Objet note : **Installation et mise en œuvre de Gnu PGP (Avec Revision 2026)**
/media/mbariou/LIBDEV/ProjLidata2026/Travaux-T1/Notes/Installe_Exploite_gpg.odt

PgP (Pretty Good Privacy) **GnuPG** / **GPG** (Gnu Privacy Guard)

Outre les outils PgP dont l'installation est décrite ci-après, il est recommandé d'installer d'autres outils qui seront précisé ultérieurement

Série de didacticiels en ligne en 6 parties (Si vous voulez en savoir plus)

La série de tutoriels est divisée en 6 parties :

- Partie 1 : Générez votre paire de clés publique/privée => https://fr.linux-terminal.com/?p=3196
- Partie 2 : Gestion des clés publiques => https://fr.linux-terminal.com/?p=3195
- Partie 3 : Chiffrer et déchiffrer le message => https://fr.linux-terminal.com/?p=3194
- Partie 4 : Comment chiffrer et déchiffrer les e-mails dans le client de messagerie Thunderbird => https://fr.linux-terminal.com/?p=3193
- Partie 5 : Signature numérique https://fr.linux-terminal.com/?p=3802
- Partie 6 : Comment vérifier la signature PGP des logiciels téléchargés sous Linux https://fr.linux-terminal.com/?p=3218

Ce qui suit synthétise le tutoriel.

Gnu GPG Clés publiques et privées

GPG est basé sur une paire de clé :

- une **clé privée** que vous gardez de par devers vous,
- une **clé publique** que vous mettez à disposition de vos interlocuteurs lors de signature de validation ou d'encryptage de message ou de document.
- Une troisième notion l'**empreinte** de votre paire de clés sera introduite plus loin.
- Une quatrième notion présentée plus loin est relative à **l'identité** de la clé.

Signature de documents et cryptage dans libre office

Un document peut faire l'objet d'une signature et/ou d'un cryptage :

- une **signature** valide et fige le contenu d'un document à un instant donné. Toute évolution du document exige une nouvelle signature afin de le figer avec son nouveau contenu. Tout lecteur du document, afin de contrôler sa validité, doit disposer de la clé publique comme de l'empreinte du couple clés publique/privée.
- Un **contenu peut-être chiffré** (non lisible par un humain) de deux manières :
  - par un **mot de passe** que tout lecteur devra avoir pour lire le document,
  - par une commande en ligne GnuGpg et l'envoi à tout lecteur de la clé publique et de l'empreinte du couple clés publique/privée.

Installation et mise en œuvre de Gnu PGP

Installation sur Ubuntu :

> sudo apt update => pour mise à jour
…. vérifier la liste pour éviter des mise à jour intempestive
> sudo apt upgrade (ou liste réduite)

> sudo apt install gnupg
Cette installation est à effectuer chez vous comme chez vos correspondants

Avant tout encryptage, il faut généré la paire de clé

```
> gpg --full-generate-key
```

Suivez le message à l'écran pour choisir votre type de clé, sa taille et sa date d'expiration. Vous devrez fournir, votre nom, votre adresse de courriel et un commentaire éventuel pour indiquer la finalité de votre clé(optionel). Donner quand même un mot de passe assez puissant pour protéger le secret de votre clé privée. Pour lister vos clés existantes faire :

> gpg --list-secret-keys --keyid-format LONG

Ensuite il faut préparer votre clé publique par un export local, dans un fichier ASCII:

gpg --armor --export marcel.bariou@laposte.net > .gnupg/PosteNonoPro_cle_publique.asc

Cette empreinte **PosteNonoPro_cle_publique.asc** vous l'adressez à votre correspondant par courriel ou par message crypté

Celui-ci en assure l'import sur son poste de travail :

 # il conduit l'opération pour Importer votre clé publique

> gpg --import PosteNonoPro_cle_publique.asc *#*

 # Il peut vérifier Vérifier l'import

> gpg --list-keys marcel.bariou@laposte.net

Rappel du processus complet de signature

Côté expéditeur (vous)

1. **Signer le document** dans LibreOffice
2. **Exporter votre clé publique dans un fichier ascii** :

```
>   gpg --armor --export marcel.bariou@laposte.net > PosteNonoPro_cle_publique.asc
```

Obtenir votre empreinte :

```
>  gpg --fingerprint marcel.bariou@laposte.net
```

ceci me donne 4 lignes dans mon terminal :

pub rsa3072 2026-01-25 [SC] [expire : 2026-07-24]

8DD9 BD24 0B20 D6EE 91C1 5532 5CF0 CBDB 4649 E2EE **<= Ceci est l'empreinte**

uid [ultime] Marcel Bariou (Clés pour usage non professionnels) <marcel.bariou@laposte.net>

sub rsa3072 2026-01-25 [E] [expire : 2026-07-24]

L'empreinte est la deuxième ligne => **8DD9 BD24 0B20 D6EE 91C1 5532 5CF0 CBDB 4649 E2EE**

4. **Envoyer** :
    - Le document signé (par email ou autre)
    - La clé publique fichier ASC (en pièce jointe ou lien vers serveur de clés)
    - L'empreinte (par canal sécurisé séparé)

Côté destinataire (votre correspondant)

1. **Importer votre clé depuis la pièce attachée** :

bash

gpg --import PosteNonoPro_cle_publique.asc

```

2. **Vérifier l'empreinte** (avec celle communiquée séparément)

3. **Ouvrir le document** dans LibreOffice

4. **Vérifier la signature** : Fichier → Signatures numériques

### *Exemple de courriel type à envoyer*

```

Bonjour,

Je vous transmets le document [nom] signé numériquement avec ma clé GPG.

Pour vérifier la signature :

1. Importez ma clé publique (ci-jointe)

2. Vérifiez que l'empreinte correspond à : 1234 5678 9ABC DEF0...

(je vous confirmerai cette empreinte par téléphone/SMS)

3. Ouvrez le document dans LibreOffice

4. Menu Fichier → Signatures numériques pour vérifier

Commande d'import local du fichier ASCII:

gpg --import ma_cle_publique.asc

Cordialement,`Bonnes pratiques de sécurité`

1. **Double canal** : Empreinte par un canal différent du document/clé
2. **Serveur de confiance** : Utilisez des serveurs de clés reconnus (keys.openpgp.org, keyserver.ubuntu.com)
3. **Expiration** : Configurez une date d'expiration sur votre clé
4. **Révocation** : Conservez un certificat de révocation en lieu sûr

Outils complémentaires

Pour des manipulations plus élaborées il peut être recommandé d'installer deux outils de contrôle et de vérification supplémentaires

- **Kleopatra** :Interface IHM de gestionnaire de clés cryptographique. Il a la capacité de gérer les certificats X.509 et OpenPGP ainsi que le gestionnaire de carte à puce GpgSM (scdaemon) et de retrouver des clés depuis des serveurs LDAP.
- **Scdaemon :** GNU Privacy Guard - prise en charge des cartes à puce GnuPG est l'outil GNU pour sécuriser les communications et le stockage de données. Il peut être utilisé pour chiffrer des données et créer des signatures numériques. Il inclut un mécanisme perfectionné de gestion de clés et est compatible avec la proposition de standard OpenPGP pour Internet tel que décrit dans la RFC 4880.

Il peut être recommandé d'unifier l'installation avec APT, plutôt que de mixer APT et SNAP quoique l'ensemble semble fonctionnel.

Stockage des clés sur les serveurs de clés

La clé publique peut-être jointe à votre message ou vous pouvez la déposer sur un serveur de confiance . Nous allons examiner la possibilité de stocker des clés sur des serveurs de distribution de clé. Nous allons introduire une dernière notion, à savoir, l'**identité de la clé** . On l'obtient avec la commande suivante, le couple de clés étant créé on interroge votre dépôt avec l'adresse de courriel associée à votre création

```
> gpg --list-sigs marcel.bariou@brasnah.fr
pub   rsa3072 2026-01-27 [SC] [expire : 2026-07-26]
      C802011C8878D2A933C2C17E1B29AD15D946506D
uid         [  ultime ] Marcel Bariou (Clés professionnelles Brasnah) <marcel.bariou@brasnah.fr>
sig 3       1B29AD15D946506D 2026-01-27  [autosignature]
sub   rsa3072 2026-01-27 [E] [expire : 2026-07-26]
sig         1B29AD15D946506D 2026-01-27  [autosignature]
```

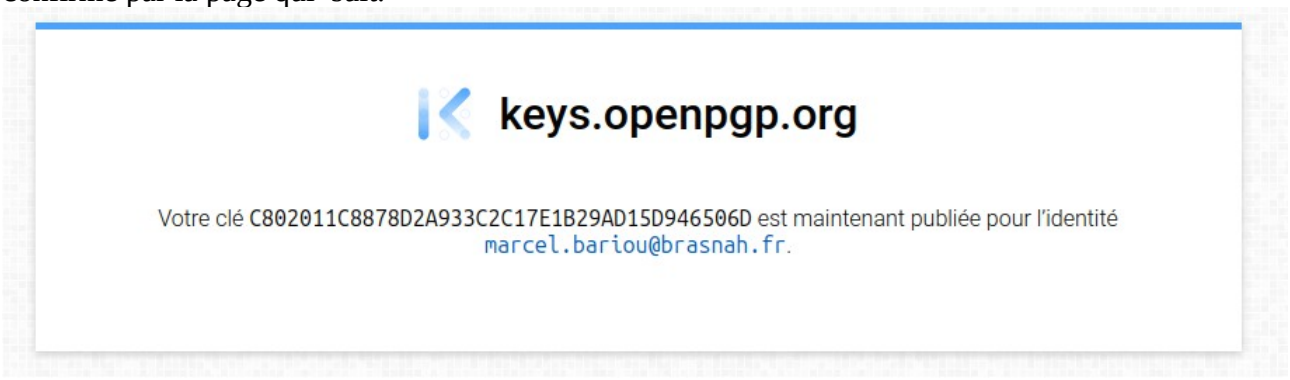
Ceci donne 6 lignes en retour dont :
- **C802011C8878D2A933C2C17E1B29AD15D946506D** est l'empreinte de votre paire de clés
- **1B29AD15D946506D** est l'identité de votre paire de clés.

GPG enverra votre clé publique au serveur de clés par défaut `hkps://keys.openpgp.org`.
L'envoi de votre clé sur le serveur par défaut qui est le précédent se fait pour la clé de marcel.bariou@brasnah.fr par la commande suivante :

```
> gpg --send-key 1B29AD15D946506D       (gpg –send-key id_cl é)
gpg: envoi de la clef 1B29AD15D946506D à hkps://keys.openpgp.org <= Message en retour
```

Ceci donne lieu à un premier message marcel.bariou@brasnah.fr rappelant l'empreinte et demandant la confirmation par un lien vers la page de confirmation

L'image ci-dessus affiche « **La confirmation est en attente** » suite au clic sur la même zone qui demandait de « **Confirmer votre dépôt** ». Pour des raisons de sécurité de dépôt, la confirmation peut prendre jusqu'à 15 mn. Ceci se traduit par l'envoi d'un dernier courriel « **Confirmer marcel.bariou@brasnah.fr pour votre clé sur openPgp.org** » avec un lien qui une fois activé permttra à vos correpondants d'accéder à votre clé publique sur le serveur de dépôt, ceci est confirmé par la page qui suit.



Ceci étant en place vos correspondant ont la possibilité de récupérer votre clé publique sue ce serveur keys.openpgp.org.

Exemple d'une seconde clé

> gpg --list-sigs marcel.bariou@laposte.net
pub rsa3072 2026-01-25 [SC] [expire : 2026-07-24]
8DD9BD240B20D6EE91C155325CF0CBDB4649E2EE
uid [ultime] Marcel Bariou (Clés non professionnelless) <marcel.bariou@laposte.net>
sig 3 5CF0CBDB4649E2EE 2026-01-25 [autosignature]
sub rsa3072 2026-01-25 [E] [expire : 2026-07-24]
sig 5CF0CBDB4649E2EE 2026-01-25 [autosignature]

GPG enverra votre clé publique au serveur de clés par défaut `hkps://keys.openpgp.org`.

L'envoi de votre clé sur le serveur par défaut qui est le précédent se fait pour la clé de marcel.bariou@laposte.net par la commande suivante :

> gpg --send-key 5CF0CBDB4649E2EE

gpg: envoi de la clef 5CF0CBDB4649E2EE à hkps://keys.openpgp.org <= Message en retour

Ces dépôts donnent lieu de la part du serveur à l'envoi d'un courriel à l'adresse courriel associé aux clés déposées ce courriel contient un lien qui vous permet de valider votre dépôt. Ceci étant fait, le serveur vous adresse une page dans lequel un lien vous demandant de confirmer le dépôt. Au bout de quelques minutes, pouvant aller jusqu'à 15 mn vous recevez un courriel doté d'un lien que vous devez activer pour valider définitivement votre dépôt. Cette validation définitive vous présentera une page à l'écran qui confirmera l'aboutissement du dépôt avec quelques recommandations.

Préparation/Envoi/Réception/ de document cryptés et/ou signés

Mettons cela en en œuvre, on exécute la commande suivante pour importer la clé publique que nous voulons, depuis le serveur openP.Notre identifiant est l'adresse courriel liée à la clé recherchée .

```
> gpg --search marcel.bariou@laposte.net

gpg: data source: https://keys.openpgp.org:443
(1)   Marcel Bariou (Clés pour usage non professionnels) <marcel.bariou@lap
        3072 bit RSA key 5CF0CBDB4649E2EE, créé : 2026-01-25

Keys 1-1 of 1 for "marcel.bariou@laposte.net".  Entrez le ou les nombres,
(S)uivant, ou (Q)uitter > 1 <=  (Rentrez le 1)
gpg: clef 5CF0CBDB4649E2EE : « Marcel Bariou (Clés pour usage non
professionnels) <marcel.bariou@laposte.net> » n'est pas modifiée
gpg: Quantité totale traitée : 1
gpg:              non modifiées : 1
```

Donc retour un enregistrement lié à l' adresse e-mail sur le serveur de clés, alors entrez le numéro `1` pour importer cette clé. Vérifiez ensuite localement l'empreinte digitale de cette clé par
`gpg --fingerprint marcel.bariou@laposte.net`

```
> gpg --fingerprint  marcel.bariou@laposte.net
pub   rsa3072 2026-01-25 [SC] [expire : 2026-07-24]
 8DD9 BD24 0B20 D6EE 91C1  5532 5CF0 CBDB 4649 E2EE
uid           [  ultime ] Marcel Bariou (Clés pour usage non
professionnels) <marcel.bariou@laposte.net>
sub   rsa3072 2026-01-25 [E] [expire : 2026-07-24]
```

Cette commande vous donne l'empreinte « `8DD9 BD24 0B20 D6EE 91C1  5532 5CF0 CBDB 4649 E2EE` » de la clé.

Exemple de signature sans cryptage

Il s'agit d'envoyer un document à quelqu'un sans confidentialité particulière, mais vous souhaitez que la personne qui le reçoit sache que le document vient de vous et n'a pas été modifié depuis votre signature.

Nous utilisons la commande suivante.

```
gpg --clearsign /chemin/au/fichier
```