Fiche Pratique N°37 : Vérifiez l'intégrité des fichiers avec GPG – Signatures et sommes de contrôle V1.1

7 nov, 2026 · @Gabriel

Objectif : *Apprendre à vérifier qu'un fichier téléchargé (logiciel, image ISO, document) est intact et qu'il vient bien de son auteur, grâce aux* ***sommes de contrôle*** *(SHA-256) et aux* ***signatures OpenPGP*** *(GnuPG).*

- ***Public visé :*** *Intermédiaire à avancé*
- ***Temps estimé :*** *20 à 40 minutes*
- ***Difficulté :*** *★★★☆☆ (Moyen)*
- ***Prérequis :*** *un ordinateur sous Windows, macOS ou Linux ; un fichier à vérifier et la somme ou la signature fournie par l'éditeur*
- ***Fiches associées :*** *N°27 (Tor Browser), N°32 (Tails), N°26 (VeraCrypt), N°13 (Linux)*

1. Pourquoi vérifier un fichier ? (Le problème)

1.1. Les risques

Situation	Conséquence
Téléchargement corrompu	Une coupure ou une erreur de transfert altère le fichier : installation ratée, clé USB qui ne démarre pas.
Fichier modifié en chemin	Un attaquant placé entre vous et le serveur peut substituer le fichier (rare avec HTTPS, mais possible avec un miroir ou un réseau compromis).
Miroir ou site piraté	Le site de téléchargement lui-même peut avoir été compromis.
Logiciel piégé	Un fichier modifié peut contenir un logiciel malveillant.

Le risque : installer un logiciel piégé en croyant installer l'original.

Le bénéfice : en deux commandes, vous savez si le fichier est intact et authentique.

1.2. Deux niveaux de vérification

Niveau	Ce que ça vérifie	Limite
Somme de contrôle (SHA-256)	Le fichier est-il **identique** à celui que l'éditeur a publié ?	Si un attaquant modifie à la fois le fichier et la somme affichée sur le même site, la vérification ne détecte rien.
Signature OpenPGP	Le fichier (ou la liste des sommes) a-t-il été **signé par l'auteur** ?	Il faut s'assurer qu'on a la bonne clé publique (vérifier son empreinte).

💡 **Règle d'or :** la somme de contrôle vérifie l'**intégrité** ; la signature vérifie l'**authenticité**. Pour un système d'exploitation ou un logiciel de sécurité, faites les deux.

gabriel.ananda@mailo.com
CC BY SA

Fiche Pratique N°37 : Vérifiez l'intégrité des fichiers avec GPG – Signatures et sommes de contrôle V1.1

2. Méthode A : vérifier une somme de contrôle

2.1. Qu'est-ce qu'une somme de contrôle ?

C'est une **empreinte** calculée à partir du contenu du fichier. La moindre modification change complètement l'empreinte.

Algorithme	Longueur (hexadécimal)	Sécurité	Usage
MD5	32 caractères	❌ Cassé : on sait fabriquer deux fichiers de même empreinte	Détection d'erreurs de transfert seulement
SHA-1	40 caractères	❌ Cassé : première collision pratique publiée en 2017 (SHAttered)	À éviter
SHA-256	64 caractères	✅ Standard actuel	**Recommandé**
SHA-512	128 caractères	✅ Robuste	Également recommandé

2.2. Où trouver la somme ?

- Dans un fichier publié à côté du téléchargement : `sha256sum.txt`, `SHA256SUMS`, `fichier.iso.sha256`.
- Directement sur la page de téléchargement.

Exemples :

- Linux Mint : fichiers `sha256sum.txt` et `sha256sum.txt.gpg` publiés avec chaque version.
- Tails : la page de téléchargement vérifie le fichier directement dans le navigateur.

2.3. Sous Linux

```
# Calculer l'empreinte SHA-256 d'un fichier
sha256sum fichier.iso

# Vérifier automatiquement avec la liste fournie (ignore les fichiers non téléchargés)
sha256sum --ignore-missing -c sha256sum.txt
# Résultat attendu : fichier.iso: OK
```

Si vous avez seulement la somme (copiée depuis la page) :

```
echo "SOMME_COPIÉE  fichier.iso" | sha256sum -c
```

(Deux espaces entre la somme et le nom du fichier.)

gabriel.ananda@mailo.com
CC BY SA

2.4. Sous Windows

PowerShell (intégré, recommandé) :

```
Get-FileHash -Algorithm SHA256 .\fichier.iso
```

Invite de commandes (intégré) :

```
certutil -hashfile fichier.iso SHA256
```

Avec une interface graphique : 7-Zip ajoute au menu contextuel (clic droit) une entrée **CRC SHA** › **SHA-256**.

Vous devez voir :

- une empreinte de 64 caractères, identique à celle publiée par l'éditeur (la casse des lettres ne compte pas).

2.5. Sous macOS

```
shasum -a 256 fichier.iso
```

3. Méthode B : vérifier une signature OpenPGP (authenticité)

3.1. Qu'est-ce que GnuPG ?

GnuPG (GPG) est le logiciel libre de référence pour le chiffrement et la signature au format OpenPGP. L'éditeur signe son fichier avec sa **clé privée** ; vous vérifiez la signature avec sa **clé publique**. Si la signature est bonne, le fichier vient bien du détenteur de la clé et n'a pas été modifié.

3.2. Principe

Étape	Ce qui se passe
1. L'éditeur crée une paire de clés	La clé privée reste secrète
2. L'éditeur signe le fichier (ou la liste des sommes)	Il publie un fichier `.sig`, `.asc` ou `.gpg`
3. Vous récupérez la clé publique	Et vous vérifiez son **empreinte** (40 caractères) auprès de plusieurs sources officielles
4. Vous vérifiez la signature	« Bonne signature » : le fichier est authentique

3.3. Installer GnuPG

Linux (Debian, Ubuntu, Mint) : GnuPG est généralement déjà installé ; sinon :

```
sudo apt install gnupg
```

Windows : installez **Gpg4win** (gpg4win.org), qui comprend GnuPG et l'interface graphique **Kleopatra**.

gabriel.ananda@mailo.com
CC BY SA

Fiche Pratique N°37 : Vérifiez l'intégrité des fichiers avec GPG – Signatures et sommes de contrôle V1.1

macOS : `brew install gnupg`, ou **GPG Suite** (gpgtools.org).

3.4. Vérifier une signature (pas à pas)

Étape 1 : Récupérez la clé publique de l'éditeur

```
# Depuis un serveur de clés, avec l'empreinte complète (40 caractères)
gpg --keyserver hkps://keys.openpgp.org --recv-keys EMPREINTE_COMPLÈTE

# Ou depuis un fichier fourni par l'éditeur
gpg --import cle_publique.asc
```

Utilisez toujours l'**empreinte complète**, jamais un identifiant court de 8 caractères : on peut fabriquer de fausses clés qui ont le même identifiant court.

Étape 2 : Vérifiez la signature

```
gpg --verify fichier.iso.asc fichier.iso
```

Résultat attendu (en français) :

```
gpg: Bonne signature de « Éditeur <email@exemple.org> »
gpg: Attention : cette clé n'est pas certifiée avec une signature de
confiance.
```

⚠️ **L'avertissement** signifie seulement que vous n'avez pas certifié vous-même cette clé. Ce qui compte : **« Bonne signature »**, et une empreinte de clé identique à celle publiée par l'éditeur. Si vous lisez **« MAUVAISE signature »**, n'utilisez pas le fichier.

3.5. Exemple concret : Tor Browser

Le Tor Project publie sa méthode de vérification. La clé de signature a pour empreinte `EF6E286DDA85EA2A4BA7DE684E2C6E8793298290`.

```
# 1. Récupérer la clé de Tor Browser (via le domaine torproject.org)
gpg --auto-key-locate nodefault,wkd --locate-keys torbrowser@torproject.org

# 2. L'exporter dans un trousseau dédié
gpg --output ./tor.keyring --export
0xEF6E286DDA85EA2A4BA7DE684E2C6E8793298290

# 3. Vérifier le fichier téléchargé avec sa signature .asc (adaptez les
noms)
gpgv --keyring ./tor.keyring tor-browser-linux-x86_64-VERSION.tar.xz.asc
tor-browser-linux-x86_64-VERSION.tar.xz
```

Vous devez voir :

- `gpgv: Good signature from "Tor Browser Developers (signing key) <torbrowser@torproject.org>"`.

gabriel.ananda@mailo.com
CC BY SA

Fiche Pratique N°37 : Vérifiez l'intégrité des fichiers avec GPG – Signatures et sommes de contrôle V1.1

Et Tails ? La page de téléchargement de Tails vérifie l'image directement dans le navigateur (fiche N°32) ; elle ne donne plus d'instructions OpenPGP.

3.6. Certifier une clé (facultatif)

Après avoir vérifié l'empreinte auprès de plusieurs sources (site officiel, documentation, autre canal), vous pouvez certifier la clé localement pour faire disparaître l'avertissement :

```
gpg --lsign-key EMPREINTE_COMPLÈTE
```

4. Cas pratique complet : image ISO de Linux Mint

Linux Mint publie un fichier `sha256sum.txt` et sa signature `sha256sum.txt.gpg`.

Étape 1 : Téléchargez les fichiers

Depuis la page de téléchargement de Linux Mint (ou un miroir), récupérez l'image ISO, `sha256sum.txt` et `sha256sum.txt.gpg`, dans le même dossier.

Étape 2 : Vérifiez la somme SHA-256

```
sha256sum --ignore-missing -c sha256sum.txt
```

Vous devez voir :

- `linuxmint-…-64bit.iso: OK`.

Étape 3 : Importez la clé de Linux Mint

```
gpg --keyserver hkp://keys.openpgp.org:80 --recv-key
27DEB15644C6B3CF3BD7D291300F846BA25BAE09
gpg --list-key --with-fingerprint A25BAE09
```

L'empreinte affichée doit être `27DE B156 44C6 B3CF 3BD7 D291 300F 846B A25B AE09` (« Linux Mint ISO Signing Key »).

Étape 4 : Vérifiez la signature de la liste des sommes

```
gpg --verify sha256sum.txt.gpg sha256sum.txt
```

Vous devez voir :

- « Bonne signature », faite avec la clé `27DEB15644C6B3CF3BD7D291300F846BA25BAE09`.

Les deux vérifications réussies prouvent que l'ISO est intacte **et** publiée par Linux Mint.

5. Tableau récapitulatif des commandes

Opération	Linux	Windows (PowerShell)	macOS
SHA-256	`sha256sum fichier`	`Get-FileHash -Algorithm SHA256 fichier`	`shasum -a 256 fichier`

gabriel.ananda@mailo.com
CC BY SA

Fiche Pratique N°37 : Vérifiez l'intégrité des fichiers avec GPG – Signatures et sommes de contrôle V1.1

Opération	Linux	Windows (PowerShell)	macOS
SHA-512	`sha512sum fichier`	`Get-FileHash -Algorithm SHA512 fichier`	`shasum -a 512 fichier`
Vérifier une liste de sommes	`sha256sum --ignore-missing -c sha256sum.txt`	Comparer visuellement	`shasum -a 256 -c sha256sum.txt`
Importer une clé	`gpg --import cle.asc`	`gpg --import cle.asc` (ou Kleopatra)	`gpg --import cle.asc`
Récupérer une clé sur un serveur	`gpg --keyserver hkps://keys.openpgp.org --recv-keys EMPREINTE`	Idem	Idem
Vérifier une signature	`gpg --verify fichier.asc fichier`	Idem (ou Kleopatra)	Idem

6. Dépannage

Symptôme	Cause possible	Action
« FAILED » ou empreinte différente	Téléchargement corrompu ou fichier modifié	Retéléchargez, de préférence depuis le site officiel
« No public key » / « Clé publique introuvable »	La clé n'a pas été importée	Importez-la (étape 1)
« MAUVAISE signature »	Fichier modifié, ou mauvais couple fichier/signature	N'utilisez pas le fichier ; retéléchargez les deux
« aucun fichier correctement formaté » avec `sha256sum -c`	Format de la liste différent	Comparez l'empreinte à la main
Serveur de clés injoignable	Pare-feu ou serveur indisponible	Essayez un autre serveur ou importez le fichier de clé de l'éditeur

7. À savoir avant de se lancer

Crainte fréquente	La réalité
« C'est trop technique. »	Commencez par les sommes SHA-256 ; la signature vient ensuite.

gabriel.ananda@mailo.com
CC BY SA

Fiche Pratique N°37 : Vérifiez l'intégrité des fichiers avec GPG – Signatures et sommes de contrôle V1.1

Crainte fréquente	La réalité
« Je télécharge depuis le site officiel, pourquoi vérifier ? »	Les sites et miroirs peuvent être compromis. C'est déjà arrivé : en 2016, le site de Linux Mint a été piraté et une image piégée a été proposée en téléchargement.
« L'avertissement de GPG m'inquiète. »	Il est normal. Vérifiez seulement « Bonne signature » et l'empreinte de la clé.
« Peu de logiciels publient des signatures. »	Les projets libres sérieux (distributions Linux, Tor, VeraCrypt, KeePassXC…) en publient.
« Sous Windows, c'est plus compliqué ? »	Gpg4win et Kleopatra offrent une interface graphique : clic droit › **Vérifier**.

8. Challenge 7 jours

Challenge : pendant 7 jours, vérifiez chaque logiciel ou image que vous téléchargez.

- **Jour 1 :** installez GnuPG (Gpg4win sous Windows, `gnupg` sous Linux et macOS).
- **Jour 2 :** téléchargez un logiciel libre qui publie une somme SHA-256 et vérifiez-la.
- **Jour 3 :** vérifiez la signature OpenPGP du même logiciel, si elle existe.
- **Jour 4 :** téléchargez l'image de Linux Mint et vérifiez sa somme.
- **Jour 5 :** importez la clé de Linux Mint et vérifiez la signature.
- **Jour 6 :** copiez un fichier, modifiez un caractère, et constatez que l'empreinte change totalement.
- **Jour 7 :** créez votre propre clé (`gpg --full-generate-key`) et signez un fichier (`gpg --detach-sign fichier`).

Vous allez constater :

- vérifier un fichier prend moins d'une minute ;
- vous savez distinguer un fichier intact d'un fichier altéré.

9. Alternatives et approfondissements

Si vous avez besoin de…	Essayez plutôt…
Une interface graphique pour GnuPG sous Windows	**Kleopatra** (inclus dans Gpg4win)
Une interface graphique sous macOS	**GPG Keychain** (GPG Suite)
Une interface graphique sous Linux	**Kleopatra** ou **GtkHash** (sommes de contrôle)
Vérifier une empreinte sur Android	Une application de calcul d'empreintes libre (F-Droid)

gabriel.ananda@mailo.com
CC BY SA

Fiche Pratique N°37 : Vérifiez l'intégrité des fichiers avec GPG – Signatures et sommes de contrôle V1.1

Si vous avez besoin de…	Essayez plutôt…
Chiffrer un fichier avec un mot de passe	`gpg -c fichier` (chiffrement symétrique)
Signer vos propres fichiers	`gpg --detach-sign fichier` (produit `fichier.sig`)

10. En résumé (ce que vous gagnez)

Action	Gagné	Renvoi
Vérifier la **somme SHA-256**	Fichier intact, sans corruption	Section 2
Vérifier la **signature OpenPGP**	Fichier publié par son auteur	Section 3
Contrôler l'**empreinte de la clé**	Pas de fausse clé	Section 3.4
Vérifier avant d'installer	Pas de logiciel piégé	Section 4
Créer votre propre clé	Signer vos fichiers et emails	Section 8

11. Conclusion

Si vous êtes…	Faites…
Débutant	Vérifiez au moins la somme SHA-256
Intermédiaire	Somme SHA-256 + signature OpenPGP pour tout système et tout logiciel de sécurité
Très exigeant	Vérifiez aussi l'empreinte de la clé auprès de plusieurs sources indépendantes
Journaliste ou militant	Vérifiez systématiquement Tails, Tor Browser et vos outils de chiffrement

À retenir absolument

- **SHA-256 + signature = intégrité + authenticité.**
- **Une somme seule ne prouve pas l'authenticité** si elle vient du même site que le fichier.
- **Vérifiez l'empreinte complète de la clé**, jamais un identifiant court.
- **« Bonne signature »** est le message qui compte ; l'avertissement de confiance est normal.

Test final :

1. Téléchargez l'image de Linux Mint, `sha256sum.txt` et `sha256sum.txt.gpg`.
2. Lancez `sha256sum --ignore-missing -c sha256sum.txt` : résultat **OK**.
3. Importez la clé de Linux Mint et vérifiez son empreinte.

gabriel.ananda@mailo.com
CC BY SA

4. Lancez `gpg --verify sha256sum.txt.gpg sha256sum.txt` : résultat **Bonne signature**.
5. Si vous obtenez « OK » et « Bonne signature » : le fichier est intact et authentique ✅

Références

- Linux Mint – Vérifier votre image ISO (sha256sum, clé 27DE B156… A25B AE09)
- Tor Project – Vérifier la signature de Tor Browser (clé EF6E 286D…, gpgv)
- Help Net Security – Première collision pratique sur SHA-1 (SHAttered, février 2017)
- The Register – Le site de Linux Mint piraté, images ISO piégées (février 2016)
- Tails – Installer depuis Linux (vérification du téléchargement dans la page)

gabriel.ananda@mailo.com
CC BY SA