Sadek Adel 07/05/2026

Déployer un service d'annuaire de clefs OpenPGP pour son domaine

Introduction

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`).

Concrètement c'est un serveur qui hébergera toutes les clefs publiques de nos utilisateurs et via un enregistrement DNS il sera reconnaissable par les autres clients mails qui souhaite communiquer avec des users de notre domaine et recevra sa clef public qui permettra de chiffrer le mail e tl'utilisateur via sa clef privé pourra dechiffrer l'email

Source

Je me baserai sur ce tuto en grande partie : https://linuxfr.org/users/gouttegd/journaux/deployer-un-service-d-annuaire-de-clefs-openpgp-pour-son-domaine

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).

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

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.

Note

Vous pouvez utiliser la commande suivante pour calculer le condensat :

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

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`.

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

```
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 indiqué, 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, envoyé à l'adresse `alice@example.org` et chiffré avec la clef publique qu'elle vient d'envoyer.

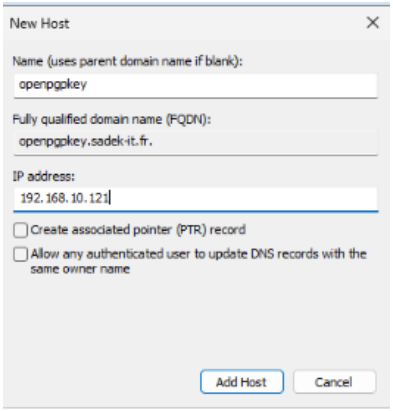
Quatrièmement, Alice déchiffre la demande de confirmation, dans laquelle elle trouve un nonce. Elle répond à la demande de confirmation en renvoyant le nonce. Ce faisant, elle prouve à l'opérateur de l'annuaire ① qu'elle contrôle l'adresse e-mail (puisqu'elle a reçu la demande de confirmation) et ② qu'elle 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.

Déploiement d'un annuaire en lecture seule

Il faut d'abord installer le paquet `gpg-wks-server`

`Il faut aussi créer l'enregistrement A ou CNAME « openpgpkey.example.org »`

`Je le créer ici dans mon AD`



Sadek Adel 07/05/2026

Création du repertoire wkd

```
mkdir -p ~/wkd/sadek-it.fr
```

On le créer dans le répertoire root nous lui attribuerons les droits adéquats ensuite

```
chmod 0751 ~/wkd
```

Ensuite on initialise l'annuaire qui distribuera les clefs

```
root@hanbal:~# gpg-wks-server -C ~/wkd --list-domains
gpg-wks-server: domain sadek-it.fr: subdir 'pending' created
gpg-wks-server: domain sadek-it.fr: subdir 'hu' created
gpg-wks-server: domain sadek-it.fr: submission address not configured
sadek-it.fr
root@hanbal:~#
```

Création de la paire de clefs

Je dois me rendre dans le repertoire de mon utilisateur

Cd /home/asadek

Et je genere la paire de clef gpg

gpg --full-generate-key

Ensuite on me demandera une passphrase

```
root@hanbal:/home/asadek# gpg --full-generate-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
  (14) Existing key from card
Your selection?
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 1y
Key expires at Wed 05 May 2027 23:43:58 CEST
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: Adel Sadek
Email address: asadek@sadek-it.fr
Comment: Clef de Monsieur Sadek Adel
```

Sadek Adel 07/05/2026

```
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: directory '/root/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/D671421F4938D5C0A3C0EF66CF4DB1AC9D46564A.rev'
public and secret key created and signed.

pub   rsa4096 2026-05-05 [SC] [expires: 2027-05-05]
      D671421F4938D5C0A3C0EF66CF4DB1AC9D46564A
uid                      Adel Sadek (Clef de Monsieur Sadek Adel) <asadek@sadek-it.fr>
sub   rsa4096 2026-05-05 [E] [expires: 2027-05-05]
```

Ensuite je verifie la presence de la clef et son empreinte avec cette commande

gpg -k asadek@sadek-it.fr

```
root@hanbal:/home/asadek# gpg -k asadek@sadek-it.fr
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2027-05-05
pub   rsa4096 2026-05-05 [SC] [expires: 2027-05-05]
      D671421F4938D5C0A3C0EF66CF4DB1AC9D46564A
uid           [ultimate] Adel Sadek (Clef de Monsieur Sadek Adel) <asadek@sadek-it.fr>
sub   rsa4096 2026-05-05 [E] [expires: 2027-05-05]
```

Ensuite exporter la clef publique

gpg --export asadek@sadek-it.fr > asadek-public.gpg

```
root@hanbal:/home/asadek# gpg --export asadek@sadek-it.fr > asadek-public.gpg
root@hanbal:/home/asadek# ls
asadek-public.gpg  Maildir
root@hanbal:/home/asadek#
```

Ensuite pour installer la clef dans le WKD

gpg-wks-server -C ~/wkd --install-key asadek-public.gpg asadek@sadek-it.fr

```
root@hanbal:/home/asadek# gpg-wks-server -C ~/wkd --install-key asadek-public.gpg asadek@sadek-it.fr
gpg-wks-server: key D671421F4938D5C0A3C0EF66CF4DB1AC9D46564A published for 'asadek@sadek-it.fr'
root@hanbal:/home/asadek#
```

Publication du WKD

Sadek Adel 07/05/2026

Toutes les requetes vers le FQDN qu'on a créer plus tot seront renvoyer vers le reverse proxy qui lui renverra vers le WKS

Coté apache du WKS

```
je precice ceci dans la conf
Alias /.well-known/openpgpkey /var/www/wkd/sadek-it.fr
<Directory /var/www/wkd/sadek-it.fr>
  Require all granted
</Directory>
```

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

L'alias ne marque pas je vais tout simplement synchroniser les deux répertoires via rsync

rsync -av --delete /root/wkd/sadek-it.fr/ /var/www/wkd/sadek-it.fr/

chown -R www-data:www-data /var/www/wkd

chmod -R 755 /var/www/wkd

Coté reverse proxy nginx

```
##################################WKS#########################
# Bloc HTTPS (port 443)
server {
    listen 443 ssl;
    server_name openpgpkey.sadek-it.fr;

    ssl_certificate     /etc/letsencrypt/live/openpgpkey.sadek-it.fr/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/openpgpkey.sadek-it.fr/privkey.pem;

    location / {

 proxy_pass https://wks/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # Redirection HTTP vers HTTPS
        if ($scheme != https) {
            return 301 https://$host$request_uri;
        }
    }

location /.well-known/acme-challenge/ {
        alias /var/www/html/.well-known/acme-challenge/;
        try_files $uri =404;
    }
}
```

Sadek Adel 07/05/2026

Test avec gpg locate external key

Cette commande deùande à gpg de chercher la clef public de l'utilisateur sur le serveur openpgp du domaine

```
root@Srv-Nginx:/etc/nginx/sites-enabled# gpg --locate-external-keys asadek@sadek-it.fr
gpg: clef CF4DB1AC9D46564A : clef publique « Adel Sadek (Clef de Monsieur Sadek Adel) <asadek@sadek-it.fr> » importée
gpg:       Quantité totale traitée : 1
gpg:                   importées : 1
pub   rsa4096 2026-05-05 [SC] [expire : 2027-05-05]
      D671421F4938D5C0A3C0EF66CF4DB1AC9D46564A
uid         [ inconnue] Adel Sadek (Clef de Monsieur Sadek Adel) <asadek@sadek-it.fr>
sub   rsa4096 2026-05-05 [E] [expire : 2027-05-05]

root@Srv-Nginx:/etc/nginx/sites-enabled#
```

Mise en place de l'update

Je dois créer d'avitaillement qui s'appelera « wks »

Je créer la cle associer à l'adresse de submission ici

```
gpg --batch --passphrase '' --quick-gen-key wks-submission@sadek-it.fr
```

Ensuite publier la clef du compte dans l'annuaire et publier l'adresse de submission

```
sudo -u wks gpg --batch --passphrase '' --quick-gen-key wks-submission@sadek-it.fr
gpg --export wks-submission@sadek-it.fr > /tmp/wks-submission-public.gpg
gpg-wks-server -C /root/wkd --install-key /tmp/wks-submission-public.gpg
wks-submission@sadek-it.fr

« Si dans /var/www/html »
rsync -av --delete /root/wkd/sadek-it.fr/ /var/www/wkd/sadek-it.fr/

echo wks-submission@sadek-it.fr > sadek-it.fr/submission-address
```

```
root@hanbal:~/wkd# echo wks-submission@sadek-it.fr > sadek-it.fr/submission-address
root@hanbal:~/wkd#
```

```
Ensuite pour transport vers /var/www/html

echo wks-submission@sadek-it.fr > /var/www/wkd/sadek-it.fr/submission-address
chown wks:www-data /var/www/wkd/sadek-it.fr/submission-address
chmod 640 /var/www/wkd/sadek-it.fr/submission-address
```

Configurer Postfix pour rediriger l'adresse vers `wks`

Dans `/etc/aliases`

Sadek Adel 07/05/2026

```
wks-submission: wks
```

Puis

newaliases

systemctl reload postfix

Ensuite créer le fichier proxmail pour l'utilisateur wks

nano /home/wks/.procmailrc

```
SHELL=/bin/sh
PATH=/usr/bin:/bin
LOGFILE=$HOME/procmail.log
VERBOSE=yes

:0
* ^TO_wks-submission@sadek-it\.fr
| /usr/bin/gpg-wks-server -C /var/www/wkd --receive
```

chown wks:wks /home/wks/.procmailrc

chmod 600 /home/wks/.procmailrc

Ensuite dire à postfix d'utiliser Proxmail

postconf mailbox_command

Si c'est vide, ajoutez dans `/etc/postfix/main.cf`

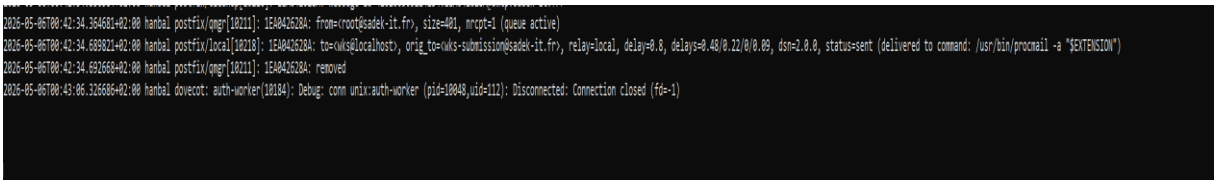
mailbox_command = /usr/bin/procmail -a "$EXTENSION"

Test envoie email vers wks submission

echo "test" | mail -s "test wks" wks-submission@sadek-it.fr

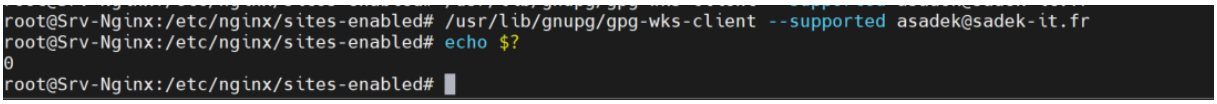
Puis regarder mail.log

Sadek Adel 07/05/2026



C’est parfait c’est bien envoyé à procmail

Ensuite nous tapons cette commande pour voir si le serveur annonce bien qu’il accepte les update si ça montre « 0 » c’est que le serveur supporte



Je créer un utilisateur choucheh que je vais faire publie sa clef sur le serveur

gpg --batch --passphrase '' --quick-add-key D96ADBB0A275A5E60FF776307FC1A23D9482AFC6 rsa3072 encr 1y

C’est la meme methode que tout à l’heure

Avec cette commande je récupérer le fingerprint de mon utilisateur choucheh qui est sur la machine cliente

gpg --list-keys --with-colons choucheh@sadek-it.fr | awk -F: '/^fpr:/ {print $10; exit}'

Envoie de la demande de submission

/usr/lib/gnupg/gpg-wks-client --create D96ADBB0A275A5E60FF776307FC1A23D9482AFC6 choucheh@sadek-it.fr > submit-msg

Pour voir la demande de submission je n’ai qua afficher le contenue du msg

Sadek Adel 07/05/2026

```
choucheh@Srv-Nginx:~$ cat submit-msg
From: choucheh@sadek-it.fr
To: wks-submission@sadek-it.fr
Subject: Key publishing request
Wks-Draft-Version: 3
MIME-Version: 1.0
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
        boundary="=-=01-ujdtk71i4oswag8pz88y=-="
Date: Wed, 06 May 2026 23:39:49 +0000


--=-=01-ujdtk71i4oswag8pz88y=-=
Content-Type: application/pgp-encrypted

Version: 1

--=-=01-ujdtk71i4oswag8pz88y=-=
Content-Type: application/octet-stream

-----BEGIN PGP MESSAGE-----
```

Il faut ensuite que j'envoie le message j'utiliserai swaks car il faut envoyer le fichier

swaks --server smtp.sadek-it.fr --data ~/submit-msg -t wks-submission@sadek-it.fr

Coté log procmail voila ce que je vois

```
procmail: [11461] Thu May  7 01:57:27 2026
procmail: Match on "(^((Original-)?(Resent-)?(To|Cc|Bcc)|(X-Envelope|Apparently(-Resent)?)-To):(.*[^-a-zA-Z0-9_.])?)wks-submission@sadek-it\.fr"
procmail: Executing "/usr/bin/gpg-wks-server,-C,/var/www/wkd,--receive"
procmail: Assigning "LASTFOLDER=/usr/bin/gpg-wks-server -C /var/www/wkd --receive"
procmail: Notified comsat: "wks@:/usr/bin/gpg-wks-server -C /var/www/wkd --receive"
From choucheh@Srv-Nginx  Thu May  7 01:57:27 2026
 Subject: Key publishing request
  Folder: /usr/bin/gpg-wks-server -C /var/www/wkd --receive               5380
gpg-wks-server: t2body for level 0
gpg-wks-server: t2body for level 1
gpg-wks-server: t2body for level 1
gpg-wks-server: gpg: encrypted with RSA key, ID 9FA2AC691ABC92C6
gpg-wks-server: gpg: encrypted with 3072-bit RSA key, ID 910E938121618A57, created 2026-05-05
gpg-wks-server: gpg:       "wks-submission@sadek-it.fr"
gpg-wks-server: draft version 2 requested
gpg-wks-server: t2body for level 0
gpg-wks-server: new 'application/pgp-keys' message part
gpg-wks-server: fingerprint: D96ADBB0A275A5E60FF776307FC1A23D9482AFC6
gpg-wks-server:   addr-spec: choucheh@sadek-it.fr
gpg-wks-server: storing address 'choucheh@sadek-it.fr'
```


```
procmail: [11461] Thu May  7 01:57:27 2026
procmail: Match on "(^((Original-)?(Resent-)?(To|Cc|Bcc)|(X-Envelope|Apparently(-Resent)?)-To):(.*[^-a-zA-Z0-9_.])?)wks-submission@sadek-it\.fr"
procmail: Executing "/usr/bin/gpg-wks-server,-C,/var/www/wkd,--receive"
procmail: Assigning "LASTFOLDER=/usr/bin/gpg-wks-server -C /var/www/wkd --receive"
procmail: Notified comsat: "wks@:/usr/bin/gpg-wks-server -C /var/www/wkd --receive"
From choucheh@Srv-Nginx  Thu May  7 01:57:27 2026
 Subject: Key publishing request
  Folder: /usr/bin/gpg-wks-server -C /var/www/wkd --receive               5380
gpg-wks-server: t2body for level 0
gpg-wks-server: t2body for level 1
gpg-wks-server: t2body for level 1
gpg-wks-server: gpg: encrypted with RSA key, ID 9FA2AC691ABC92C6
gpg-wks-server: gpg: encrypted with 3072-bit RSA key, ID 910E938121618A57, created 2026-05-05
gpg-wks-server: gpg:       "wks-submission@sadek-it.fr"
gpg-wks-server: draft version 2 requested
gpg-wks-server: t2body for level 0
gpg-wks-server: new 'application/pgp-keys' message part
gpg-wks-server: fingerprint: D96ADBB0A275A5E60FF776307FC1A23D9482AFC6
gpg-wks-server:   addr-spec: choucheh@sadek-it.fr
gpg-wks-server: storing address 'choucheh@sadek-it.fr'
```

Sadek Adel 07/05/2026

Problème WKS : demande reçue mais aucune réponse envoyée

Le serveur WKS recevait bien les demandes envoyées à `wks-submission@sadek-it.fr` : Postfix les redirigeait correctement vers l'utilisateur local `wks`, puis Procmail exécutait `gpg-wks-server`. Cependant, les demandes restaient simplement stockées dans le dossier `pending/` et aucun mail de confirmation n'était envoyé à l'utilisateur. La solution a été d'ajouter l'option `--send` dans la règle Procmail, afin que `gpg-wks-server` traite la demande avec `--receive` puis envoie automatiquement le message de confirmation via Postfix.

Nouveau fichier procmail apres correction

Log procmail apres correction

Problème de réception IMAP après livraison Postfix

Postfix indiquait bien que les mails envoyés à `choucheh@sadek-it.fr` étaient livrés avec `status=sent`, mais le client mail restait vide. Le problème venait du fait que Procmail livrait probablement les messages dans une boîte locale classique, alors que Dovecot/IMAP lisait le dossier `~/Maildir`. La solution a été de configurer Procmail pour utiliser explicitement le format Maildir avec `MAILDIR=$HOME/Maildir` et `DEFAULT=$MAILDIR/`, afin que les messages apparaissent correctement dans le client mail.

Procmail de Chouche pas wks attention

Contournement du problème d'envoi automatique WKS

Lors de la configuration du Web Key Service, le serveur recevait bien les demandes envoyées à `wks-submission@sadek-it.fr`. Postfix redirigeait correctement le message vers l'utilisateur local `wks`, Procmail lançait bien `gpg-wks-server`, et la demande était bien déchiffrée puis stockée. Cependant, l'option `--send` de `gpg-wks-server` n'envoyait pas automatiquement le mail de confirmation.

Le fonctionnement attendu était le suivant :

```
Client GPG
   |
   | 1. Envoi de la demande WKS
   v
wks-submission@sadek-it.fr
   |
   | 2. Redirection Postfix
   v
Utilisateur local wks
   |
   | 3. Procmail exécute gpg-wks-server
   v
gpg-wks-server
   |
   | 4. Génération du mail de confirmation
   v
choucheh@sadek-it.fr
```

Le problème venait de l'étape 4 : `gpg-wks-server` traitait la demande, mais ne déclenchait pas correctement l'envoi du mail de confirmation.

Pour contourner ce problème, un script shell a été créé. Ce script demande d'abord à `gpg-wks-server` de générer le mail de réponse dans un fichier temporaire avec l'option `-o`, puis il envoie ce fichier manuellement avec `sendmail`.

```
Procmail
   |
   v
Script wks-receive-send.sh
   |
   | 1. gpg-wks-server génère la réponse dans un fichier temporaire
   |
   | 2. sendmail envoie ce fichier
   v
Mail de confirmation reçu par l'utilisateur
```

Emplacement prévu pour le script

Le script a été placé ici :

```
/usr/local/sbin/wks-receive-send.sh
```

Il est appelé automatiquement par Procmail à chaque mail reçu sur l'adresse de soumission WKS.

```
  GNU nano 7.2                                                    /usr/local/sbin/wks-receive-send.sh
#!/bin/sh

TMPMAIL=$(mktemp /tmp/wks-reply.XXXXXX.eml)

# Traite la demande WKS et génère la réponse dans un fichier
/usr/bin/gpg-wks-server --receive \
  -C /var/www/wkd \
  --from wks-submission@sadek-it.fr \
  -o "$TMPMAIL"

STATUS=$?

# Si gpg-wks-server a réussi et que le fichier n'est pas vide, on l'envoie
if [ "$STATUS" -eq 0 ] && [ -s "$TMPMAIL" ]; then
    /usr/sbin/sendmail -oi -t < "$TMPMAIL"
fi

rm -f "$TMPMAIL"

exit "$STATUS"
```

Emplacement prévu pour la configuration Procmail

Le fichier Procmail de l'utilisateur `wks` se trouve ici :

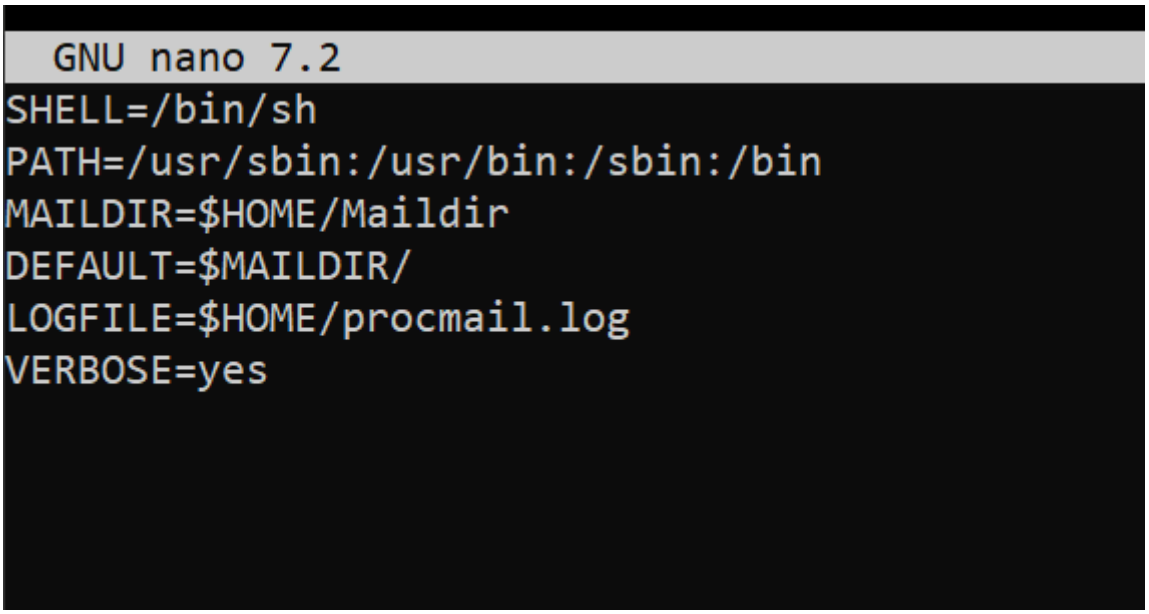
`/home/wks/.procmailrc`

Son rôle est simple : lorsqu'un mail arrive à `wks-submission@sadek-it.fr`, il lance le script `wks-receive-send.sh`.

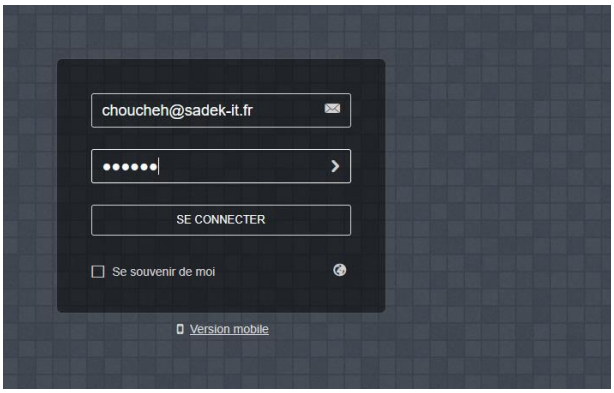
```
  GNU nano 7.2                                                                    /home/
SHELL=/bin/sh
PATH=/usr/sbin:/usr/bin:/sbin:/bin
LOGFILE=$HOME/procmail.log
VERBOSE=yes

:0
* ^TO_wks-submission@sadek-it\.fr
| /usr/local/sbin/wks-receive-send.sh
```

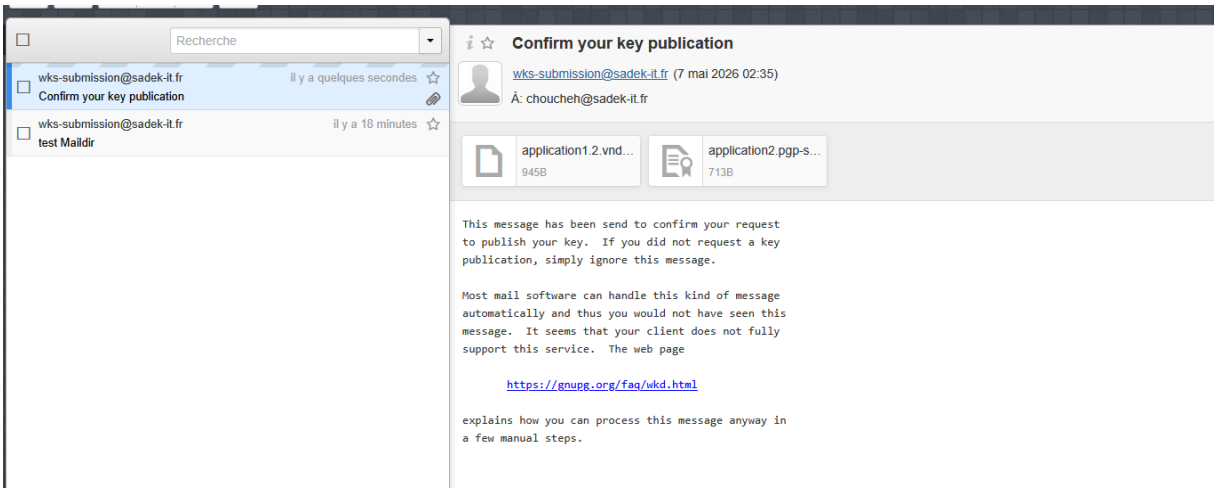
Grâce à ce contournement, la chaîne WKS devient fonctionnelle : la demande est reçue, traitée par `gpg-wks-server`, puis le mail de confirmation est envoyé manuellement via `sendmail`. Cela permet de conserver un fonctionnement automatisé malgré le problème rencontré avec l'option `--send`.

Sadek Adel 07/05/2026



Je me connecte à mon client mail pour voir le message renvoyer par wks

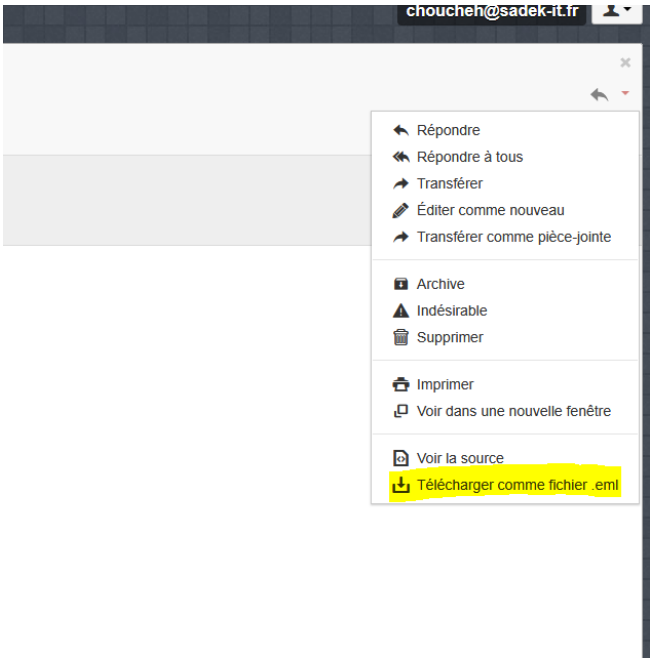




On voit bien maintenant le message de wks qui nous demande de confirmer la publication de clef nous devons maintenant télécharger le fichier et le dechiffrer avec notre clef prive

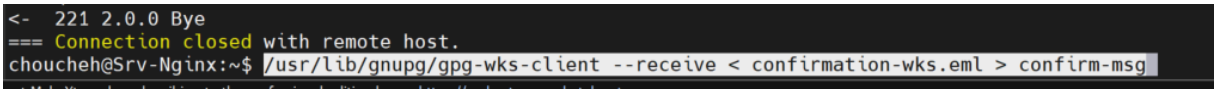
Sadek Adel 07/05/2026

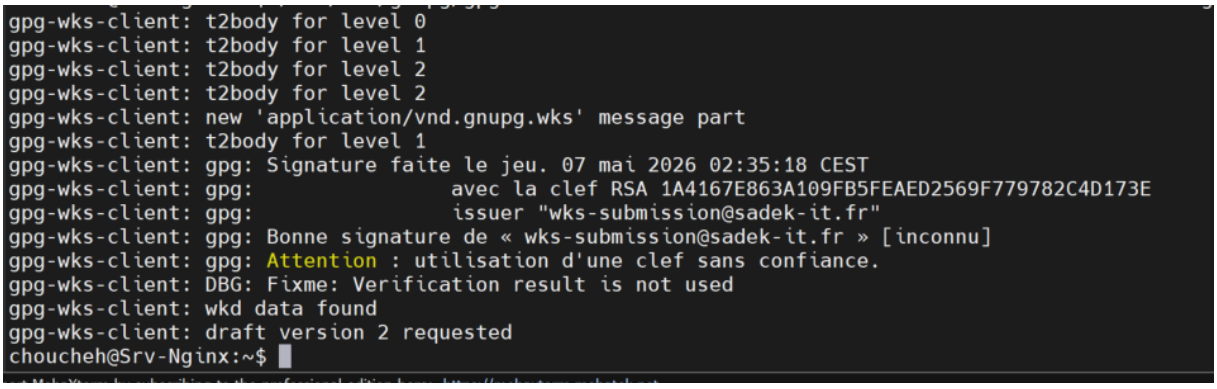
Je télécharge le mail au format .eml



Ensuite, sur la machine où se trouve la clé privée de choucheh, donc Srv-Nginx avec l'utilisateur choucheh, je fais :

/usr/lib/gnupg/gpg-wks-client --receive < confirmation-wks.eml > confirm-msg

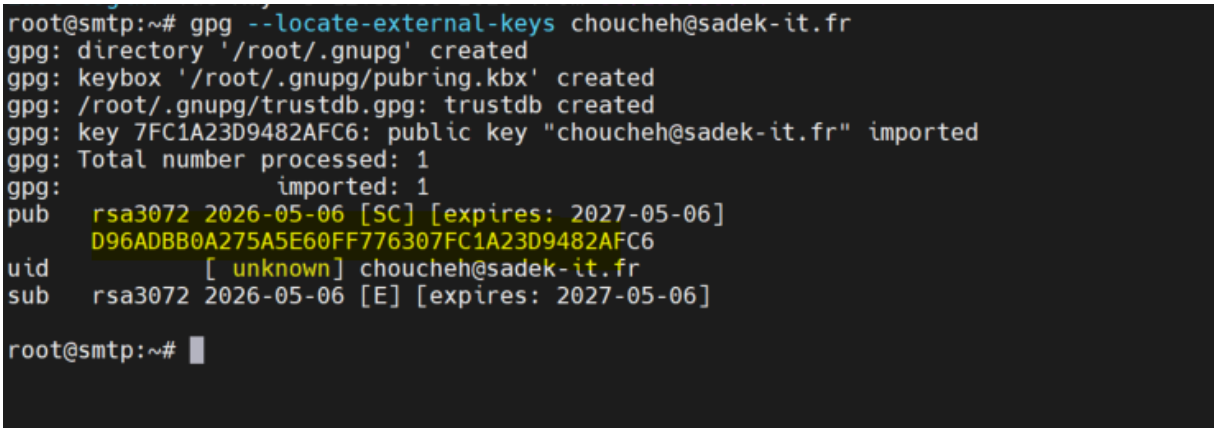




Ensuite j'envoie le message de confirmation

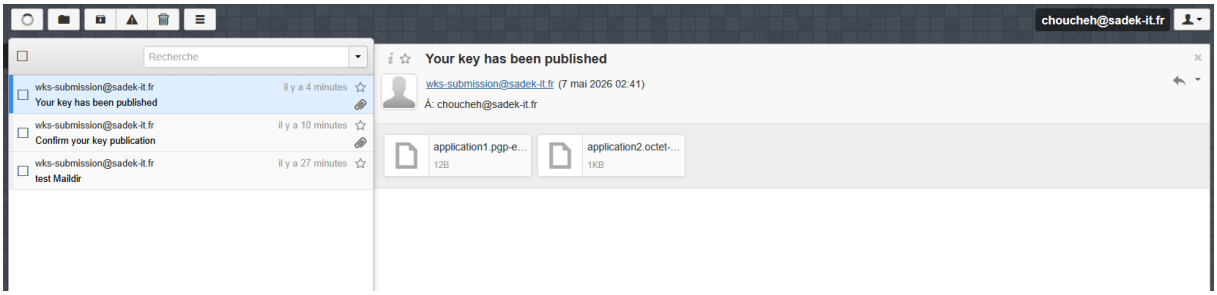
Sadek Adel 07/05/2026

Test de récupération de la clef publique de choucheh depuis une machine hors de notre réseau



La clef s'affiche notre serveur est donc fonctionnelle !

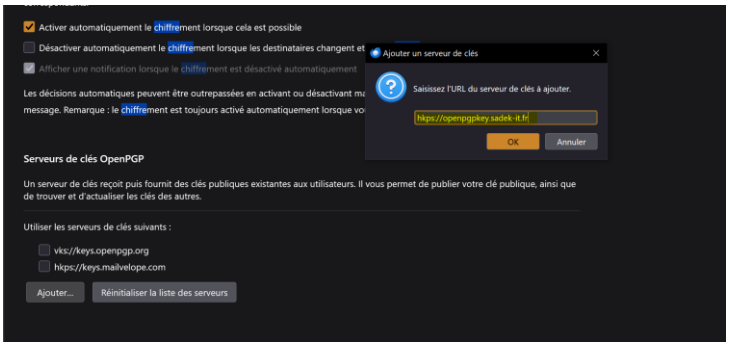
Nous avons aussi reçut un mail sur choucheh pour nous informer que notre clef a bien été publier



Test via Thunderbird

Je connecte les deux utilisateurs via mon client thunderbird

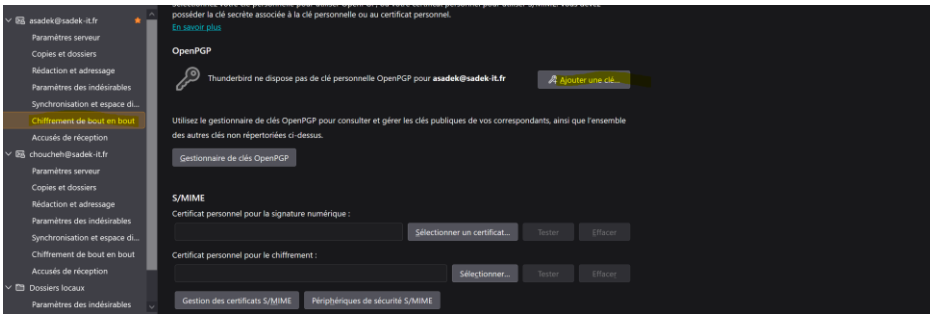
Sadek Adel 07/05/2026

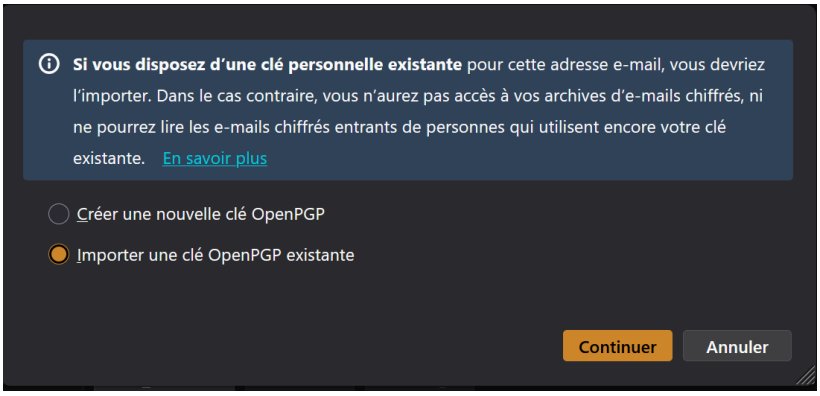


Il faut exporter la clef privé de chaque utilisateur comme cela

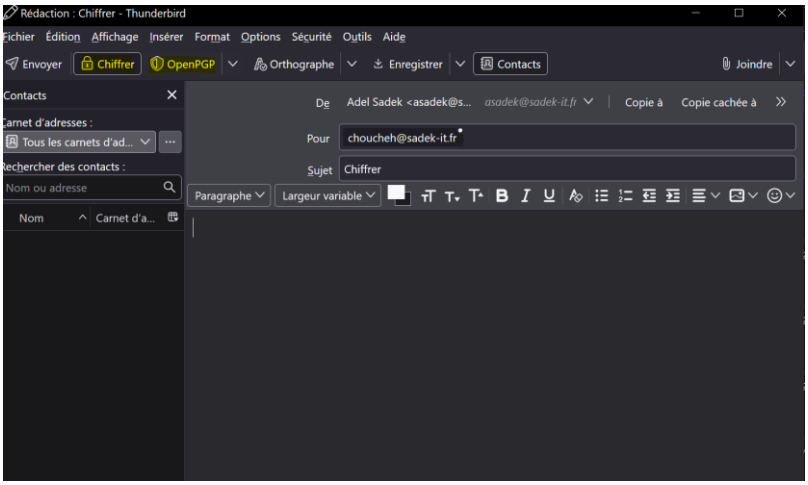
gpg --export-secret-keys --armor asadek@sadek-it.fr > asadek-private.asc

Ensuite je me l'envoie par email et j'enregistre dans un dossier





Sadek Adel 07/05/2026



Sadek Adel 07/05/2026