Do It Yourself Privacy Guard (DIYPG)
Précision de certaines spécifications

Vincent Dugat

Janvier 2020

Table des matières

1 Commandes de l'interprète **1**
1.1 V1.1 : Gestion des clefs de chiffrement de l'utilisateur 1
1.2 V1.2 : Gestion des clefs de signature de l'utilisateur V1 1
1.3 V2 : Gestion des clefs publiques des contacts de l'utilisateur 2

On donne ici les spécifications des commandes que doit gérer l'interprète. Il y a deux niveaux V1 et V2. Le niveau V2 ajoute de nouvelles fonctionnalités mais demande également une modification des commandes de la V1.

1 Commandes de l'interprète

L'interprète gère les clefs de l'utilisateur sous forme d'une liste d'identificateurs associés chacun à un type (chiffrement ou signature). Un identificateur représente deux clefs : la clef publique et la clef privée associée.

L'interprète gère également les contacts sous forme d'un identificateur, du type, de l'identité nom, prénom, d'un commentaire et d'une liste de clefs selon le format ci-dessus.

1.1 V1.1 : Gestion des clefs de chiffrement de l'utilisateur

- *quit* : sort de l'interprète
- *listkeys* [<keyid>] : liste l'ensemble des clefs présentes en donnant leur identificateur et leur type (pas les clefs elles-mêmes)
- *rmkeys* <keyid> : remove keys = détruit un identificateur et les clefs associées
- *newkeys* <keyid> <type> : crée une nouvelle paire de clefs aléatoires avec l'identificateur et le type fournis
- *crypt* <fileIn> <fileOut> <keyid> : chiffre un fichier avec la clé publique de l'identificateur et sauve le résultat en base 64 dans le fichier de sortie. Le type de la clef doit être "crypt".
- *uncrypt* <filein> <fileout> <keyid> : déchiffre un fichier en base 64 avec la clé privée de l'identificateur et sauve le résultat dans le fichier de sortie. Le type de la clef doit être "crypt".
- *save* [<fileout>] : sauve l'ensemble des informations dans un fichier par défaut ou donné en paramètre. On peut chiffrer ce fichier avec une clef par défaut.
- *savepub* <keyid> <file> : sauve la clef publique <keyid> dans le fichier <file> en base 64.
- *load* [<filein>] : charge de fichier de sauvegarde.
- *show* <keyid> ["pub"] ["priv"] : affiche les clefs en b64.

1.2 V1.2 : Gestion des clefs de signature de l'utilisateur V1

On ajoute aux fonctions précédentes les fonctionnalités suivantes :

— *signtext* <filein> <keyid> <fileout> : signe un texte (ensemble d'octets), lu dans le fichier d'entrée, avec la clef privé de l'identificateur et écrit le résultat en base 64 dans le fichier <outfile>. Le type de la clef doit être "sign"
— *verifysign* <filein> <filesign> <keyid> : vérifie la signature d'un un texte. Le texte est lu dans le fichier d'entrée, la signature en base 64 est lue dans le fichier signature. On utilise la clef publique de l'identificateur. Le type de la clef doit être "sign".
— *show* <id> ["pub"] ["priv"] : affiche les clefs en b64.
— *certify* <id> : envoie une requête à l'autorité de certification (écriture dans un fichier).
— *revoke* <id> : idem pour une révocation.

1.3 V2 : Gestion des clefs publiques des contacts de l'utilisateur

On étend les possibilités de l'interprète en ajoutant la gestion des contacts. On doit pouvoir utiliser les anciennes commandes avec les nouvelles possibilités. Cela suppose de les adapter.

Exemple : *crypt msg.txt msgCode.txt idcontact/idclef.*
Le caractère "/" permet de donner l'identificateur du contact et celui de la clef s'il en a plusieurs.

Les contacts sont associés à un identificateur, des informations, et une liste de clefs : identificateur ; type ; nom ; prénom ; comment ; liste de clefs (id, type).

Exemple : jeannot, Jean Dupuis, voisin du dessus, clef : boulot crypt, perso crypt, boulotsign sign

— *listcontacts* [<idcontact>] [<nom>] : liste l'ensemble des contacts et les clefs associées (identificateur et type) ou celui correspondant au nom ou à l'identificateur donné.
— *addcontact* <id> : ajoute un nouveau contact, crée l'identificateur et affiche un menu de saisie.
— *modifycontact* <id> : affiche des informations et un menu de modification.
— *addkeys* <id> ou <nom> : ajoute une clef à un contact
— *rmcontact* <id> : supprime le contact et toutes ses clefs.
— *rmkeys* : étend la commande de la V1.1 pour supprimer une clef d'un contact (et seulement celle-là).
— *save, load, show, ..* : étendre ces commandes