GNU - Linux // PGP (Pretty Good Privacy)

Le Concept (La Théorie)

Le PGP (Pretty Good Privacy) repose sur la **cryptographie asymétrique**. Imaginez un cadenas et une clé :

- **La Clé Publique (pgp-key.txt) :** C'est un cadenas ouvert. Vous en distribuez des milliers. N'importe qui peut prendre ce cadenas, mettre un message dans une boîte, et le fermer. Une fois fermé, personne (même pas celui qui l'a fermé) ne peut le rouvrir.
- **La Clé Privée (private.asc) :** C'est la *seule* clé capable d'ouvrir le cadenas. Elle ne doit **JAMAIS** quitter votre possession.

Dans notre contexte security.txt (RFC 9116), nous donnons le cadenas (Clé Publique) aux chercheurs de sécurité pour qu'ils y enferment les détails des failles découvertes.

Le choix de l’algorithme

(1) RSA and RSA (Le Choix "Standard Industriel")

C'est quoi ? L'algorithme **Rivest-Shamir-Adleman**. C'est le grand-père de la crypto. Il base sa sécurité sur la difficulté de factoriser de très grands nombres premiers.

- **Pourquoi ?** Pour la **COMPATIBILITÉ ABSOLUE**.
- Tout le monde supporte RSA. Du vieux serveur Linux oublié dans un placard depuis 1999 jusqu'au dernier MacBook.
- Dans un fichier security.txt, tu ne sais pas qui va t'envoyer une faille. Ça peut être un chercheur avec des outils dernier cri, ou un script automatique codé en Python 2 il y a dix ans. RSA, c'est le "passe-partout".
- **Le bémol :** C'est lourd. Une clé de 4096 bits, c'est gros. C'est lent à générer et à utiliser (comparé à ECC).
- **Verdict :** Le Tank. Indestructible, mais pas subtil.

(2) DSA and Elgamal (Le Musée des Horreurs)

- **C'est quoi ? Digital Signature Algorithm**. Un standard du gouvernement US des années 90.
- **Pourquoi l'éviter ?** C'est "Jurassic Park".
- Limité souvent à des clés de 1024 ou 2048 bits (trop faible aujourd'hui).
- Lent et complexe.
- **Note :** Choisir ça, c'est comme essayer de hacker la NASA avec un modem 56k. "Ça a sa place dans un musée."

(9) ECC (Elliptic Curve Cryptography) (Le Futur / Le Défaut)

- **C'est quoi ?** La cryptographie sur les **Courbes Elliptiques** (Ed25519 / Curve25519).
- **Pourquoi c'est mieux (en théorie) ?**
- **Efficacité :** Une clé ECC de 256 bits est aussi robuste qu'une clé RSA de 3072 bits !
- **Vitesse :** C'est ultra-rapide pour chiffrer/déchiffrer.
- **Pourquoi on ne l'a pas pris ?** Par pure paranoïa de compatibilité. Certains vieux clients mail ou vieux systèmes GPG (avant la version 2.1) galèrent avec l'ECC.
- **Verdict :** La nanotechnologie d'Iron Man. C'est l'avenir, c'est mieux, mais si ton interlocuteur a un vieux lecteur, il ne pourra pas lire ton message.

La Forge (Génération des Clés)

Nous utilisons l'algorithme RSA avec une taille de clé robuste (4096 bits).

```
user@lab-debian-72:~$ gpg --full-generate-key
gpg (GnuPG) 2.4.7; 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.

Please select what kind of key you want:
   (1) RSA and RSA
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
   (9) ECC (sign and encrypt) *default*
  (10) ECC (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 1y
Key expires at Tue 19 Jan 2027 09:18:36 AM CET
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: Admin unsec.fr
Email address: admin@unsec.fr
Comment: Security Key
You selected this USER-ID:
    "Admin unsec.fr (Security Key) <admin@unsec.fr>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: directory '/home/user/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as
'/home/user/.gnupg/openpgp-revocs.d/7AFBA13B7F6DCA19FC42C6FEACD4C156E86431B1.rev'
public and secret key created and signed.

pub   rsa4096 2026-01-19 [SC] [expires: 2027-01-19]
      7AFBA13B7F6DCA19FC42C6FEACD4C156E86431B1
uid                      Admin unsec.fr (Security Key) <admin@unsec.fr>
sub   rsa4096 2026-01-19 [E] [expires: 2027-01-19]
```

L'Exportation (La Distribution)

Une fois la paire générée, il faut extraire les deux parties pour les utiliser.

La Clé Publique (À diffuser)

C'est celle que nous mettrons sur le serveur web pour le security.txt.

```
user@lab-debian-72:~$ gpg --armor --export admin@unsec.fr > pgp-key.txt
```

--armor : Crée un format ASCII (texte) facile à copier-coller.

Destination : À placer dans /var/www/html/pgp-key.txt.

La Clé Privée (TOP SECRET)

C'est votre identité numérique.

```
user@lab-debian-72:~$ gpg --armor --export-secret-keys admin@unsec.fr > pgp-private-key.asc
```

> **ALERTE SÉCURITÉ :** Ce fichier pgp-private-key.asc ne doit **JAMAIS** rester sur le serveur.

Copiez-le sur une clé USB chiffrée ou dans votre gestionnaire de mots de passe (KeepassXC / Bitwarden).

Supprimez-le du serveur avec la commande shred -u pgp-private-key.asc (suppression sécurisée).

L'Inventaire (État des Lieux)

Avant d'agir, il faut savoir ce qu'on a en stock. Vous ne voulez pas chiffrer un message pour une clé expirée depuis 2012.

Lister les Clés Publiques (Le Trousseau)

C'est la liste des gens que vous connaissez (vos correspondants) et de vos propres cadenas publics.

```
user@lab-debian-72:~$ gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2027-01-20
/home/user/.gnupg/pubring.kbx
------------------------------
pub   rsa4096 2026-01-19 [SC] [expires: 2027-01-19]
      7AFBA13B7F6DCA19FC42C6FEACD4C156E86431B1
uid                      Admin unsec.fr (Security Key) <admin@unsec.fr>
sub   rsa4096 2026-01-19 [E] [expires: 2027-01-19]
```

(Ou la version courte pour les pressés : gpg -k)

Lister les Clés Privées (Vos Identités)

C'est la liste des identités que vous possédez réellement (celles pour lesquelles vous avez le droit de signature).

```
user@lab-debian-72:~$ gpg --list-secret-keys
/home/user/.gnupg/pubring.kbx
------------------------------
pub   rsa4096 2026-01-19 [SC] [expires: 2027-01-19]
      7AFBA13B7F6DCA19FC42C6FEACD4C156E86431B1
uid                      Admin unsec.fr (Security Key) <admin@unsec.fr>
sub   rsa4096 2026-01-19 [E] [expires: 2027-01-19]
```

(Ou la version courte : gpg -K)

Le Test Pratique (Chiffrer / Déchiffrer)

Voici la preuve par l'exemple que tout fonctionne. Nous allons jouer le rôle de "Pablo" (qui envoie un secret) et de "L'Admin" (qui le lit).

Le Chiffrement (Ce que fait le hacker)

Récupérer le Cadenas (Téléchargement)

L'utilisateur télécharge ton fichier public.

```
user@lab-debian-72:~$ wget https://unsec.fr/pgp-key.txt
```

Mettre le Cadenas dans sa poche (Importation)

C'est la commande magique qui dit à son GPG local : *"Hey, apprends qui est ce 'Admin UnSec'."*

```
user@lab-debian-72:~$ gpg --import pgp-key.txt
```

On crée un message secret

```
user@lab-debian-72:~$ echo "La matrice est partout, Neo." > secret.txt
```

2. On le chiffre pour "admin@unsec.fr"

```
user@lab-debian-72:~$ gpg --encrypt --armor -r admin@unsec.fr secret.txt
```

--encrypt : On chiffre

--armor : Pour avoir du texte affichable

-r : Recipient (Destinataire)

Le Déchiffrement (Ce que fait l'Admin)

Vous recevez ce bloc de texte. Vous seul pouvez le lire car vous avez la clé privée.

On déchiffre le fichier reçu

```
user@lab-debian-72:~$ gpg --decrypt secret.txt.asc
```

Le système vous demandera la "Passphrase" (le mot de passe défini lors de la création de la clé).

Maintenance (Renouvellement)

Dans 1 an (le 19/01/2027), la clé expirera. Les messages chiffrés avec cette clé seront rejetés.

Procédure de renouvellement

```
user@lab-debian-72:~$ gpg --edit-key admin@unsec.fr
gpg (GnuPG) 2.4.7; 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.

Secret key is available.

sec  rsa4096/ACD4C156E96431B1
     created: 2026-01-19  expires: 2027-01-19  usage: SC
     trust: ultimate      validity: ultimate
ssb  rsa4096/E1F8A478581E7BE8
     created: 2026-01-19  expires: 2027-01-19  usage: E
[ultimate] (1). Admin UnSec (Security Key) <admin@unsec.fr>

gpg> expire
Changing expiration time for the primary key.
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 1y
Key expires at Tue 19 Jan 2027 07:56:30 PM CET
Is this correct? (y/N) y

sec  rsa4096/ACD4C156E96431B1
     created: 2026-01-19  expires: 2027-01-19  usage: SC
     trust: ultimate      validity: ultimate
ssb  rsa4096/E1F8A478581E7BE8
     created: 2026-01-19  expires: 2027-01-19  usage: E
[ultimate] (1). Admin UnSec (Security Key) <admin@unsec.fr>

gpg>
gpg: signal Interrupt caught ... exiting
```

Enfin ont exporte les clé comme vu précedement

Celle que nous mettrons sur le serveur web pour le security.txt.

```
user@lab-debian-72:~$ gpg --armor --export admin@unsec.fr > pgp-key.txt
```

La clé privée à mettre dans votre gestionnaire de mots de passe (KeepassXC / Bitwarden).

```
user@lab-debian-72:~$ gpg --armor --export-secret-keys admin@unsec.fr > pgp-private-key.asc
```

Changer la Passphrase (Changer les codes du coffre)

C'est utile si tu penses que ton mot de passe actuel est aussi robuste qu'un papier cadeaux le soir de noel, ou si tu l'as tapé devant quelqu'un.

```
user@lab-debian-72:~$ gpg --edit-key admin@unsec.fr
gpg (GnuPG) 2.4.7; 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.

Secret key is available.

gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2027-01-19
sec  rsa4096/ACD4C156E86431B1
     created: 2026-01-19  expires: 2027-01-19  usage: SC
     trust: ultimate      validity: ultimate
ssb  rsa4096/E1F8A578581E7BE8
     created: 2026-01-19  expires: 2027-01-19  usage: E
[ultimate] (1). Admin UnSec (Security Key) <admin@unsec.fr>

gpg> passwd

gpg> save
Key not changed so no update needed.
```

La Procédure Interactive

Invite GPG	**Ta Commande**	**Action**
gpg>	passwd	"Je veux changer le mot de passe."
(Pop-up)	(Ancienne phrase)	Il te demande l'actuelle pour prouver que c'est toi.
(Pop-up)	(Nouvelle phrase)	Tape la nouvelle. Fais un effort, pas de "123456".
gpg>	save	**CRITIQUE.** Si tu quittes sans sauver, tu as travaillé pour rien.

> **Note :** C'est indolore. Ça ne change pas la clé publique. Tes correspondants ne verront rien. C'est juste toi qui changes la clé de ton propre coffre-fort local.

La Suppression Totale (Le Protocole "Winston Wolf")

Rappel Culturel : Pulp Fiction. "Je suis Winston Wolf, je résous les problèmes."

Parfois, il faut tout effacer. Une clé compromise, une erreur de génération, ou une envie de repartir à zéro. Attention, GPG est **paranoïaque** (à juste titre). Il t'interdira de supprimer la clé publique si tu as encore la clé privée. Ordre obligatoire.

Supprimer la Clé Privée (L'Identité)

C'est l'action destructrice. Une fois fait, si tu n'as pas de backup, c'est fini. Game Over.

```
user@lab-debian-72:~$ gpg --delete-secret-keys admin@unsec.fr
gpg (GnuPG) 2.4.7; 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.


sec  rsa4096/ACD4C156E86431B1 2026-01-19 Admin UnSec (Security Key) <admin@unsec.fr>

Delete this key from the keyring? (y/N) y
This is a secret key! - really delete? (y/N) y
```

GPG va te demander confirmation plusieurs fois (parfois 3 fois !). Il va te dire "Delete this key from the keyring?". Dis **Yes**.

Supprimer la Clé Publique (Le Cadenas)

Une fois l'identité supprimée, tu peux jeter le cadenas qui ne sert plus à rien.

```
user@lab-debian-72:~$ gpg --delete-keys admin@unsec.fr
gpg (GnuPG) 2.4.7; 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.


pub  rsa4096/ACD4C156E86431B1 2026-01-19 Admin UnSec (Security Key) <admin@unsec.fr>

Delete this key from the keyring? (y/N) y
```