PGP

~

Sébastien Delcroix

seb [at] sebnet [dot] org

@Seb_Net

Download & check me @ : http://www.sebnet.org/prez/crypto_pgp{.pdf,.pdf.asc}

Sommaire

Introduction

Généralités

- Cryptographie
  - **intégrité** : le message reçu est strictement identique à celui qui a été envoyé
  - **confidentialité** : le message est uniquement compréhensible par toutes les personnes concernées
  - **authentification** : L'émetteur du message est vérifiable
- Cryptographie
  - Symétrique
    - Sécret partagé
  - Asymétrique
    - Clés publique et privée
- Pourquoi ?
  - Protéger ses données
  - Protéger ses communications
  - Authentifié un émetteur (signature, emails, fichiers, etc.)

PGP

- Pretty Good Privacy
  - Phil Zimmermann (1991)
- Standard OpenPGP
  - RFC 4880
- Biclé
  - 1 Publique
  - 1 Privé
- Master key
  - Biclé la plus importante
- Sub key
  - Signée par la master key
  - Utilisation à la place de la master key
  - => mise « offline » de la master key pour plus de sécurité

PGP vs X509

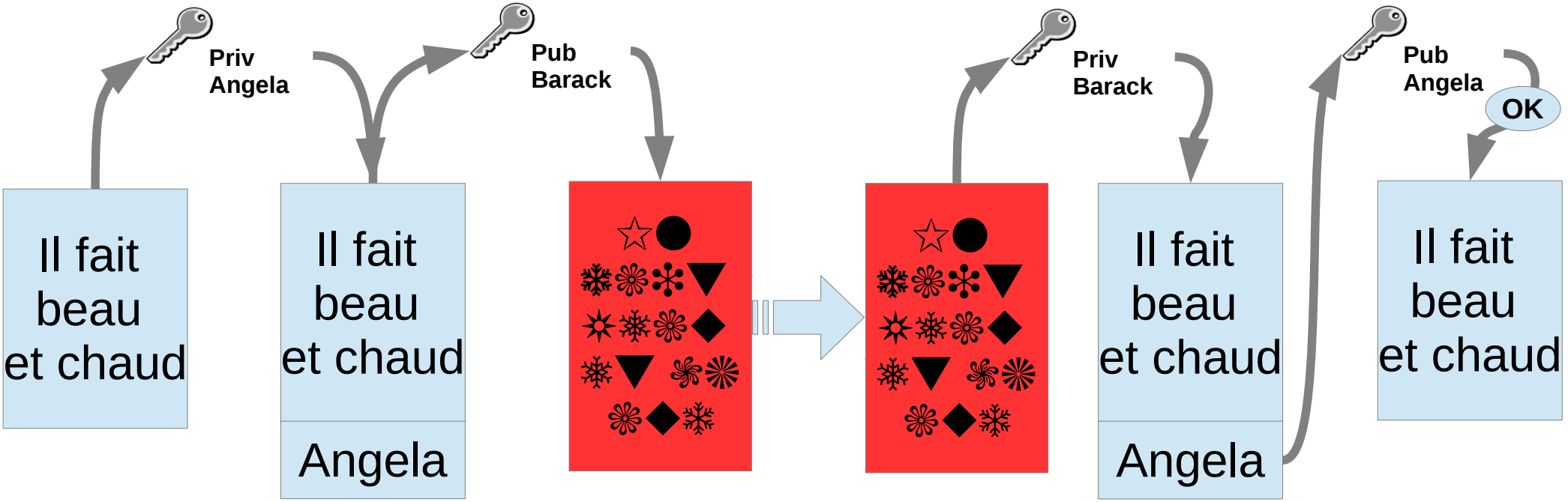
- PGP
  - Confiance de pair à pair => décentralisation
  - Distribution et signature des clés plus « compliquées »
- X509
  - Une entité centrale détient la confiance
  - Si entité compromise => risque pour tous les certificats qui en découlent

Signature et chiffrement asymétrique

Angela
clé privée
clé publique Barack

NSA
clé publique Barack
clé publique Angela

Barack
clé privée
clé publique Angela



Gnu Privacy Guard GPG

- **Libre** / projet Gnu
- Outils multi plateforme
  - Windows
  - Linux
    - Installé en standard dans de nombreuses versions de l'OS
    - Utilisé pour vérifier la signature des packages par exemple
  - MacOS X

Rappel

- Ce qu'on doit dire :
  - Chiffrer
  - Déchiffrer
  - Chiffrement
  - Déchiffrement
  - Décrypter lorsque l'on est la NSA ou un pirate
- Le reste est à oublier !!!

Gérer son trousseau de clés

Générer ses clés

- Choisir les algorithmes
  - **RSA and RSA (default)**
  - DSA and Elgamal
  - DSA (sign only)
  - RSA (sign only)
- La taille des clés
  - 1024, 2048, **4096**
- Date d'expiration
  - X jours, semaines, mois ou années
- Charger la machine lors de la génération des clés
  - => Ajouter de l'entropie (random) : lire des vidéos, bouger la souris, taper au clavier, faire des accès disques en même temps
  - Ça prend du temps !
  - Pas dans une machine virtuelle

Générer ses clés

```
alice@h0me:~$ gpg --gen-key
gpg (GnuPG) 1.4.14; Copyright (C) 2013 Free Software Foundation, Inc.
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)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 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) 1
Key expires at mar. 21 janv. 2014 16:26:02 CET
Is this correct? (y/N) y
You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Alice
Email address: alice@example.com
Comment:
You selected this USER-ID:
    "Alice <alice@example.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.
```

Générer ses clés

```
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.

Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 284 more bytes)
...+++++

Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 200 more bytes)


Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 256 more bytes)
+++++
gpg: /home/alice/.gnupg/trustdb.gpg: trustdb created
gpg: key 1F5C48DC marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2014-01-21
pub   4096R/1F5C48DC 2014-01-20 [expires: 2014-01-21]
      Key fingerprint = C9A8 DC4C 2815 B347 28B2  AD9C 0106 30A8 1F5C 48DC
uid                  Alice <alice@example.com>
sub   4096R/6D6A7CBC 2014-01-20 [expires: 2014-01-21]
```

PGP ID

Finger Print

Exporter sa clé publique

```
bob@h0me:~$ gpg --armor --export bob@example.com > bob_pub_key.asc

bob@h0me:~$ cat bob_pub_key.asc

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.14 (GNU/Linux)

mQINBFLdRsMBEAC1i7W5mWiVRxC/cVa34npo+hjBbwX27OO7pq9lBF9qvo6BZD9r
iPW2clVTzgPa4NKNp0vVwV+BNgfdrF3FotCP997fnGYvmcv2Zk/8ZZOoNub83r9L
0RNSZIAkx+ZTW70BA1+tSciQY3hlegVWpO7fzV/aqWQDS4VTjQ7VbhcDcJ+5SLVB
mG9MJpKw3JjKaaDNLdp9V77hyqaEI97gJhWwa4rO6Z0eTpdpco7i9ulu/pXv5wac
GyzsT2EdaFTKvXLssZYYrrO+A53W+pgBmiQzp82f+MAT6HMnJa07RaxKg9AQdSzT
1QLVHmNj+RvkNNTOnSuuNQ3oNEQHhJth35nPx8Oa+m4xWlrWLD9l9kPOmWbiV76Y
o8vpE9elVGbePQL3rimgWN7ZIIlVIwJx9a4izucbED+rlBOWVXcva9XQHK+jEDPR
JwRI2S05BGTXXkWlyI8A6FXSIlifWQ/5ly3LZECordJMw86O4lvOJ57agJfv+yXm
lOf9hWB+F9mur0Cp7wjFdzieqkkkoiiJgjMCTVYQIXG5Eu9JBlp5lMub5IhooQxg
hLa4WgCcDbL4zArEZnWcWNwlOv13hUeBEQADHpV2ZgMRLAUrF2rdbXsKsSda8LUw
5CjJVZdW+mSdEekmsfqxcYPd9yNcS1sMbnJKRm7+RmZY5eIklJfoMGrUdwARAQAB
tB5Cb2IgWmVzcG9uZ2UgPGJvYkBleGFtcGxlLmNvbT6JAj4EEwECACgFAlLdRsMC
GwMFCQABUYAGCwkIBwMCBhUIAgkKCwQWAgMBAh4BAheAAAoJEJ6GpMQNN6q/NmoQ
AKJLm6oXf/eMX6lRDdwrGmPvxd7MhobYW54fy8WWNTqvSLCW66LwP5g0n3fV86dA
[...]
plhfiNtuHlxZgYqXZuatO1TG+Z8BBctc/1VLqgYzPSSVj2lDPdyGYNBw1MvjtH1h
KxoX6Tx28j+n8iQCdEqXt7cGnDqHYXiyf/iuzWlgacebLXiIjr4hAzsoLXiuPfnb
J9CiI5PYWeCOx9E/ChrqnocCPEu+yJb6z4s2aGRKlJWgnNUHox4dWea6SB1G8sml
xoAbXd8HZQktMGnjwN4FTK5pA0BHfJZpdLHduQARAQABiQIlBBgBAgAPBQJS3UbD
AhsMBQkAAVGAAAoJEJ6GpMQNN6q/daIQAK9RV6y7GCkmcc+cjBncHhZauYLnynIL
wWsJcIw4FjRfwW7zg7V2EF0kKWsRXRjHljtkD3mmR4zvwmEHspLBRiukqLxEWAiq
GweLjblepEWmzd/4l3FCAw8GPIadpP2m9Z1QrjZXmk3MvADyQEd5B9wj+hmhuu/4
3YKv8GtQppBDpKN3zLJTHxcLZp/UdDZJed3UfFqLLH+gwCP0kj4Cr6GCX+aEwGUw
nSlPU52g2/uxuYbaw5GD/H+8EmY8QRNzUQgVAvj1Z81zvMzsEwYIVFIXQ2FUP3RG
JNSuNNuGfdnlQTs30vnbWvVMrVWPJyralp8+rB6aR6OKV4if++N03MYEjPZ0mRX4
2tHHucTpYyhxK1+wMiNvq3rnTpptMwphs0jhO/klMdA2l6ffcFVKy75W4d5Yl4ps
nJVBE/DYr5Uh4lVtDcmXMB9EL2oiL7lkx6Ocxbh6gTkY0ObA0Vxa0BOns7EaiVws
qYyJf/fSXB1zs0+UO0Om7jNDFDkKVBEBnpJ/Rt8qPXfBf/cZ3l8a8VlV23ZcaUKw
ldvwrsHJ1av8CspfxxMuIJU36kWTMyxD/q9kMHWaBk6t11tNL3U/d1LGIHdhyJoX
yBn+djb73Us3vDNt9e5PZJ8EVRVnOabOpcmsUJaX9JOstZnNsxXeUaop+AdqfLAq
ibibO7jAZ/Sh
=aYWx
-----END PGP PUBLIC KEY BLOCK-----
```

Échange de clés publiques

- Transmettre la clé via :
  - Un support amovible
  - Un site de stockage de clé publiques (ex : pgp.mit.edu)
- Pour indentifier l'interlocuteur
  - ID
  - Fingerprint
  - Carte d'identité ou passeport
- Vérifier la clé via l'ID et le Fingerprint
- Pour des « key signing party » il est préférable d'utiliser des outils qui permettent d'automatiser la signature, comme « pius ».

Importer une clé publique & lister les clés de son trousseau

```
alice@h0me:~/Documents$ gpg --import bob_pub_key.asc
gpg: key 0D37AABF: public key "Bob Zesponge <bob@example.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
alice@h0me:~/Documents$ gpg --list-key
/home/alice/.gnupg/pubring.gpg
------------------------------
pub   4096R/1F5C48DC 2014-01-20 [expires: 2014-01-21]
uid                  Alice <alice@example.com>
sub   4096R/6D6A7CBC 2014-01-20 [expires: 2014-01-21]

pub   4096R/0D37AABF 2014-01-20 [expires: 2014-01-21]
uid                  Bob Zesponge <bob@example.com>
sub   4096R/86CAC576 2014-01-20 [expires: 2014-01-21]
```

Signer une clé de son trousseau

```
alice@h0me:~/Documents$ gpg --edit-key bob@example.com
gpg (GnuPG) 1.4.14; Copyright (C) 2013 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.


pub  4096R/0D37AABF  created: 2014-01-20  expires: 2014-01-21  usage: SC
                     trust: unknown       validity: unknown
sub  4096R/86CAC576  created: 2014-01-20  expires: 2014-01-21  usage: E
[ unknown] (1). Bob Zesponge <bob@example.com>

gpg> sign

pub  4096R/0D37AABF  created: 2014-01-20  expires: 2014-01-21  usage: SC
                     trust: unknown       validity: unknown
 Primary key fingerprint: 8569 844D DD35 E98F 0F9E  ADD0 9E86 A4C4 0D37 AABF

     Bob Zesponge <bob@example.com>

This key is due to expire on 2014-01-21.
Are you sure that you want to sign this key with your
key "Alice <alice@example.com>" (1F5C48DC)

Really sign? (y/N) y

You need a passphrase to unlock the secret key for
user: "Alice <alice@example.com>"
4096-bit RSA key, ID 1F5C48DC, created 2014-01-20

gpg: gpg-agent is not available in this session
```

Signer une clé de son trousseau

```
gpg> trust
pub  4096R/0D37AABF  created: 2014-01-20  expires: 2014-01-21  usage: SC
                     trust: unknown       validity: unknown
sub  4096R/86CAC576  created: 2014-01-20  expires: 2014-01-21  usage: E
[ unknown] (1). Bob Zesponge <bob@example.com>

Please decide how far you trust this user to correctly verify other users' keys
(by looking at passports, checking fingerprints from different sources, etc.)

  1 = I don't know or won't say
  2 = I do NOT trust
  3 = I trust marginally
  4 = I trust fully
  5 = I trust ultimately
  m = back to the main menu

Your decision? 5
Do you really want to set this key to ultimate trust? (y/N) y

pub  4096R/0D37AABF  created: 2014-01-20  expires: 2014-01-21  usage: SC
                     trust: ultimate      validity: unknown
sub  4096R/86CAC576  created: 2014-01-20  expires: 2014-01-21  usage: E
[ unknown] (1). Bob Zesponge <bob@example.com>
Please note that the shown key validity is not necessarily correct
unless you restart the program.

gpg> save
```

Exporter et envoyer la clé signée

```
alice@h0me:~/Documents$ gpg --armor --export bob@example.com >  bob_signed.asc

alice@h0me:~/Documents$ gpg --armor --encrypt --sign --output bob_encrypted.asc -r bob@example.com -u alice@example.com \
bob_signed.asc

You need a passphrase to unlock the secret key for
user: "Alice <alice@example.com>"
4096-bit RSA key, ID 1F5C48DC, created 2014-01-20
/You enter your passphrase here/
gpg: gpg-agent is not available in this session
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2014-01-21
```

Importer la clé signée par un tiers

```
bob@h0me:~$ gpg --decrypt bob_encrypted.asc > bob_decrypted.asc

You need a passphrase to unlock the secret key for
user: "Bob Zesponge <bob@example.com>"
4096-bit RSA key, ID 86CAC576, created 2014-01-20 (main key ID 0D37AABF)

gpg: encrypted with 4096-bit RSA key, ID 86CAC576, created 2014-01-20
      "Bob Zesponge <bob@example.com>"
gpg: Signature made Mon 20 Jan 2014 06:10:44 PM CET using RSA key ID 1F5C48DC
gpg: Good signature from "Alice <alice@example.com>"

bob@h0me:~$ gpg --import < bob_decrypted.asc
gpg: key 0D37AABF: "Bob Zesponge <bob@example.com>" 1 new signature
gpg: Total number processed: 1
gpg:         new signatures: 1
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   1  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: depth: 1  valid:   1  signed:   0  trust: 1-, 0q, 0n, 0m, 0f, 0u
gpg: next trustdb check due at 2014-01-21
```

Déchiffre et vérifie la signature du tiers

Importe la clé publique signée par le tiers

Publier sa clé publique

```
bob@h0me:~$ gpg --send-keys --keyserver pgp.mit.edu bob@example.com
```

Exemple de ma clé publique

Search results for '0xe4b7dbc2860d2586'

```
Type bits/keyID     cr. time   exp time   key expir
------------------------------------------------------------------------------

pub  4096R/860D2586 2013-12-24
        Fingerprint=F8F4 BFB3 0137 3588 797B  0022 E4B7 DBC2 860D 2586

uid Sebastien Delcroix (Seb) <seb@sebnet.org>
sig  sig3  860D2586 2013-12-24 __________ 2015-12-23 [selfsig]
sig  sig2  A5BCB3A2 2014-01-17 __________ __________ Kevin Raymond <shaitontm@gmail.com>

sub  4096R/965DB141 2013-12-24
sig sbind  860D2586 2013-12-24 __________ 2015-12-23 []
```

Changer la date d'expiration des clés

```
bob@h0me:~$ date
Tue Jan 21 14:27:11 CET 2014

bob@h0me:~$ gpg --edit-key bob@example.com
gpg (GnuPG) 1.4.14; Copyright (C) 2013 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Secret key is available.

pub  4096R/0D37AABF  created: 2014-01-20  expires: 2014-01-21  usage: SC
                     trust: ultimate      validity: ultimate
sub  4096R/86CAC576  created: 2014-01-20  expires: 2014-01-21  usage: E
[ultimate] (1). Bob Zesponge <bob@example.com>


gpg> expire
Changing expiration time for the primary key.
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) 10
Key expires at ven. 31 janv. 2014 14:22:29 CET
Is this correct? (y/N) y

You need a passphrase to unlock the secret key for
user: "Bob Zesponge <bob@example.com>"
4096-bit RSA key, ID 0D37AABF, created 2014-01-20


pub  4096R/0D37AABF  created: 2014-01-20  expires: 2014-01-31  usage: SC
                     trust: ultimate      validity: ultimate
sub  4096R/86CAC576  created: 2014-01-20  expires: 2014-01-21  usage: E
[ultimate] (1). Bob Zesponge <bob@example.com>
```

Changer la date d'expiration des clés

```
pub  4096R/0D37AABF  created: 2014-01-20  expires: 2014-01-31  usage: SC
                     trust: ultimate      validity: ultimate
sub  4096R/86CAC576  created: 2014-01-20  expires: 2014-01-21  usage: E
[ultimate] (1). Bob Zesponge <bob@example.com>

gpg> key 1

pub  4096R/0D37AABF  created: 2014-01-20  expires: 2014-01-31  usage: SC
                     trust: ultimate      validity: ultimate
sub* 4096R/86CAC576  created: 2014-01-20  expires: 2014-01-21  usage: E
[ultimate] (1). Bob Zesponge <bob@example.com>

gpg> expire
Changing expiration time for a subkey.
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) 10
Key expires at Fri 31 Jan 2014 03:29:56 PM CET
Is this correct? (y/N) y

You need a passphrase to unlock the secret key for
user: "Bob Zesponge <bob@example.com>"
4096-bit RSA key, ID 0D37AABF, created 2014-01-20


pub  4096R/0D37AABF  created: 2014-01-20  expires: 2014-01-31  usage: SC
                     trust: ultimate      validity: ultimate
sub* 4096R/86CAC576  created: 2014-01-20  expires: 2014-01-31  usage: E
[ultimate] (1). Bob Zesponge <bob@example.com>



gpg> save
```

Exemple de clés expirées

```
bob@h0me:~$ gpg --list-key
/home/bob/.gnupg/pubring.gpg
----------------------------
pub   4096R/0D37AABF 2014-01-20 [expires: 2014-01-31]
uid                  Bob Zesponge <bob@example.com>
sub   4096R/86CAC576 2014-01-20 [expires: 2014-01-31]

pub   4096R/1F5C48DC 2014-01-20 [expired: 2014-01-21]
uid                  Alice <alice@example.com>

bob@h0me:~$ gpg --armor --sign --encrypt --output message.asc -u bob@example.com -r alice@example.com message

You need a passphrase to unlock the secret key for
user: "Bob Zesponge <bob@example.com>"
4096-bit RSA key, ID 0D37AABF, created 2014-01-20

gpg: alice@example.com: skipped: unusable public key
gpg: message: sign+encrypt failed: unusable public key
```

Révoquer une clé

- Il faut générer un certificat de révocation pour chacune de vos clés
- Les mettre dans un lieu sûr
  - Si une personne récupère ce certificat de révocation il peut révoquer vos clés
  - Ex : chiffrée dans un autre trousseau de clé comme KeePassX
- Si on n'a pas de certificat de révocation et que l'on perd sa bi-clé on ne pourra plus la révoquer

Révoquer une clé

```
bob@h0me:~$ gpg --list-key
/home/bob/.gnupg/pubring.gpg
----------------------------
pub   4096R/0D37AABF 2014-01-20 [expires: 2014-01-31]
uid                  Bob Zesponge <bob@example.com>
sub   4096R/86CAC576 2014-01-20 [expires: 2014-01-31]

pub   4096R/1F5C48DC 2014-01-20 [expires: 2015-01-20]
uid                  Alice <alice@example.com>
sub   4096R/6D6A7CBC 2014-01-20 [expires: 2015-01-20]

pub   2048R/4F5AEC0C 2014-01-22
uid                  Bob II <bob2@example.com>
sub   2048R/B7899120 2014-01-22

bob@h0me:~$ gpg --gen-revoke bob2@example.com

sec  2048R/4F5AEC0C 2014-01-22 Bob II <bob2@example.com>

Create a revocation certificate for this key? (y/N) y
Please select the reason for the revocation:
  0 = No reason specified
  1 = Key has been compromised
  2 = Key is superseded
  3 = Key is no longer used
  Q = Cancel
(Probably you want to select 1 here)
Your decision? 3
Enter an optional description; end it with an empty line:
> No more comment
>
Reason for revocation: Key is no longer used
No more comment
Is this okay? (y/N) y

You need a passphrase to unlock the secret key for
user: "Bob II <bob2@example.com>"
2048-bit RSA key, ID 4F5AEC0C, created 2014-01-22
```

Révoquer une clé

```
gpg: gpg-agent is not available in this session
ASCII armored output forced.
Revocation certificate created.

Please move it to a medium which you can hide away; if Mallory gets
access to this certificate he can use it to make your key unusable.
It is smart to print this certificate and store it away, just in case
your media become unreadable.  But have some caution:  The print system of
your machine might store the data and make it available to others!
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.14 (GNU/Linux)
Comment: A revocation certificate should follow

iQEuBCABAgAYBQJS34tfER0DTm8gbW9yZSBjb21tZW50AAoJEFCAhtBPWuwMCK0H
/0cdPg3my/OU+OT0p/0xOUlKxYhSls2wd0J3HX78DgoCaj3+T+DtIa3f8YAiacAc
9tuE2jqx5of72bL88yOQ5oayCqsliWOWHI7mSIL9j0il8VGqMboM0/l27AdS4OEF
IGcgj8ToHvpByaZvNQfqpTUwA83tyci4XkYqGpWxkxMlzwXwgGEZ/DcQqXOhcWFZ
8lZAnH3+mZxepS5YLrLGK6e4w4RVv41T+5T9NlkQpmA2VMAJPlhuOItKClGNi/B8
0bDg6n+KAIa5o0q5nQyP3NDbqDyFASmtFJecQa3HP8dsPNh/Fp+qRdDnBKGK4mqe
3s41BR4RXTbvf7O5bjWNmms=
=CzKT
-----END PGP PUBLIC KEY BLOCK-----
```

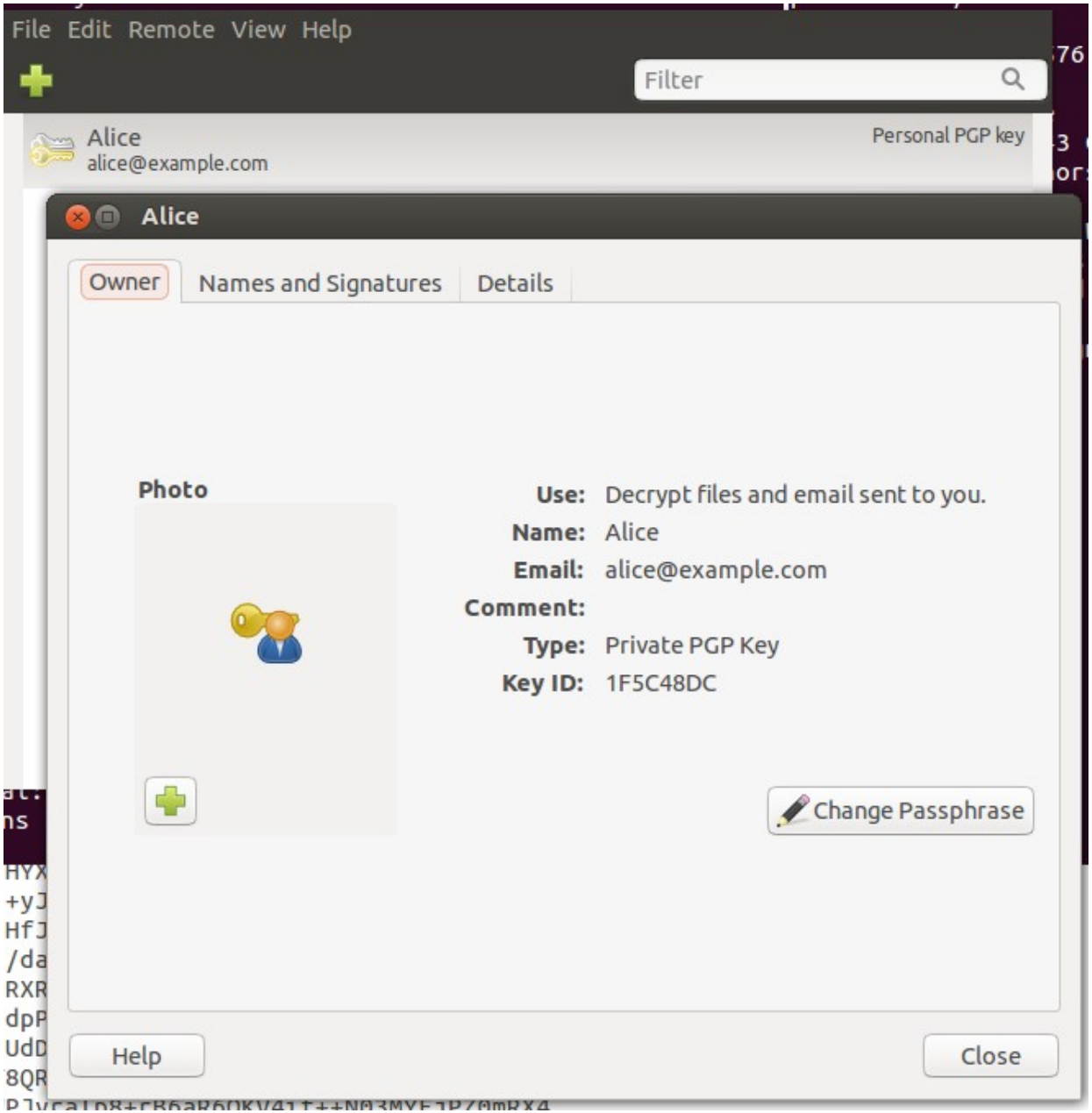
Révoquer une clé

```
alice@h0me:~$ gpg --list-key
/home/alice/.gnupg/pubring.gpg
------------------------------
pub   4096R/1F5C48DC 2014-01-20 [expires: 2015-01-20]
uid                  Alice <alice@example.com>
sub   4096R/6D6A7CBC 2014-01-20 [expires: 2015-01-20]

pub   4096R/0D37AABF 2014-01-20 [expires: 2014-01-31]
uid                  Bob Zesponge <bob@example.com>
sub   4096R/86CAC576 2014-01-20 [expires: 2014-01-31]

pub   2048R/4F5AEC0C 2014-01-22
uid                  Bob II <bob2@example.com>
sub   2048R/B7899120 2014-01-22

alice@h0me:~$ gpg --import bob2_revoke.asc
gpg: key 4F5AEC0C: "Bob II <bob2@example.com>" revocation certificate imported
gpg: Total number processed: 1
gpg:    new key revocations: 1
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   1  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: depth: 1  valid:   1  signed:   0  trust: 0-, 1q, 0n, 0m, 0f, 0u
gpg: next trustdb check due at 2014-01-31
alice@h0me:~$ gpg --list-key
/home/alice/.gnupg/pubring.gpg
------------------------------
pub   4096R/1F5C48DC 2014-01-20 [expires: 2015-01-20]
uid                  Alice <alice@example.com>
sub   4096R/6D6A7CBC 2014-01-20 [expires: 2015-01-20]

pub   4096R/0D37AABF 2014-01-20 [expires: 2014-01-31]
uid                  Bob Zesponge <bob@example.com>
sub   4096R/86CAC576 2014-01-20 [expires: 2014-01-31]

pub   2048R/4F5AEC0C 2014-01-22 [revoked: 2014-01-22]
uid                  Bob II <bob2@example.com>
```

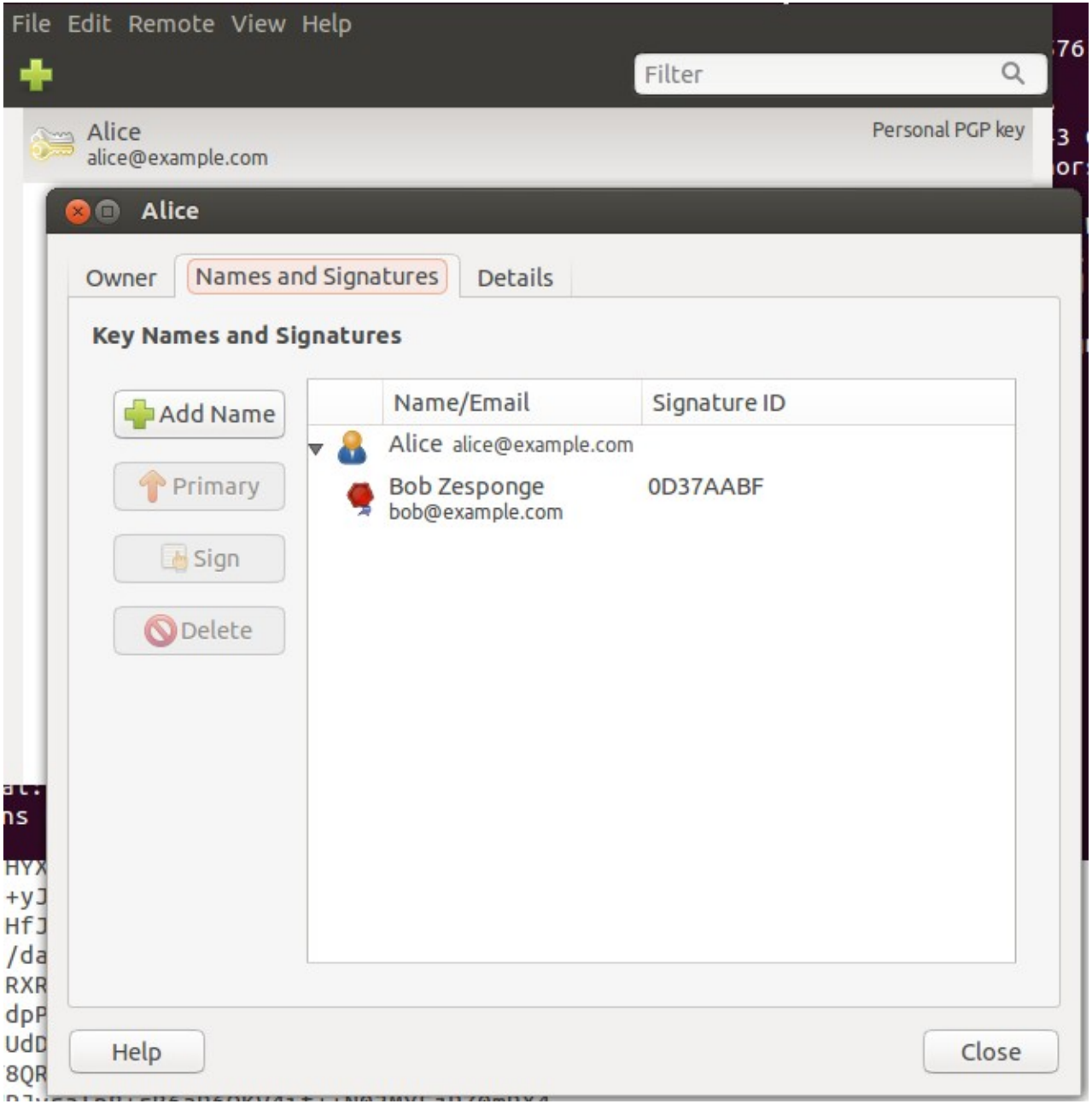
Ajouter une adresse email à une clé

```
bob@h0me:~$ gpg --edit-key bob@example.com
gpg (GnuPG) 1.4.14; Copyright (C) 2013 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Secret key is available.

pub  4096R/0D37AABF  created: 2014-01-20  expires: 2014-01-31  usage: SC
                     trust: ultimate      validity: ultimate
sub  4096R/86CAC576  created: 2014-01-20  expires: 2014-01-31  usage: E
[ultimate] (1). Bob Zesponge <bob@example.com>

gpg> adduid
Real name: Bob Ze Sponge II
Email address: bob2@example.com
Comment:
You selected this USER-ID:
    "Bob Ze Sponge II <bob2@example.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O

You need a passphrase to unlock the secret key for
user: "Bob Zesponge <bob@example.com>"
4096-bit RSA key, ID 0D37AABF, created 2014-01-20

gpg: gpg-agent is not available in this session

pub  4096R/0D37AABF  created: 2014-01-20  expires: 2014-01-31  usage: SC
                     trust: ultimate      validity: ultimate
sub  4096R/86CAC576  created: 2014-01-20  expires: 2014-01-31  usage: E
[ultimate] (1)  Bob Zesponge <bob@example.com>
[ unknown] (2). Bob Ze Sponge II <bob2@example.com>

gpg> save
```

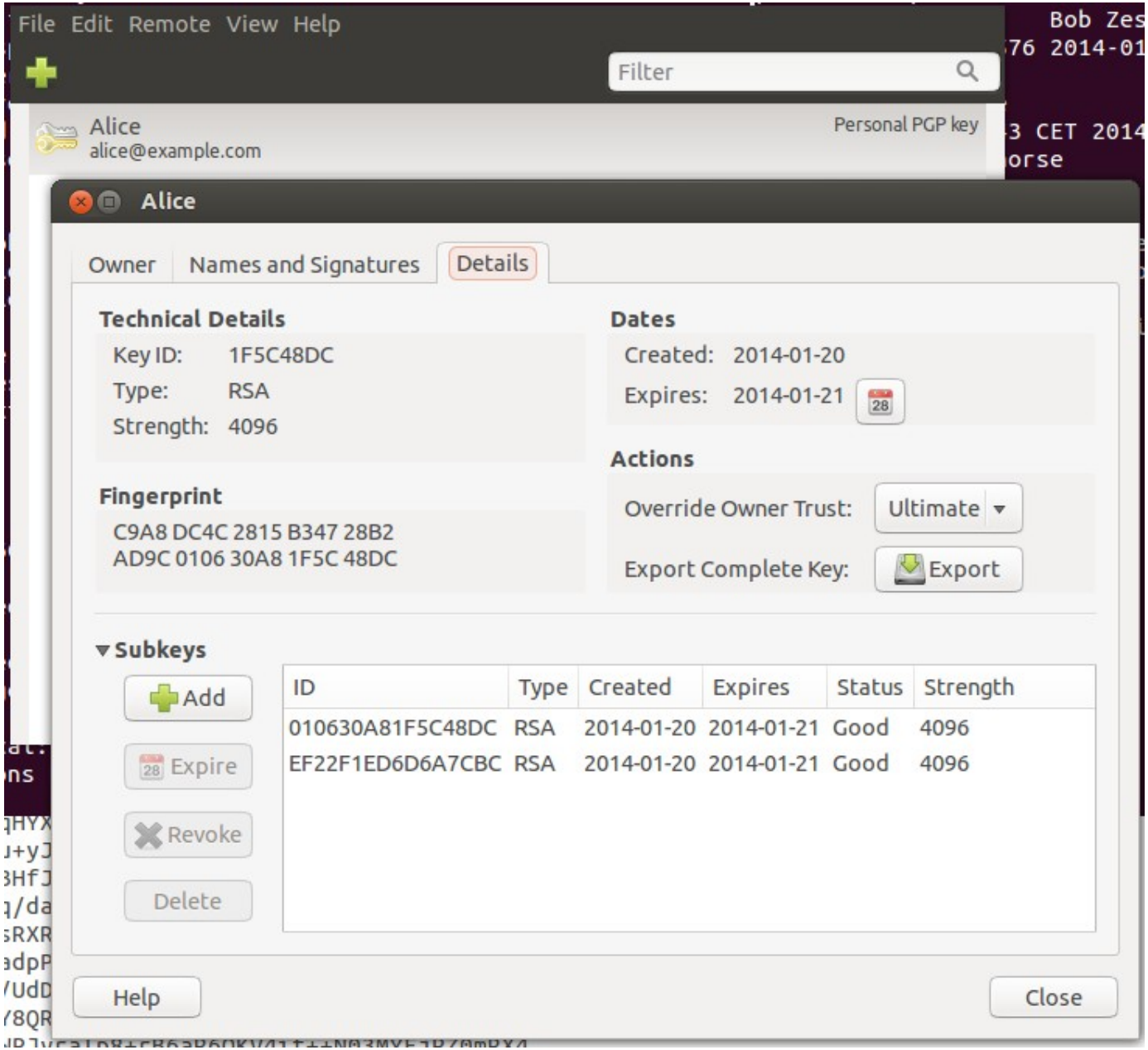
Trousseau de clés « graphique »



Trousseau de clés « graphique »



Trousseau de clés « graphique »



Chiffrer et signer des messages

Chiffrer/Déchiffrer un message

```
bob@h0me:~$ cat message
Ceci est un message confidentiel !!!

bob@h0me:~$ gpg --armor --encrypt --output message.asc -r alice@example.com message

bob@h0me:~$ cat message.asc
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.14 (GNU/Linux)

hQIMA+8i8e1tany8AQ//YBym1XhKnIt3jsSTpUvYRSCb8XA2GkUg9+mwSufv0G0/
rFN8aRM/OWWiMbZZ3M9fOy9A9qjUjFy+9EYe6Q2x6a1e1S4kXNFCnPDIoTUEBDqw
cSuIlRaa8k85N6SxhaKS+g4tz3psjYBxkLnJXCmd1mzNVUiAIb0zVT2rkhY+ssIi
1L4n+qpLNNrazfAnt8xH6K0FJ9g3BgkBtEnNT1OR/v/qnO3wgldod1sQJGVO60Ih
I7eeG0LhvFAW37at/zomy+iDPOYBtL2WoTc1E12H4nawtK5TzEBjaFNz6NgRM57T
7GKl9aYORvoZv4Jw9G0IGUqwV9OK1/39pkUEvKugnsgwjjU/6n/AOO00j1btUJBJ
lm3njPM7Btgr64Lai1w1Tkyg1QLRaW9YfCikkyY/TDXEwIjO6Cdjkrpp60Gh21eW
cBUIfftLeGkeG1HXLPv8FIG4MYNtizjVJhTDqyPXr1+Nn4CPQvbCa3oWTUAD+iYk
q2fFQEBT+M8Hc0USSGXmlXyAigFmowTy6yd2TyNAmHIwQh+nyyOK+p9MkYkNLlOm
wE8GZd+Z6TTYpmuCBSQUYBAXdZKkcgEnmi6s4Mfx4H+dBP2zHjnnR3QSwVZDrlbM
9w50qeiFSwvUXUgE6PAo9DkCPnkaVj4489J/vv5HQ0qXhWrH/TL8liK4rwZODhjS
YgF8t7F1sAwtzsoud5A1XgzlOyo/Q3KAng+bCgpp360K7RQx+rtkk6KXDwBryJis
+/i1yzBRE23rMfWjYpT4lGHRRtqFvcV8ICw4egLH/u5cf9z3CYQL+lABhfvuJ/uA
yAvS
=DJcW
-----END PGP MESSAGE-----

######################################################################

alice@h0me:~$ gpg --decrypt message.asc > message

You need a passphrase to unlock the secret key for
user: "Alice <alice@example.com>"
4096-bit RSA key, ID 6D6A7CBC, created 2014-01-20 (main key ID 1F5C48DC)

gpg: gpg-agent is not available in this session
gpg: encrypted with 4096-bit RSA key, ID 6D6A7CBC, created 2014-01-20
      "Alice <alice@example.com>"
alice@h0me:~$ cat message
Ceci est un message confidentiel !!!
```

Message non signé => /!\ N'importe qui aurait pu envoyer ce message

Signer/Vérifier une signature d'un message

```
bob@h0me:~$ gpg --armor –detach-sign --output message_sign.asc -u bob@example.com message

You need a passphrase to unlock the secret key for
user: "Bob Zesponge <bob@example.com>"
4096-bit RSA key, ID 0D37AABF, created 2014-01-20


########################################################################

alice@h0me:~$ gpg --verify message_sign.asc message
gpg: Signature made Tue 21 Jan 2014 04:06:36 PM CET using RSA key ID 0D37AABF
gpg: Good signature from "Bob Zesponge <bob@example.com>
```

Signature et message détachés => Nécessite de Transmettre le message et la signature dans deux fichiers séparés

Message bien signé Par Bob

Chiffrer ET Signer un message

```
bob@h0me:~$ gpg --armor --sign --encrypt --output message.asc -u bob@example.com -r alice@example.com message

You need a passphrase to unlock the secret key for
user: "Bob Zesponge <bob@example.com>"
4096-bit RSA key, ID 0D37AABF, created 2014-01-20

##########################################################################

alice@h0me:~$ gpg –decrypt –output message message.asc

You need a passphrase to unlock the secret key for
user: "Alice <alice@example.com>"
4096-bit RSA key, ID 6D6A7CBC, created 2014-01-20 (main key ID 1F5C48DC)

gpg: gpg-agent is not available in this session
gpg: encrypted with 4096-bit RSA key, ID 6D6A7CBC, created 2014-01-20
      "Alice <alice@example.com>"
Ceci est un message confidentiel !!!
gpg: Signature made Tue 21 Jan 2014 04:41:30 PM CET using RSA key ID 0D37AABF
gpg: Good signature from "Bob Zesponge <bob@example.com>"

alice@h0me:~$ cat message
Ceci est un message confidentiel !!!
```

On précise pour qui chiffre et qui signe

Message bien signé Par Bob

Signer une arborescence de fichiers

```
bob@h0me:~$ ls src
loworld.pl  loworld.rb  loworld.sh
bob@h0me:~$ cat src/*
#!/usr/bin/perl -w

use strict ;

print "Hello World!\n"
#!/usr/bin/env ruby

puts "Hello world!"
#!/bin/bash

echo "Hello world!"

bob@h0me:~$ find src/* -exec sha256sum {} \; > src-sha256
bob@h0me:~$ cat src-sha256
14eee2e36e51f777c8bc0633ed487cf5b3c25b1188a2793d161528eaac797927  src/loworld.pl
7e7417742a1736b1783e1f6aa4ee99cf6ad5626fe20d6601e720ed84445ad95b  src/loworld.rb
7bb8d0981a5d2deb513b70390b9ee32327002c5c4864ab0564f847a5d2e67133  src/loworld.sh

bob@h0me:~$ gpg --armor --detach-sign --output src-sha256.sign -u bob@example.com src-sha256

You need a passphrase to unlock the secret key for
user: "Bob Zesponge <bob@example.com>"
4096-bit RSA key, ID 0D37AABF, created 2014-01-20


bob@h0me:~$ tar cvf src.tar src src-*src/
src/loworld.rb
src/loworld.pl
src/loworld.sh
src-sha256
src-sha256.sign
```

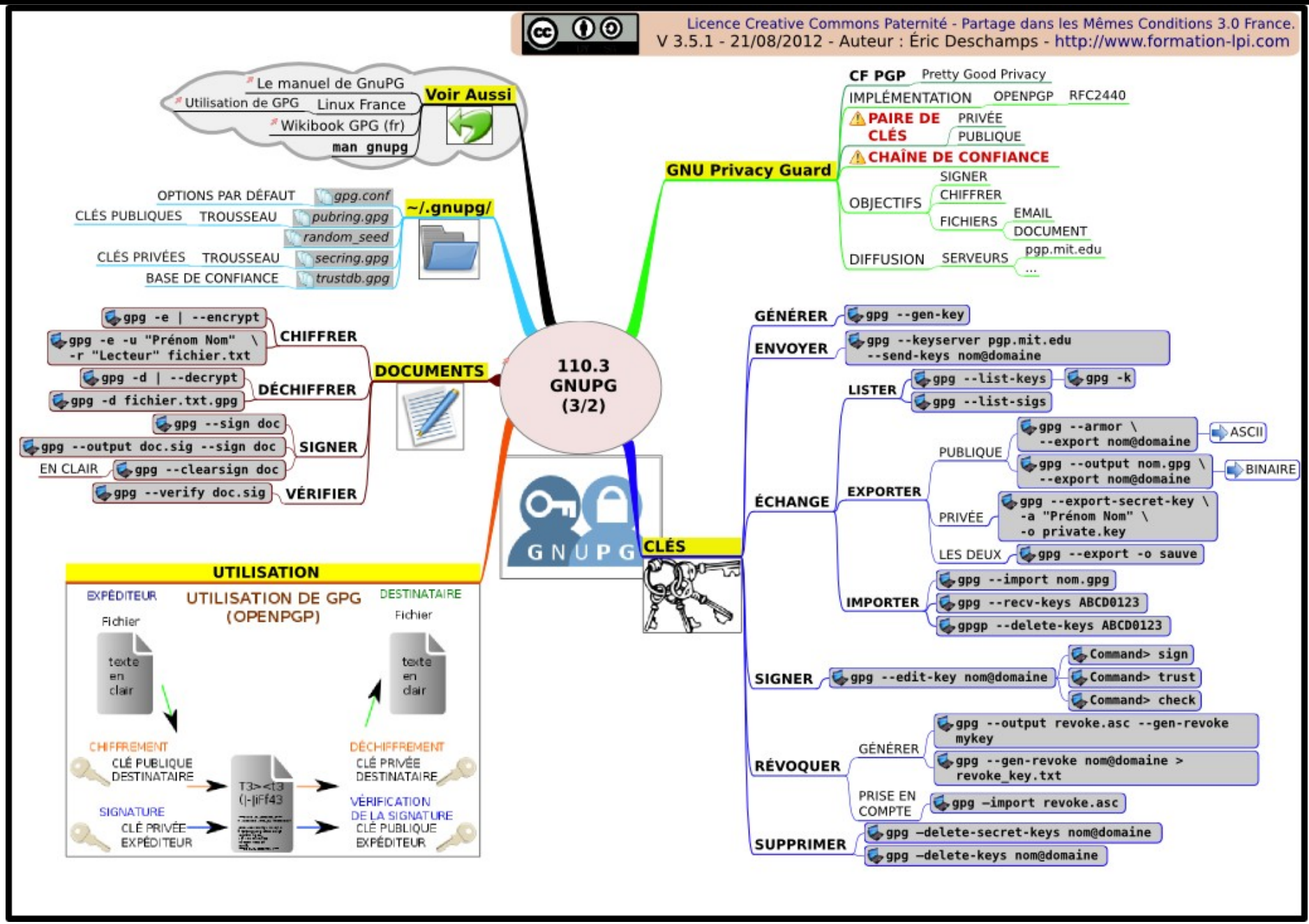
Vérifier une arborescence de fichiers

```
alice@h0me:~$ tar xvf src.tar
src/
src/loworld.rb
src/loworld.pl
src/loworld.sh
src-sha256
src-sha256.sign
alice@h0me:~$ mv src-sha256 src-sha256.orig
alice@h0me:~$ find src/* -exec sha256sum {} \; > src-sha256
alice@h0me:~$ diff src-sha256 src-sha256.orig
alice@h0me:~$ echo $?
0
alice@h0me:~$ gpg --verify src-sha256.sign src-sha256
gpg: Signature made Tue 21 Jan 2014 05:38:22 PM CET using RSA key ID 0D37AABF
gpg: Good signature from "Bob Zesponge <bob@example.com>"
```

Chiffrement symétrique

```
bob@h0me:~$ gpg -c message
bob@h0me:~$ ls -la message message.gpg
-rw-r--r-- 1 bob bob 37 Jan 21 15:55 message
-rw-rw-r-- 1 bob bob 78 Jan 22 14:57 message.gpg
bob@h0me:~$ sha256sum message
42cfa9fecfde7a610f30305df52bcd60cc281f8164a484824f41c762e6e13800  message

################################################################

alice@h0me:~$ gpg --output message.ok --decrypt message.gpg
gpg: CAST5 encrypted data
gpg: gpg-agent is not available in this session
gpg: encrypted with 1 passphrase
gpg: WARNING: message was not integrity protected
alice@h0me:~$ sha256sum message.ok
42cfa9fecfde7a610f30305df52bcd60cc281f8164a484824f41c762e6e13800  message.ok
```

Résumé



Questions ?