

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

Objectif : Apprendre à vérifier l'intégrité et l'authenticité des fichiers téléchargés (logiciels, ISO, documents) en utilisant :

- **les sommes de contrôle (checksums)** : MD5, SHA1, SHA256
- **les signatures GPG** (GNU Privacy Guard) : vérification que le fichier provient bien de son auteur légitime.

Public visé : Intermédiaire à Avancé

Temps estimé : 20 à 40 minutes

Niveau de difficulté : ★★★☆☆ (Moyen)

Prérequis :

- Un ordinateur sous Windows, macOS ou Linux
- Avoir un fichier à vérifier (ex: ISO Linux, logiciel open-source)
- Avoir la somme de contrôle ou la signature GPG fournie par l'éditeur

1. Pourquoi vérifier l'intégrité d'un fichier ? (Le problème)


1.1 Les risques

Risque	Explication
Corruption pendant le téléchargement	Une coupure réseau ou une erreur de transfert peut altérer le fichier.
Attaque "man-in-the-middle"	Un pirate peut modifier le fichier pendant le téléchargement (ex: sur un WiFi public).

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

Risque	Explication
Site miroir compromis	Le site que vous utilisez pour télécharger peut avoir été piraté.
Logiciel malveillant	Un fichier modifié peut contenir un virus, un ransomware ou une porte dérobée.

1.2 Les deux niveaux de vérification

Niveau	Ce que ça vérifie	Sécurité
Checksum (MD5, SHA1, SHA256)	Le fichier est-il identique à l'original ?	Le fichier n'a pas été corrompu (mais un pirate peut modifier à la fois le fichier ET la checksum)
Signature GPG	Le fichier provient-il vraiment de l'auteur légitime ?	Le fichier est authentique (même si le site est piraté, la signature ne peut pas être forgée sans la clé privée)
 Règle d'or : La checksum vérifie l' intégrité (pas de corruption). La signature GPG vérifie l' authenticité (c'est bien l'auteur qui a signé). Pour une sécurité maximale, utilisez les deux .		

2. Méthode A : Vérifier une somme de contrôle (checksum)

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

Une somme de contrôle est une **empreinte numérique** (une chaîne de caractères) calculée à partir du contenu d'un fichier. La moindre modification du fichier change complètement l'empreinte.

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

Algorithme	Longueur	Sécurité	Usage
MD5	32 caractères	❌ Faible (attaques possibles)	À éviter (sauf vérification rapide non critique)
SHA1	40 caractères	⚠️ Faible (attaques théoriques)	À éviter
SHA256	64 caractères	✅ Fort (standard actuel)	Recommandé
SHA512	128 caractères	✅ Très fort	Pour les fichiers sensibles

2.2 Où trouver la checksum ?

Les éditeurs fournissent généralement un fichier contenant les sommes de contrôle :

- fichier.iso.sha256 (le fichier des sommes)
- fichier.iso.sha256sum (même principe)
- Ou directement sur la page de téléchargement (ex: SHA256: a3b2c1d4e5f6...)

Exemples :

- Linux Ubuntu : <https://ubuntu.com/download/desktop> → "Verify your download"
- Linux Mint : <https://linuxmint.com/verify.php>
- Tails : <https://tails.net/install/download/> (vérification intégrée)

2.3 Sous Linux (Ubuntu / Mint / Debian)

Vérification d'un seul fichier :

```
# Calculer la somme SHA256 d'un fichier
sha256sum fichier.iso
```

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

Comparer manuellement avec la valeur fournie

Vérification automatique avec fichier .sha256 :

Si vous avez un fichier .sha256 contenant la somme et le nom du fichier
sha256sum -c fichier.iso.sha256

Résultat attendu :
fichier.iso: OK

Vérification avec redirection :

Si vous avez la somme directement dans le presse-papier
echo "a3b2c1d4e5f6... fichier.iso" | sha256sum -c

Exemple concret (Tails) :

Téléchargez l'image Tails et le fichier de signature
wget https://tails.net/tails-amd64-6.0.img
wget https://tails.net/tails-amd64-6.0.img.sha256

Vérifiez
sha256sum -c tails-amd64-6.0.img.sha256

2.4 Sous Windows

Méthode 1 : PowerShell (intégré, recommandé)

powershell

Calculer la somme SHA256
Get-FileHash -Algorithm SHA256 .\fichier.iso

Résultat attendu :

# Algorithm Hash	Path
# -----	----
# SHA256 a3b2c1d4e5f6...	C:\fichier.iso

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

Méthode 2 : CertUtil (intégré)

cmd

certutil -hashfile fichier.iso SHA256

Méthode 3 : Logiciel tiers (interface graphique)

Logiciel	Open-source	Lien
----------	-------------	------

QuickSFV	✗ Non	https://quicksfv.org
RapidCRC	✗ Non	https://rapidcrc.sourceforge.io
GtkHash	✓ Oui	https://gtkhash.org

2.5 Sous macOS

Calculer la somme SHA256

shasum -a 256 fichier.iso

Ou

openssl dgst -sha256 fichier.iso

3. Méthode B : Vérifier une signature GPG (authenticité)

3.1 Qu'est-ce que GPG ?

GPG (GNU Privacy Guard) est un logiciel de chiffrement et de signature. Un éditeur signe son fichier avec sa **clé privée**. Vous vérifiez cette signature avec sa **clé publique**. Si la signature est valide, vous êtes certain que le fichier provient bien de l'éditeur (et qu'il n'a pas été modifié).

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

3.2 Principe de fonctionnement

Étape	Ce qui se passe
1. L'éditeur génère une clé publique et une clé privée	La clé privée est gardée secrète par l'éditeur
2. L'éditeur signe le fichier avec sa clé privée	Création d'un fichier .sig ou .asc
3. Vous téléchargez le fichier + la signature + la clé publique	Vous récupérez la clé publique de l'éditeur
4. Vous vérifiez la signature avec la clé publique	Si valide → le fichier est authentique

3.3 Installation de GPG

Sous Linux (Ubuntu / Debian / Mint) :

```
sudo apt install gnupg -y
```

Sous Windows :

- Téléchargez **Gpg4win** : <https://www.gpg4win.org>
- Installez (cochez "Kleopatra" et "GnuPG")

Sous macOS :

```
brew install gnupg
```

Ou téléchargez **GPG Suite** : <https://gpgtools.org>

3.4 Vérification d'une signature GPG (pas à pas)

Étape 1 : Récupérez la clé publique de l'éditeur

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

Importer une clé depuis un serveur de clés (avec son ID)

```
gpg --keyserver keyserver.ubuntu.com --recv-keys 12345678ABCDEFGH
```

Ou importer depuis un fichier .asc

```
gpg --import cle_publique.asc
```

Étape 2 : Vérifiez la signature

Vérifier que la signature correspond au fichier

```
gpg --verify fichier.iso.sig fichier.iso
```

Ou si la signature est en fichier .asc

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

Résultat attendu :

text

```
gpg: Signature faite le ... par "Éditeur <email@exemple.com>"
```

```
gpg: Signature correcte
```

```
gpg: WARNING: This key is not certified with a trusted signature!
```

⚠ **Le warning** : Il signifie que vous n'avez pas explicitement "certifié" (validé) cette clé. C'est normal pour une première utilisation. Le message important est **"Good signature"**.

3.5 Exemple concret : Vérifier une image Tails

Tails fournit une signature GPG pour chaque version.

1. Téléchargez l'image Tails et la signature

```
wget https://tails.net/tails-amd64-6.0.img
```

```
wget https://tails.net/tails-amd64-6.0.img.sig
```

2. Importez la clé publique de Tails

```
gpg --keyserver https://keys.openpgp.org --recv-keys FBAE108F5B0D7F54
```

3. Vérifiez la signature

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

```
gpg --verify tails-amd64-6.0.img.sig tails-amd64-6.0.img
```

3.6 Gérer la confiance des clés (faire disparaître le warning)

Pour supprimer le warning "not certified with a trusted signature", vous pouvez certifier la clé.

Signer la clé (après avoir vérifié son empreinte sur plusieurs sources)

```
gpg --sign-key 12345678ABCDEFGH
```

4. Cas pratique complet : Vérifier un logiciel open-source

4.1 Exemple avec Linux Mint ISO

Linux Mint fournit SHA256 et signature GPG.

Étape 1 : Téléchargez les fichiers

[wget https://mirrors.kernel.org/linuxmint/stable/21.3/linuxmint-21.3-cinnamon-64bit.iso](https://mirrors.kernel.org/linuxmint/stable/21.3/linuxmint-21.3-cinnamon-64bit.iso)

[wget https://mirrors.kernel.org/linuxmint/stable/21.3/sha256sum.txt](https://mirrors.kernel.org/linuxmint/stable/21.3/sha256sum.txt)

[wget https://mirrors.kernel.org/linuxmint/stable/21.3/sha256sum.txt.gpg](https://mirrors.kernel.org/linuxmint/stable/21.3/sha256sum.txt.gpg)

Étape 2 : Vérifiez la checksum SHA256

```
sha256sum -c sha256sum.txt 2>&1 | grep OK
```

Étape 3 : Importez la clé GPG de Linux Mint

```
gpg --keyserver keyserver.ubuntu.com --recv-keys "A25BAE09"
```

Étape 4 : Vérifiez la signature GPG

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

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

Si la signature est correcte, le fichier est authentique.

5. Tableau récapitulatif des commandes

Opération	Commande Linux	Commande Windows (PowerShell)
MD5	<code>md5sum fichier</code>	<code>Get-FileHash -Algorithm MD5 fichier</code>
SHA1	<code>sha1sum fichier</code>	<code>Get-FileHash -Algorithm SHA1 fichier</code>
SHA256	<code>sha256sum fichier</code>	<code>Get-FileHash -Algorithm SHA256 fichier</code>
Importer clé GPG	<code>gpg --import cle.asc</code>	<code>gpg --import cle.asc</code>
Recevoir clé depuis serveur	<code>gpg --recv-keys ID</code>	<code>gpg --recv-keys ID</code>
Vérifier signature	<code>gpg --verify fichier.sig fichier</code>	<code>gpg --verify fichier.sig fichier</code>

6. À savoir avant de se lancer

Crainte fréquente	La réalité
"C'est trop technique, je ne vais pas y arriver."	Commencez par les checksums (plus simples). La signature GPG vient ensuite.
"Pourquoi vérifier ? Je télécharge depuis le site officiel."	Le site officiel peut être piraté, ou vous pouvez être victime d'une attaque MITM. Snowden l'a montré.
"Le warning GPG m'inquiète."	C'est normal. Tant que vous voyez "Good signature", c'est valide.
"Je n'ai jamais vu de signature GPG"	Tous les logiciels open-source sérieux (Linux, Tails, Tor,

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

Crainte fréquente

La réalité

sur les logiciels que je télécharge." VeraCrypt, GIMP) fournissent des signatures.

"Et sur Windows, c'est plus compliqué ?"

Gpg4win rend l'utilisation aussi simple que sur Linux (interface graphique Kleopatra).

7. Challenge 7 jours

Challenge : Pendant 7 jours, vérifiez systématiquement l'intégrité de chaque fichier que vous téléchargez (logiciel, ISO, document important).

Jour 1 : Installez GPG sur votre système (Gpg4win sur Windows, gnupg sur Linux/macOS)

Jour 2 : Téléchargez un logiciel open-source (ex: GIMP, VLC, LibreOffice) et vérifiez sa checksum SHA256

Jour 3 : Trouvez la signature GPG du même logiciel et vérifiez-la

Jour 4 : Téléchargez une ISO Linux (Ubuntu, Mint, Tails) et vérifiez sa checksum

Jour 5 : Importez la clé GPG de l'éditeur et vérifiez la signature de l'ISO

Jour 6 : Comparez le résultat avec un fichier volontairement modifié (touchez un octet)

Jour 7 : Générez votre propre clé GPG (`gpg --full-generate-key`) et signez un fichier

À la fin : Vous saurez vérifier qu'un fichier est authentique et non corrompu.

8. Alternatives et approfondissements

Si vous avez besoin de...

Essayez plutôt...

Une interface graphique pour GPG sur

Kleopatra (inclus dans Gpg4win)

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

Si vous avez besoin de...

Essayez plutôt...

Windows

Une interface graphique pour GPG sur macOS

GPG Keychain (inclus dans GPG Suite)

Vérifier la checksum sur smartphone

Application "Hash Checker" (Android, open-source)

Vérifier un fichier sans ligne de commande

GtkHash (interface graphique multiplateforme)

Chiffrer vos fichiers avec GPG

`gpg -c fichier` (chiffrement symétrique)

Signer vos propres fichiers

`gpg --sign fichier` ou `gpg --detach-sign fichier`

9. En résumé – ce que vous gagnez

Action

Bénéfice

Vérifier la **checksum SHA256**

S'assurer que le fichier n'a pas été corrompu (intégrité)

Vérifier la **signature GPG**

S'assurer que le fichier provient bien de son auteur (authenticité)

Importer une clé publique

Faire confiance à un éditeur

Vérifier un fichier avant de l'exécuter

Éviter d'installer un logiciel malveillant

Générer sa propre clé GPG

Pouvoir signer ses propres fichiers ou emails

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

10. Conclusion

Si vous êtes...	Faites...
Utilisateur débutant	Vérifiez au moins la checksum SHA256 (simple et efficace)
Utilisateur intermédiaire	Vérifiez systématiquement checksum + signature GPG
Expert / paranoïaque	Vérifiez la signature GPG en plusieurs fois (plusieurs serveurs de clés)
Journaliste / militant	Utilisez toujours GPG pour les fichiers sensibles


Ce qu'il faut retenir absolument :

- **SHA256 + GPG** = intégrité + authenticité.
- La **checksum** seule vérifie que le fichier n'est pas corrompu, mais **ne prouve pas** qu'il est authentique.
- La **signature GPG** prouve l'authenticité, mais vous devez d'abord importer la bonne clé publique.
- **Tails** a une vérification intégrée très simple pour les débutants.
- **Gpg4win** (Windows) et **GPG Tools** (macOS) rendent la tâche accessible aux non-techniciens.

Test final :

1. ✓ Téléchargez un fichier (ex: ISO Ubuntu)
2. ✓ Téléchargez son fichier SHA256
3. ✓ Exécutez `sha256sum -c fichier.sha256`
4. ✓ Résultat : OK
5. ✓ Téléchargez la signature GPG (.sig ou .asc)
6. ✓ Importez la clé publique de l'éditeur
7. ✓ Exécutez `gpg --verify fichier.sig fichier`

Fiche Pratique N°37 : Vérifier l'intégrité des fichiers avec GPG (signatures, checksums) V1.0

8.  Résultat : Good signature

Si vous avez "OK" et "Good signature" : **le fichier est intègre et authentique** 

Ressources officielles :

- GnuPG : <https://gnupg.org>
- Gpg4win : <https://www.gpg4win.org>
- GPG Tools (macOS) : <https://gpgtools.org>
- Ubuntu vérification : <https://ubuntu.com/download/desktop/verify>
- Tails vérification : <https://tails.net/install/download/>