendre en charge les clés.

```
enable-ssh-support >> $HOME/.gnupg/gpg-agent.conf
```

Nous modifions également notre fichier ~/.bash_profile pour y ajouter un script permettant d'échanger les socket ssh et gpg afin que l'on puisse utiliser les clé gpg pour une connexion ssh.

```
nano ~/.bash_profile
```

```
#script d'activation de l'agent gpg
unset SSH_AGENT_PID
if [ "${gnupg_SSH_AUTH_SOCK_by:-0}" -ne $$ ]; then
  export SSH_AUTH_SOCK="$(gpgconf --list-dirs agent-ssh-socket)"
fi
export GPG_TTY=$(tty)
gpg-connect-agent updatestartuptty /bye >/dev/null
```




Pour que les machines puissent prendre en compte tous les changements, nous nous déconnectons du compte utilisateur et nous nous reconnectons.

```
#afficher la liste des clés avec le grip de la clé d'authentification
gpg --list-keys --with-grip

echo 44750BFD68325FBA4FCF6D14D0FED6C4DD8814FB >> ~/.gnupg/sshcontrol
```

```
ssh-add -L
```


Nous pouvons ensuite modifier le TTL de notre passphrase afin de ne pas avoir à la rentrer chaque jours :


## II) Configuration de Sudo

Quelle est la différence entre les commandes su, su - et sudo ?

- La commande su permet de se connecter en tant que root en gardant les variables d'environnement de l'utilisateur précédent.
- La commande su - permet de se connecter en tant que root en utilisant les variables d'environnement de l'utilisateur root, ce qui permet d'utiliser des commandes apparaissant comme introuvables pour les autres utilisateurs.
- La commande sudo permet d'entrer une commande avec les privilèges administrateur si l'utilisateur fait partie des groupes pouvant utiliser la commande sudo.

Pour configurer la commande sudo, nous devons ajouter l'utilisateur std dans le groupe sudo.

```
usermod -a -G sudo std
```

Pour que les changements soient effectifs, nous devons nous déconnecter puis nous reconnecter à l'utilisateur std. Nous pouvons répéter cette manipulation sur chaque machine de notre réseau pour nous assurer que l'utilisateur std puisse utiliser la commande sudo.

Pour vérifier que l'utilisateur std soit dans le groupe sudo, nous utilisons la commande suivante :

```
cat /etc/group
```

Nous pouvons voir la liste des groupes et des utilisateurs qui leurs sont associés :