Introduction à PGP Command Line

A

Pour maîtriser rapidement les opérations de base de PGP Command Line.

SOMMAIRE

PGP Corporation propose une gamme complète de logiciels. Celui dont il a été question dans ce livre est PGP Desktop, mais un autre logiciel susceptible de vous intéresser est PGP Command Line, qui s'adresse aux entreprises plutôt qu'aux particuliers. Il est disponible pour Windows et divers systèmes Unix (notamment Mac OS X), et il est fourni avec un manuel complet de plus de 300 pages (en anglais).

Cette annexe présente les fonctions de base de PGP Command Line afin que vous maîtrisiez rapidement ses opérations de base. Pour des informations plus approfondies, nous vous invitons à consulter le manuel de l'application.

Le fonctionnement de PGP Command Line est identique sous Windows et sous les systèmes d'exploitation de type Unix, la seule différence étant l'aspect de l'invite de commande. Dans les exemples qui suivent, nous utiliserons un dièse (`#`), comme c'est le cas dans certains shells Unix, mais les utilisateurs ne disposant que de privilèges normaux peuvent très bien utiliser PGP Command Line.

> **À RETENIR** **Non interactivité**
>
> Contrairement à la plupart des applications fonctionnant depuis l'invite de commande, PGP Command Line est entièrement non-interactif une fois l'exécution lancée. Il est conçu pour les environnements automatisés où aucun utilisateur n'est présent, et par conséquent, ne demande aucune information supplémentaire. Ce type de comportement est bien entendu particulièrement pratique dans le cadre de scripts.

Configurer PGP Command Line

Toutes les informations concernant PGP Command Line, notamment la configuration et les clés, sont stockées dans un répertoire d'application. Sous Windows, ce répertoire est défini lors de l'installation. Sous Unix, il s'agit du répertoire `$HOME/.pgp`. Pour changer le fonctionnement de PGP Command Line, modifiez le fichier de configuration `PGPPrefs.xml`. Les paramètres par défaut conviennent à presque toutes les situations ; vous en trouverez une explication complète dans le manuel.

Le fichier de configuration doit contenir du XML valide. Heureusement, sa structure est très simple. En voici un exemple :

```
<key>CLpassphraseCacheTimeout</key> (1)
<integer>120</integer> (2)
```

Dans cet exemple, la variable `CLpassphraseCacheTimeout` (1), qui détermine pendant combien de temps la phrase secrète reste mémorisée, a une valeur de `120`. Pour modifier cette valeur, entrez un nombre à la place de `120` (2) sans toucher aux balises environnantes.

`pubring.pkr` et `secring.skr` sont deux autres fichiers importants de ce répertoire. Le premier stocke votre trousseau de clés publiques tandis que le second stocke votre trousseau de clés privées. Est-il nécessaire de revenir sur l'importance de la confidentialité de ce second fichier ?

CONSEIL Erreurs, sauvegardes, versions

Lors de la modification d'un fichier de configuration tel que celui-ci, il arrive couramment d'effacer par accident un chevron ouvrant ou fermant sans s'en rendre compte, et d'empêcher ainsi tout fonctionnement du logiciel. Par conséquent, gardez toujours une copie de votre dernière configuration fonctionnelle ! Il est beaucoup plus rapide de revenir à une configuration antérieure et de la modifier à nouveau que d'examiner un fichier pendant des heures en essayant de détecter la cause de l'erreur. Sous Unix, nous vous conseillons fortement d'utiliser le système d'administration de versions CVS (logiciel libre sous licence GPL), ou son successeur Subversion (logiciel libre sous licence Apache), pour la gestion de vos fichiers de configuration. Il est gratuit et, ayant été utilisé depuis des décennies, il est particulièrement fiable.

Vérification et licence

Une fois l'installation de PGP Command Line terminée, assurez-vous qu'il fonctionne en essayant les options `--version`, qui affiche le numéro de version de PGP Command Line dont vous disposez, et `--help`, qui affiche toutes les options disponibles. Si ces deux options fonctionnent, votre installation est correcte.

Entrez ensuite la licence de votre logiciel. Dans le cas contraire, les fonctions disponibles de PGP Command Line (PGPCL) seront très limitées. Vous devez disposer d'un accès Internet pour que le logiciel effectue une vérification à distance du numéro de licence, du nom, de l'adresse électronique et de l'organisation. Ces informations doivent être saisies exactement comme elles apparaissent dans la licence :

```
# pgp --license-authorize (1)
  ➥  --license-name "Michael Lucas" (2)
  ➥  --licenseorganization "Author (Press)" (3)
  ➥  --license-number "longue-chaîne-de-caractères" (4)
  ➥  --license-email "mwlucas@blackhelicopters.org" (5)
```

Nous demandons ici à PGPCL d'autoriser la licence (1) correspondant au nom (2), à l'organisation (3), au numéro de licence (4) et à l'adresse électronique indiqués (5).

Si toutes les informations ont été correctement saisies, PGPCL ajoutera les informations de licence à `PGPprefs.xml` après avoir contacté le serveur de PGP Corporation, et PGPCL est prêt à être utilisé.

Créer une paire de clés

PGPCL sait créer toutes sortes de paires de clés différentes en plus des clés OpenPGP standards pour la signature et le chiffrement. Le format pour la création d'une paire de clés OpenPGP est le suivant, où `UID` (1) est l'identifiant utilisateur standard de PGP, sous la forme d'un nom suivi d'une adresse électronique entre chevrons.

```
# pgp --gen-key "UID" (1)
  ➥  --key-type rsa (2)
  ➥  --encryption-bits 2048 (3)
  ➥  --passphrase "phrase-secrète" (4)
  ➥  --other-options
```

Pour ajouter un commentaire facultatif à l'identifiant utilisateur (voir chapitre 2), insérez-le entre parenthèses entre le nom et l'adresse électronique. PGP vous informera que ce format n'est pas standard, mais il est néanmoins couramment employé et fonctionne très bien. Ainsi, l'identifiant de l'auteur de ces lignes est le suivant :

```
Michael Warren Lucas Jr (Author, consultant, sysadmin)
<mwlucas@blackhelicopters.org>
```

Les commentaires servent à vous différencier de vos homonymes qui utilisent également OpenPGP.

PGPCL permet l'utilisation de plusieurs types de paires de clés ❷. Si vous avez des besoins particuliers en matière de cryptographie, nous vous invitons à consulter le manuel ; nous ne parlerons ici que des paires de clés OpenPGP.

Choix du type de clé

Les clés OpenPGP modernes utilisent le type RSA. Pour définir le type de clé, utilisez le paramètre `--key-type`.

Pour le nombre de bits de la paire de clés ❸ (voir chapitre 1), choisissez une valeur entre 1 024 et 4 096. Nous conseillons d'utiliser une valeur de 2 048, qui protégera vos documents pendant encore longtemps.

Choix de la phrase secrète

Vous devez indiquer une phrase secrète ❹ lors de la création de la clé. Vous pourrez en changer par la suite avec l'option `--change-passphrase`.

Choix d'une date d'expiration

Par défaut, les nouvelles clés n'ont pas de date d'expiration. Pour leur en donner, utilisez le paramètre facultatif `--expiration-date` et spécifiez la date au format AAAA-MM-JJ, par exemple 2008-12-31.

Voici un exemple de création de paire de clés :

```
# pgp --gen-key "Michael Warren Lucas Jr (Consultant, author,
  ➥             sysadmin)<mwlucas@blackhelicopters.org>"
  ➥  --key-type rsa
  ➥  --encryption-bits 2048
  ➥  --passphrase "Ceci n'est pas une bonne phrase secrète"
  ➥  --expiration-date 2008-12-31
```

Lorsque la clé est créée, PGP affiche son keyid.

Génération d'un certificat de révocation

Comme pour n'importe quelle clé OpenPGP, il est important de créer un certificat de révocation immédiatement après avoir créé la clé. Pour ce faire, utilisez l'option `--gen-revocation` :

```
# pgp --gen-revocation "UID"❶
  ➥  --passphrase "phrase-secrète"❷
  ➥  --force ❸
```

Le premier argument indique à PGPCL quelle clé doit être révoquée. Il suffit pour cela de spécifier une partie de l'identifiant utilisateur ❶ de manière à identifier sans ambiguïté une clé de votre trousseau. Indiquez ensuite votre phrase secrète ❷. Pour finir, le paramètre `--force` ((3)) évite l'affichage d'un message de confirmation.

Ainsi, pour créer un certificat de révocation pour la clé précédemment calculée, voici la commande à saisir :

```
# pgp --gen-revocation mwlucas@blackhelicopters.org
  ➥ --passphrase "Ceci n'est pas une bonne phrase secrète"
  ➥ --force
```

Exportation de la clé publique

La prochaine étape consiste à distribuer votre clé publique auprès de vos correspondants, soit par l'intermédiaire d'un serveur de clés, soit sous la forme d'un fichier au format texte.

Serveur de clés

Utilisez l'option `--keyserver-send` pour envoyer votre clé sur un serveur de clés, et l'option `--keyserver` pour choisir le serveur de clés :

```
# pgp --keyserver-send UID
  ➥  --keyserver protocole://serveur-de-clés
```

Comme lors de la génération d'un certificat de révocation, il suffit de spécifier une partie suffisante de l'identifiant utilisateur pour identifier sans ambiguïté la clé à envoyer.

Vous devez spécifier à la fois le nom du serveur de clés et le protocole employé pour y accéder. Les protocoles les plus courants sont LDAP (qui est utilisé par le serveur de clés de PGP Corporation) et HTTP (qui est employé par des serveurs de clés tels que subkeys.pgp.net). Indiquez le protocole avant le nom du serveur, à la manière d'une URL. Ainsi, pour envoyer notre clé PGP nouvellement créée au serveur de clés de PGP Corporation, keyserver.pgp.com, nous exécuterions ce qui suit :

```
# pgp --keyserver-send mwlucas@blackhelicopters.org
  ➥  --keyserver ldap://keyserver.pgp.com
```

Par défaut, PGPCL contacte le serveur de clés public de PGP Corporation. L'argument `--keyserver` n'est nécessaire que si vous souhaitez contacter un autre serveur de clés.

Fichier au format texte

Pour exporter au format texte votre clé publique depuis votre trousseau, utilisez la commande `--export` :

```
# pgp --export UID
```

Vous créez ainsi un fichier au format texte dont le nom est identique à votre identifiant utilisateur. Si vous avez créé un certificat de révocation et si vous ne l'avez pas renommé, PGPCL vous indiquera qu'il ne peut pas créer le fichier. Dans ce cas, renommez votre certificat de révocation et réexportez la clé.

Afficher des clés

Pour afficher toutes les clés de votre trousseau, utilisez l'option `--list-keys`. Pour afficher une clé particulière, spécifiez l'identifiant utilisateur correspondant, ou une partie de cet identifiant, après `--list-keys`.

Tableau A–1 Options de PGPCL pour l'affichage de clés

Option	Fonction
`--pgp-fingerprint`	Affiche l'empreinte de la clé spécifiée
`--list-key-details`	Affiche toutes les informations de la clé
`--list-sigs`	Affiche toutes les signatures de la clé
`--list-sig-details`	Affiche des informations détaillées sur les signatures de la clé
`--list-userids`	Affiche tous les identifiants utilisateur de la clé

Voici par exemple comment afficher toutes les clés dont l'identifiant utilisateur comprend la chaîne mwlucas :

```
# pgp --list-keys mwlucas
Alg   Type Size/Type Flags   Key ID     User ID
----- ---- --------- ------- ---------- -------
*RSA4 pair 2048/2048 [VI---] 0x7E02501C Michael Warren Lucas Jr
 ➥ (Consultant, author, sysadmin)
 ➥ <mwlucas@blackhelicopters.org>
1 key found
```

Ici, PGPCL n'a trouvé qu'une seule clé. La possibilité de filtrer les résultats s'avère très utile dès lors que le nombre de clés du trousseau commence à augmenter.

La liste des clés par défaut ne comporte que des informations sommaires, mais vous pouvez en afficher d'autres à l'aide des options du tableau A-1.

Vous pouvez utiliser ces options seules ou suivies d'un identifiant ou d'une partie d'identifiant. Ainsi, pour afficher toutes les signatures de notre clé, nous exécuterions ce qui suit :

```
# pgp --list-sigs mwlucas@blackhelicopters.org
```

Gestion des clés

Les opérations les plus couramment effectuées au quotidien sur les clés sont la recherche, la signature et la mise à jour.

Recherche de clés

Pour trouver la clé de quelqu'un sur un serveur de clés avec PGPCL, vous devez spécifier l'identifiant utilisateur de la clé recherchée ou une partie de cet identifiant, ainsi que le nom du serveur de clés où la recherche doit être effectuée.

La meilleure méthode consiste à rechercher l'adresse électronique du correspondant. Voici comment chercher sur le serveur subkeys.pgp.net toutes les clés comportant la chaîne `Michael Lucas` :

```
# pgp --keyserver-search "Michael Lucas" --keyserver http://subkeys.pgp.net
http://subkeys.pgp.net:keyserver search (2504:successful search)
Alg Type Size/Type Flags Key ID User ID
----- ---- --------- ------- ---------- -------
DSS pub 2048/1024 [-----] 0xE68C49BC Michael Warren Lucas Jr (Author,
consultant, sysadmin) <mwlucas@blackhelicopters.org> (1)
DSS pub 2048/1024 [-----] 0xAB6CA178 Michael Lucas <mike.lucas@teamlucas.com>
DSS pub 2048/1024 [-----] 0x4922B639 Michael P. Lucas <mlucas@jharris.com>
DSS pub 2048/1024 [-----] 0x4768326E B. Michael Lucas <n1tba@snet.net>
DSS pub 2048/1024 [-----] 0xAD08B0C7 David Michael Lucas <trajan97@yahoo.com>
DSS pub 2048/1024 [-----] 0xFB31770D David Michael Lucas
<Buckeye_D@yahoo.com>
6 keys found
```

Le serveur de clés indique qu'il a trouvé six clés correspondant à la chaîne `Michael Lucas` et affiche les résultats, mais une seule d'entre elles nous intéresse. La liste comprend l'identifiant de chaque clé (1), ce qui permet de spécifier par la suite une partie d'identifiant qui désigne sans ambiguïté la clé à importer.

> **À RETENIR keyserver.pgp.com**
>
> Une recherche sur keyserver.pgp.com n'aurait renvoyé qu'une seule clé, en raison du système de vérification de PGP via l'adresse électronique.

Importation de clés

Utilisez `--keyserver-recv` pour importer une clé publique depuis un serveur de clés et l'ajouter à votre trousseau public :

```
# pgp --keyserver-recv mwlucas@blackhelicopters.org
  ➥  --keyserver http://subkeys.pgp.net
```

PGP importe la clé et l'ajoute à votre trousseau, la rendant localement accessible de manière permanente.

Signature d'une clé

Une fois que vous avez examiné l'empreinte d'une clé et son identifiant, et que vous avez vérifié les pièces d'identité de son propriétaire (voir chapitre 5), vous pouvez choisir de signer cette clé.

PGPCL offre plusieurs formats de signature ; nous allons voir ici comment ajouter une signature « à l'ancienne », celle employée dans le réseau de confiance. Pour signer une clé, utilisez `--sign-key` comme suit :

```
# pgp --sign-key UID-de-la-clé-à-signer ❶
  ➥  --signer votre-UID ❷
  ➥  --sig-type exportable
  ➥  --passphrase phrase-secrète ❸
```

Vous devez indiquer à PGPCL quelle clé vous souhaitez signer ❶ en spécifiant une partie de son identifiant utilisateur ❷ qui permette de la désigner sans ambiguïté (vous ne pouvez signer qu'avec votre propre clé). Spécifiez pour finir votre phrase secrète ❸. PGP signe alors la clé et la stocke dans votre trousseau.

Mise à jour des clés sur un serveur

Après la signature de la clé, vous pouvez l'exporter dans un fichier et la restituer à son propriétaire ou la renvoyer sur un serveur de clés, si c'est là que vous l'avez obtenue (voir le chapitre 5 pour plus de détails).

Pour envoyer la clé mise à jour sur un serveur de clés, utilisez `--keyserver-update` :

```
# pgp --keyserver-update UID
```

Maintenant que vous savez gérer vos clés, intéressons-nous aux fonctions les plus importantes de PGPCL : le chiffrement et le déchiffrement.

Chiffrement et déchiffrement

Pour chiffrer un fichier avec PGPCL, utilisez `--encrypt` et `--recipient`.

```
# pgp --encrypt nom-de-fichier --recipient UID
```

Ainsi, pour chiffrer le fichier `ComptesBancaires.ods` que vous avez prévu d'envoyer à M. Lucas, entrez ce qui suit :

```
# pgp --encrypt ComptesBancaires.ods
  ➥  --recipient mwlucas@blackhelicopters.org
```

RAPPEL Fichiers ASCII

Par défaut, les fichiers chiffrés sont de type binaire. Pour obtenir des fichiers ASCII, ajoutez l'argument `--armor`.

Pour déchiffrer un fichier, utilisez `--decrypt` et `--passphrase`.

```
# pgp --decrypt nom-de-fichier --passphrase phrase-secrète
```

Pour déchiffrer le fichier mentionné plus haut, par exemple, nous utiliserions :

```
# pgp --decrypt ComptesBancaires.ods
  ➥  --passphrase "Ceci n'est pas une bonne phrase secrète"
```

Nous obtiendrions alors le fichier non chiffré `ComptesBancaires.ods`, qu'il ne resterait plus qu'à ouvrir dans OpenOffice.

Signature et vérification de fichiers

Pour signer et vérifier des fichiers, utilisez respectivement `--sign` et `--verify`. Les fichiers signés comportent l'extension `.pgp`, ou `.asc` s'ils sont au format ASCII.

Résumé des commandes GnuPG

B

Récapitulation des diverses commandes et options de GnuPG, cette annexe servira de référence.

SOMMAIRE

Tout au long de ce livre, nous avons évoqué diverses commandes et options de GnuPG. Elles sont toutes récapitulées dans cette annexe, qui vous servira de référence. Il est important de comprendre les tenants et les aboutissants de chacune de ces fonctions : par conséquent, ne sautez pas le reste de ce livre pour ne vous fier qu'à cette annexe.

Configurer GnuPG

AVANCÉ Options avancées de GnuPG

GnuPG est un logiciel puissant proposant de nombreuses options que nous n'avons pas évoquées. Ces options ne concernent pour la plupart que des situations très particulières, ou nécessitent une connaissance approfondie d'OpenPGP pour être correctement mises en œuvre.

Les informations de configuration et les trousseaux de clés de GnuPG sont stockés dans le même répertoire. Sous Windows, le répertoire est défini lors de l'installation. Sous les systèmes Unix, il s'agit de `$HOME/.gnupg` par défaut.

Le répertoire contient trois fichiers importants pour la plupart des utilisateurs : `gpg.conf`, `pubring.gpg` et `secring.gpg`, qui sont respectivement le fichier de configuration de GnuPG, le trousseau de clés publiques et le trousseau de clés privées.

Format et type de sortie

Lors de l'utilisation de clés GnuPG, il est important de spécifier le format et le type de sortie voulus. L'argument `-a` (ou `--armor`) indique à GnuPG que la sortie doit être effectuée dans un format « lisible » (ASCII) à la place du format binaire employé par défaut. L'argument `--output`, quant à lui, indique à GnuPG de créer un fichier plutôt que d'afficher le résultat à l'écran.

Créer une paire de clés, révoquer et exporter

Pour créer une nouvelle paire de clés, utilisez l'option interactive `--gen-key`. GnuPG vous fait alors parcourir une à une les étapes de création de la clé (ce sujet a été traité en détail au chapitre 4).

Révocation d'une clé

Pour générer un certificat de révocation pour votre paire de clés, utilisez l'option `--gen-revoke` en spécifiant l'identifiant utilisateur (UID) de la clé à révoquer.

Vous pouvez également choisir le format ASCII (armor) et demander à ce que le résultat soit un fichier, et non simplement affiché à l'écran.

```
# gpg  -a (1)
  ➥    --output mwlucas@blackhelicopters.org.asc.revoke (2)
  ➥    --gen-revoke (3) mwlucas@blackhelicopters.org (4)
```

Ici, le fichier généré (2) est `mwlucas@blackhelicopters.org.asc.revoke`, au format ASCII (1). Le fichier résultant est un certificat de révocation (3) pour la clé dont l'identifiant contient la chaîne `mwlucas@blackhelicopters.org` (4).

Exportation d'une clé

Pour exporter une clé dans un fichier texte, utilisez l'option `--export`. Dans la mesure où il s'agit d'un fichier texte, utilisez de préférence `--armor`.

```
# gpg --output pubkey.mwlucas@blackhelicopters.org.gpg.asc (1)
  ➥   --armor (2)
  ➥   --export (3) mwlucas@blackhelicopters.org (4)
```

Ici, nous créons un fichier nommé `pubkey.mwlucas@blackhelicopters.org.gpg.asc` (1) au format ASCII (2). Ce fichier correspond à l'exportation (3) de la clé dont l'identifiant comprend la chaîne `mwlucas@blackhelicopters.org` (4).

Envoi d'une clé sur un serveur de clés

Pour envoyer une clé sur un serveur de clés, utilisez l'option `--send-keys`. L'option `--keyserver` spécifie le serveur où la clé doit être envoyée. Si vous ne spécifiez aucun serveur, GnuPG utilisera le serveur par défaut spécifié dans `gpg.conf`.

```
# gpg --send-keys (1) mwlucas@blackhelicopters.org (2)
  ➥  --keyserver (3) subkeys.pgp.net (4)
```

Ici, la clé dont l'identifiant contient `mwlucas@blackhelicopters.org` (2) est envoyée (1) au serveur de clés (3) `subkeys.pgp.net` (4).

Gestion des trousseaux de clés

Lorsque vous créez une clé, vous créez en même temps un trousseau pour cette clé (et d'autres). Vous devrez de temps à autre ajouter et ôter des clés à ce trousseau.

Affichage des clés

GnuPG vous permet d'afficher vos clés et leurs caractéristiques :

- Avec --list-keys, vous affichez toutes les clés publiques de votre trousseau. Vous pouvez aussi spécifier une partie d'un identifiant utilisateur pour n'afficher que les clés correspondantes.
- Pour afficher les clés secrètes de votre trousseau, utilisez l'option `--list-secret-keys`.
- Pour afficher l'empreinte d'une clé, utilisez `--fingerprint` en spécifiant l'identifiant de la clé ou une partie de celui-ci.

Ajouter et ôter des clés

GnuPG vous permet d'effectuer directement depuis l'invite de commande toutes les opérations en rapport avec les serveurs de clés.

- Pour extraire une clé d'un serveur de clés, utilisez l'option `--recv-keys` et spécifiez le keyid de la clé à télécharger.

```
# gpg --recv-keys E68C49BC
```

- Pour importer une clé publique depuis un fichier, utilisez l'option --import et indiquez le nom du fichier. Aucune autre option n'est requise. Pour supprimer une clé de votre trousseau, utilisez l'option `--delete-key` et spécifiez l'identifiant ou le keyid de la clé à supprimer.

Signatures de clés

La validation, l'ajout et la mise à jour de signatures sont des éléments importants de GnuPG.

- Pour afficher les signatures d'une clé, utilisez l'option `--list-sigs` suivie du keyid ou de l'identifiant.
- Pour signer une clé, utilisez l'option `--sign-key` suivie du keyid ou de l'identifiant :

```
# gpg --sign-key mwlucas@blackhelicopters.org
```

- Pour exporter une clé nouvellement signée dans un fichier texte, utilisez --export, ainsi que `--output` pour créer le fichier et `--armor` pour spécifier le format ASCII.

```
# gpg --output gedonner.asc
  ➥ --armor
  ➥ --export mwlucas@blackhelicopters.org
```

- Pour envoyer la clé publique signée sur un serveur de clés, utilisez `--send-keys` en spécifiant le keyid ou l'identifiant de la clé signée.
- Pour mettre à jour toutes les clés de votre trousseau depuis un serveur de clés, utilisez `--refresh-keys`.
- Après avoir signé plusieurs clés, utilisez l'option `--update-trustdb` pour mettre en place votre propre réseau de confiance.

Chiffrement et déchiffrement

Utilisez l'option `--encrypt` pour chiffrer un fichier dont vous spécifiez le nom. GnuPG vous demandera l'identifiant des clés publiques que vous souhaitez employer pour le chiffrement, puis chiffrera le fichier de telle manière qu'il ne puisse plus être lu qu'avec la clé privée correspondante.

Pour déchiffrer un fichier, utilisez --decrypt. GnuPG vous demandera votre phrase secrète, puis affichera le message déchiffré à l'écran.

Signer un fichier

Pour signer un fichier, utilisez l'option `--sign` et spécifiez le nom du fichier. Pour vérifier une signature numérique, utilisez `--verify`. GnuPG vous indiquera si la signature est valide ou non.

Formats de sortie

Par défaut, les fichiers chiffrés et les clés sont générés dans un format binaire, et les noms de fichier sont basés sur les noms des fichiers d'origine. Vous pouvez spécifier un autre type de sortie avec les options `--armor` et `--output`.

- Avec l'option `--armor`, la sortie est au format ASCII, plus facile à lire pour les humains.
- L'option `--output` vous permet de choisir le nom du fichier où sera stocké le résultat de la commande.
- Par défaut, GnuPG crée un nouveau fichier dont le nom est identique à l'original, auquel est ajouté l'extension `.gpg`. Si vous spécifiez `--armor`, le nouveau fichier aura le même nom que l'original, mais avec l'extension `.asc`.

Voici comment chiffrer un fichier en utilisant le format ASCII et en choisissant le nom du fichier résultant :

```
# gpg --armor
  ➥  --output encryptedfile.asc
  ➥  --encrypt SecretPasswordList.txt
```

Les options `--armor` et `--output` doivent être utilisées avant `--sign` ou `--encrypt`.