Déployer un service d’annuaire de clefs OpenPGP

Damien Goutte-Gattat <dgouttegattat@incenp.org>

2020/05/30

Table des matières

Depuis quelques années, les développeurs de GnuPG proposent un nouveau mécanisme de distribution et de découverte des clefs OpenPGP. Au lieu de se reposer sur un réseau de serveurs de clefs où tout le monde peut librement déposer des clefs, le principe est de confier la distribution aux opérateurs de messagerie électronique, chaque opérateur devenant responsable de la distribution des clefs pour les adresses de son propre domaine (*i.e.*, l’opérateur de `example.org` a la charge de distribuer les clefs pour les adresses en `@example.org`).

Le mécanisme proposé s’appelle *Web Key Directory* ou WKD et est décrit dans un brouillon IETF [https://tools.ietf.org/html/draft-koch-openpgp-webkey-service-10]. J’en avais déjà parlé brièvement dans un précédent article [https://incenp.org/dvlpt/docs/publication-clefs-openpgp/index.html]. Ici, je vais décrire plus en détail la mise en œuvre du protocole côté serveur. Cet article s’adresse donc davantage aux opérateurs de serveurs de courrier électronique qu’aux utilisateurs.

1. Vue d’ensemble du protocole

Le protocole comprend deux parties distinctes :

- le *Web Key Directory* proprement dit est l’annuaire permettant à Bob d’obtenir la clef d’Alice ;
- le *Web Key Directory Update Protocol* est le protocole d’avitaillement, par lequel Alice fait connaître sa clef à son opérateur de messagerie afin que ce dernier l’ajoute à son annuaire.

Les deux parties peuvent s’utiliser indépendamment l’une de l’autre : un annuaire WKD peut ne pas être avitaillé via le *WKD Update Protocol* (il peut par exemple être renseigné « manuellement » par l’opérateur, comme on le verra plus loin), et inversement le *WKD Update Protocol* peut servir à avitailler d’autres méthodes de distribution que WKD (par exemple le DNS, avec DANE OpenPGP [https://tools.ietf.org/html/rfc7929]).

La combinaison du WKD et du *WKD Update Protocol* est parfois appelé *Web Key Service* ou WKS.

1.1. Le Web Key Directory

Quand Bob veut obtenir la clef d'Alice à partir de son adresse e-mail (`alice@example`), il construit une URL avec la forme suivante :

```
https://openpgpkey.example.org/.well-known/openpgpkey/example.org/hu/↪
kei1q4tipxxu1yj79k9kfukdhfy631xe?l=alice
```

La chaîne `kei1q4tipxxu1yj79k9kfukdhfy631xe` est le condensat SHA-1[1] de la partie locale de l'adresse d'Alice (`alice` donc, dans notre exemple), encodé en Z-Base32 [http://philzimmermann.com/docs/human-oriented-base-32-encoding.txt].

Vous pouvez utiliser la commande suivante pour calculer le condensat :

```
$ echo -n alice | openssl dgst -sha1 -binary | zbase32
kei1q4tipxxu1yj79k9kfukdhfy631xe
```

(bien sûr, en « situation réelle » vous n'avez jamais à le faire, c'est votre client de messagerie qui s'en charge.)

Le serveur `openpgpkey.example.org` doit répondre à une requête HTTP sur cette URL en renvoyant une copie de la clef d'Alice, sous la forme d'une *clef publique transférable* (*Transferable Public Key* ou TPK) conforme au standard OpenPGP. La clef reçue est directement importable dans le trousseau public de Bob.

Notez la répétition du nom de domaine `example.org` dans l'URL ci-dessus (une fois dans le nom d'hôte, une fois dans le chemin de la ressource). L'URL est construite ainsi afin de faciliter le déploiement d'un annuaire WKD qui gérerait plusieurs domaines. Dans le même ordre d'idée, le composant `openpgpkey` du nom d'hôte, qui semble redondant avec la ressource « bien connue » `.well-known/openpgpkey`, offre un niveau d'indirection qui permet d'héberger l'annuaire sur une autre machine que celle située derrière le nom `example.org`.

Les premières versions du brouillon WKD utilisaient, pour fournir ce niveau d'indirection, un enregistrement SRV [https://tools.ietf.org/html/rfc2782] plutôt qu'un nom d'hôte avec un composant fixé. Malheureusement, cela posait trop de problèmes aux développeurs Javascript, ce langage (ou plutôt son environnement d'exécution, dans les navigateurs) n'offrant pas de fonctionnalités de résolution DNS digne de ce nom.

Si une indirection n'est pas nécessaire, l'opérateur de `example.org` peut diffuser la clef d'Alice sous une URL légèrement plus simple, qu'on appelle la *méthode directe* (par opposition à l'URL précédente, qui représente la *méthode avancée*) :

```
https://example.org/.well-known/openpgpkey/hu/↪
kei1q4tipxxu1yj79k9kfukdhfy631xe?l=alice
```

Les opérateurs déployant WKD sont libres de choisir de supporter la méthode directe ou la méthode avancée, selon qu'ils ont besoin du niveau d'indirection offert par la méthode avancée ou non. Les clients eux doivent essayer en premier lieu la méthode avancée, puis se rabattre sur la méthode directe si le sous-domaine `openpgpkey` n'existe pas.

1.2. Le Web Key Directory Update Protocol

Pour déposer sa clef publique dans l'annuaire WKD de son fournisseur de messagerie, Alice doit suivre les étapes suivantes.

Premièrement, elle doit demander à son fournisseur l'*adresse de soumission*, par une requête HTTP sur

[1] SHA-1 n'est pas utilisé pour fournir une quelconque sécurité ici ; son seul intérêt est de transformer la partie locale de l'adresse e-mail en une chaîne de taille fixe et utilisant un jeu de caractères restreint. Aucune propriété cryptographique (résistance aux collisions ou aux recherches de pré-image) n'est attendue.

```
https://openpgpkey.example.org/.well-known/openpgpkey/example.org/↪
submission-address
```

(méthode « avancée ») ou bien (si `openpgpkey.example.org` n'existe pas) sur

```
https://example.org/.well-known/openpgpkey/submission-address
```

(méthode « directe »). Le serveur répond par une seule ligne contenant l'adresse e-mail à laquelle Alice devra envoyer sa clef.

Deuxièmement, Alice doit obtenir la clef publique de l'annuaire WKD, qu'elle devra utiliser pour chiffrer son message. Elle utilise pour ça le protocole *Web Key Directory* décrit en section précédente, sur l'adresse de soumission qu'elle vient de recevoir.

Troisièmement, Alice envoie un message à l'adresse de soumission indiquée, chiffré avec la clef publique qu'elle vient d'obtenir et contenant une copie de sa propre clef publique. Elle reçoit alors une demande de confirmation, envoyée à l'adresse `alice@example.org` et chiffrée avec la clef publique qu'elle vient d'envoyer.

Quatrièmement, Alice déchiffre la demande de confirmation, dans laquelle elle trouve un nonce. Elle répond à la demande de confirmation en renvoyant le nonce. Ce faisant, elle prouve à l'opérateur de l'annuaire i) qu'elle contrôle l'adresse e-mail (puisqu'elle a reçu la demande de confirmation) et ii) qu'elle possède la clef privée correspondant à la clef publique qu'elle a soumise à l'annuaire (puisqu'elle a pu déchiffrer la demande de confirmation et en extraire le nonce). Conséquemment, l'opérateur accepte de publier sa clef.

2. Déploiement d'un annuaire en lecture seule

Dans cette section, nous verrons comment déployer un simple annuaire WKD, *sans* le service d'avitaillement fourni par le *WKD Update Protocol*. Ce type de déploiement peut par exemple convenir pour un petit serveur personnel qui n'héberge qu'une petite poignée d'adresses. Ça peut aussi servir de base pour la mise en œuvre d'un système d'avitaillement personnalisé (à partir d'une base de données d'utilisateurs par exemple).

Les manipulations décrites ci-dessous peuvent, au choix, être réalisées directement sur le serveur web qui hébergera l'annuaire ou bien sur une machine locale à partir de laquelle on copiera les fichiers vers le serveur.

Installez le programme gpg-wks-server, l'implémentation de référence du protocole. Ce programme est normalement fourni avec GnuPG, mais il peut être empaqueté séparément sur votre distribution (sur Debian, il est dans le paquet nommé… `gpg-wks-server`, tout simplement).

Créez un dossier qui servira de racine pour l'annuaire :

```
$ mkdir ~/wkd
$ chmod 0751 ~/wkd
```

Le `chmod` est pour retirer à tout le monde (à part l'utilisateur et le groupe propriétaires du dossier) le droit de lecture sur le dossier. `gpg-wks-server` est psychorigide sur cette question et refusera catégoriquement de travailler sur un dossier lisible par tout le monde.

Créez ensuite sous cette racine un dossier pour chaque domaine pour lequel votre annuaire distribuera des clefs, puis initialisez l'annuaire avec la commande `--list-domains` de `gpg-wks-server` :

```
$ mkdir ~/wkd/example.org
$ gpg-wks-server -C ~/wkd --list-domains
gpg-wks-server: domain example.org: subdir 'pending' created
gpg-wks-server: domain example.org: subdir 'hu' created
gpg-wks-server: domain example.org: submission address not configured
```

```
example.org
```

Vous pouvez tranquillement ignorer le message `submission address not configured`, puisque cet annuaire ne sera pas couplé au *WKD Update Protocol*. Vous pouvez maintenant déposer des clefs dans l'annuaire. Si la clef que vous voulez déposer est dans le trousseau publique de votre compte utilisateur, obtenez son empreinte :

```
$ gpg -k alice@example.org
pub   rsa2048 2020-05-13 [SC] [expires: 2022-05-11]
              7685DC4214D727BB011BD6B754B4CC7749CAE7C3
uid           [ultimate] Alice <alice@example.org>
sub   rsa2048 2020-05-13 [E]
```

Puis passez-là à `gpg-wks-server` :

```
$ gpg-wks-server -C ~/wkd --install-key 7685DC4214D727BB011BD6B754B4CC7749CAE7C3 ↪
alice@example.org
gpg-wks-server: key 7685DC4214D727BB011BD6B754B4CC7749CAE7C3 published for ↪
alice@example.org
```

À la place d'une empreinte, vous pouvez aussi fournir à `gpg-wks-server` un fichier contenant directement la clef à publier.

L'annuaire est prêt à être publié. Copiez le dossier `~/wkd` vers votre serveur web si vous n'y étiez pas déjà et configurez votre logiciel serveur pour rendre le dossier disponible.

À ce moment-là, vous devrez décider si vous voulez utiliser des URL au format « direct » ou au format « avancé ». Pour le format direct, publiez le dossier `~/wkd/example.org` sous le nom `.well-known/openpgpkey`. Par exemple, en supposant que vous utilisez Apache httpd et que vous avez copié le dossier `wkd` dans `/var/www/wkd`, vous pouvez ajouter les lignes suivantes dans la configuration de l'hôte `example.org` :

```
Alias /.well-known/openpgpkey /var/www/wkd/example.org
<Directory /var/www/wkd/example.org>
  Require all granted
</Directory>
```

Si vous optez pour le format avancé, ajoutez dans votre zone DNS des enregistrements `A` et `AAAA` sous le nom `openpgpkey`, faites pointer ces enregistrements vers le serveur web, et ajoutez les lignes suivantes dans la configuration de l'hôte `openpgpkey.example.org` :

```
Alias /.well-known/openpgpkey /var/www/wkd
<Directory /var/www/wkd>
  Require all granted
</Directory>
```

Notez la différence : dans le cas où on utilise le format avancé, c'est la racine de l'annuaire qui est publiée, tandis que dans le format direct, c'est le sous-dossier du domaine `example.org`.

Pour tester votre annuaire, sur votre machine locale vous pouvez utiliser la commande `--locate-external-keys` de GnuPG :

```
$ gpg --locate-external-keys alice@example.org
```

Pour tester avec Thunderbird et Enigmail, commencez simplement à composer un message pour `alice@example.org`. Si vous n'avez pas déjà la clef d'Alice dans votre trousseau, Enigmail interrogera automatiquement l'annuaire de `example.org` et rapatriera la clef.

3. Déploiement d'un Web Key Service complet

Dans cette section, nous verrons comment déployer un annuaire WKD couplé à un service d'avitaillement implémentant le *WKD Update Protocol*.

On supposera pour garder les choses simples que le serveur web qui hébergera l'annuaire et le serveur de messagerie qui recevra les soumissions de clefs sont sur la même machine. Si vous voulez séparer les deux, toutes les manipulations ci-dessous sont à faire sur la machine où tourne le serveur de messagerie. Pour la publication, il vous appartiendra de mettre en place un dispositif de synchronisation permettant de copier le contenu de l'annuaire depuis le serveur de messagerie vers le serveur web — soit à intervalle régulier, soit à chaque fois que le contenu de l'annuaire changera (via *inotify* par exemple).

3.1. Préparation de l'annuaire

Commencez par installer GnuPG et `gpg-wks-server` sur votre serveur, puis créez un compte utilisateur pour le service d'avitaillement, qu'on appelera ici `wks` :

```
# groupadd --system wks
# useradd --comment "Web Key Service" --home-dir "/var/lib/gnupg/wks ↪
  -g wks --no-user-group --system --shell /bin/bash
```

Créez ensuite le dossier qui abritera l'annuaire, `/var/lib/gnupg/wks`. Notez qu'il s'agit là du dossier par défaut utilisé par `gpg-wks-server` ; vous êtes libres de choisir un autre dossier, mais dans ce cas vous devrez rajouter l'option `-C` *`DOSSIER`* à toutes les invocations de `gpg-wks-server` ci-dessous. Où que vous décidiez de créer le dossier, donnez-le à l'utilisateur `wks` et assurez-vous que le « reste du monde » n'y a pas accès en lecture.

```
# mkdir -p /var/lib/gnupg/wks
# chown wks:wks /var/lib/gnupg/wks
# chmod 0751 /var/lib/gnupg/wks
```

Les commandes suivantes sont à faire sous le compte *`wks`*. Créez dans le dossier */var/lib/gnupg/wks* un sous-dossier pour chacun de vos domaines dont vous voulez publier les clefs et initialisez l'annuaire :

```
$ mkdir example.org
$ gpg-wks-server --list-domains
gpg-wks-server: domain example.org: subdir 'pending' created
gpg-wks-server: domain example.org: subdir 'hu' created
gpg-wks-server: domain example.org: submission address not configured
example.org
```

Décidez de ce que sera l'adresse de soumission. Ici, nous choisirons *`wks-submission@example.org`*. Si vous hébergez plusieurs domaines, chaque domaine *peut* avoir sa propre adresse de soumission, mais vous pouvez aussi utiliser une seule et même adresse pour tous les domaines de votre annuaire.

Créez la clef associée à l'adresse de soumission :

```
$ gpg --batch --passphrase '' --quick-gen-key wks-submission@example.org
```

Notez que la clef privée n'a pas de phrase de passe, puisqu'elle doit pouvoir être utilisée de manière autonome. Notez aussi qu'elle expirera deux ans après sa création, il vous appartiendra de penser à repousser sa date d'expiration avant l'échéance. Pour demander dès le départ que la clef n'expire jamais, vous pouvez ajouter les mots-clefs `default default never` à la fin de la commande ci-dessus (les deux premiers mots-clefs demandent à GnuPG d'utiliser les algorithmes et le profil d'utilisation par défaut, le dernier demande que la clef générée n'expire jamais).

Finalement, publiez la clef du compte de soumission dans l'annuaire puis publiez l'adresse de soumission elle-même :

```
$ gpg -k wks-submission@example.org
pub   rsa2048 2020-05-29 [SC] [expires: 2022-05-29]
```

```
      0D2652F979E05D919B64C808AAB89C4C439B2F67
uid           [ultimate] wks-submission@example.org
sub   rsa2048 2020-05-29 [E]

$ gpg-wks-server --install-key 0D2652F979E05D919B64C808AAB89C4C439B2F67 wks-submission@example.org
gpg-wks-server: key 0D2652F979E05D919B64C808AAB89C4C439B2F67 published for ↪
wks-submission@example.org

$ echo wks-submission@example.org > example.org/submission-address
```

3.2. Configuration du serveur de messagerie

L'annuaire étant en place, il faut maintenant configurer le serveur de messagerie afin que les messages envoyés à l'adresse de soumission choisie dans la section précédente (`wks-submission@example.org`) soit passés au programme `gpg-wks-server`, appelé sous le compte utilisateur responsable de l'annuaire (`wks`).

Il y a bien sûr plusieurs façons de procéder, qui vont dépendre de votre serveur de messagerie et de sa configuration déjà existante. Il n'est pas possible de couvrir toutes les configurations possibles et en fin de compte je vous renvoie vers la documentation de votre logiciel serveur. Ce qui suit est à titre illustratif.

L'approche probablement la plus simple est de rediriger les messages envoyés à `wks-submission@example.org` vers le compte local `wks`, puis d'installer la règle Procmail suivante dans le fichier `~/.procmailrc` du compte `wks` :

```
:0
|gpg-wks-server --receive --from wks-submission@example.org --send
```

La commande `--receive` instruit `gpg-wks-server` de lire le message qu'il reçoit sur son entrée standard et d'agir en fonction du contenu du message (si c'est une nouvelle soumission, envoyer une demande de confirmation ; si c'est une confirmation, vérifier qu'elle correspond à une soumission en cours et que le nonce est correct). L'option `--send` demande à ce que la réponse soit directement renvoyée au système de messagerie, via la commande `/usr/sbin/sendmail`.

Une autre configuration possible avec Postfix est d'avoir les lignes suivantes dans le `main.cf` :

```
virtual_alias_maps = hash:/etc/postfix/virtual
mailbox_command_maps = hash:/etc/postfix/local_commands
```

La première ligne installe une « table d'alias virtuels », permettant entre autres choses de rediriger une adresse virtuelle vers un compte local. La seconde ligne installe une table permettant de spécifier la commande de livraison à exécuter pour un compte local donné. On ajoutera la ligne suivante à la table `/etc/postfix/virtual` :

```
wks-submission@example.org  wks
```

et la ligne suivante à la table `/etc/postfix/local_commands` :

```
wks  gpg-wks-server --receive --from wks-submission@example.org --send
```

On n'oubliera pas bien sûr d'exécuter `postmap` sur les tables après édition.

La dernière chose à faire est d'installer une tâche cron pour purger régulièrement les soumissions non confirmées :

```
42 3 * * * gpg-wks-server --cron
```

Ajoutez cette tâche à la *crontab* de l'utilisateur `wks`. Modifiez la périodicité comme bon vous semble. À chaque invocation, **gpg-wks-server --cron** supprimera les soumissions non-confirmées vieilles de plus de trois jours.

3.3. Configuration du serveur web

L'annuaire est maintenant prêt et connecté au système de messagerie pour recevoir les soumissions de clefs, il ne reste plus qu'à le rendre accessible via le web.

En supposant i) que votre serveur web est sur la même machine que votre serveur mail, ii) que votre serveur est Apache httpd, et iii) que vous avez opté pour le format d'URL dit direct, alors il devrait vous suffire d'ajouter les lignes suivantes dans la configuration du *virtual host* servant le domaine `example.org` :

```
Alias /.well-known/openpgpkey /var/lib/gnupg/wks/example.org
<Directory /var/lib/gnupg/wks/example.org>
  Require all granted
</Directory>
```

Si vous avez plusieurs domaines, vous pouvez soit ajouter les lignes ci-dessus dans la configuration du *virtual host* de chaque domaine, soit opter pour le format d'URL avancé. Dans ce dernier cas, créez un nouveau *virtual host* qui sera dédié à l'annuaire et qui répondra au sous-domaine `openpgpkey` de chacun de vos domaine :

```
ServerName openpgpkey.example.org
ServerAlias openpgpkey.example.com
ServerAlias openpgpkey.example.net

Alias /.well-known/openpgpkey /var/lib/gnupg/wks
<Directory /var/lib/gnupg/wks>
  Require all granted
</Directory>
```

Notez que les requêtes WKD se font systématiquement via HTTPS, donc votre nouvel hôte doit avoir une configuration TLS correcte et un certificat valable pour tous vos domaines `openpgpkey.*`.

3.4. Test du service

Si vous utilisez Thunderbird et Enigmail, vous pouvez tester la publication de votre clef dans votre annuaire en ouvrant le « gestionnaire de clefs » (*Key Management*). Sélectionnez votre clef, puis lancez la commande *Upload to your provider's Web Key Directory* dans le menu Keyserver.

Enigmail se chargera alors d'envoyer le message de soumission de clef et vous devriez rapidement recevoir sur votre adresse un message provenant de `wks-submission@example.org`. Ouvrez ce message, et Enigmail reconnaîtra qu'il s'agit d'une demande de confirmation et vous demandera si vous souhaitez confirmer ou non la publication de votre clef. Acceptez et vous recevrez un nouveau message vous annonçant que votre clef a bien été publiée dans l'annuaire.

Si vous utilisez un client de messagerie sans support natif pour le protocole *WKD Update Protocol*, vous pouvez utiliser l'outil `gpg-wks-client` fourni par GnuPG. Notez que cet outil n'est pas forcément dans le `PATH` de votre interpréteur de commandes, il est par défaut installé dans `/usr/libexec` (parce qu'il n'est pas vraiment conçu pour être appelé directement par l'utilisateur, mais plutôt par les clients de messagerie).

La commande suivante permet de tester que le service d'annuaire pour `example.org` est bel et bien disponible :

```
$ /usr/libexec/gpg-wks-client --supported alice@example.org
```

Si la commande renvoie une valeur différente de zéro, inutile d'aller plus loin, il y a un problème avec votre annuaire, probablement dans la configuration du serveur web.

Créez ensuite le message de soumission :

```
$ /usr/libexec/gpg-wks-client 7685DC4214D727BB011BD6B754B4CC7749CAE7C3 ↪
  alice@example.org > submit-msg
```

Le fichier *submit-msg* contient le message de soumission prêt à être envoyé via votre client de messagerie habituel.

Si vous avez un programme `/usr/sbin/sendmail` fonctionnel sur votre machine (qu'il s'agisse réellement de Sendmail ou d'un programme utilisant la même interface, comme `msmtp` par exemple), vous pouvez demander à `gpg-wks-client` d'envoyer le message lui-même avec l'option `--send`.

Lorsque vous recevrez la demande de confirmation dans votre client de messagerie, déchiffrez-là et sauvez-là dans un fichier, puis passez ce fichier à `gpg-wks-client` pour créer le message de confirmation :

```
$ /usr/libexec/gpg-wks-client --read < confirm-request > confirm-msg
```

Comme à l'étape précédente, envoyez le fichier *confirm-msg* via votre client habituel, ou utilisez l'option `--send` pour laisser `gpg-wks-client` s'en charger.

A. À propos de ce document