Dimecres 23/02/22:

MP11UF1-NF1 A05 Seguretat GnuPG

Pràctica 1: XIFRAR

- generar keys i xifrar de marta a pere

crear usuaris pere, marta, anna

pere es crea les seves claus i les exporta

```
gpg --full-generate-key
gpg --gen-key
gpg --list-keys
gpg --output pere.gpg --export pere@edt.cat
cat pere.gpg
gpg --armor --export pere@edt.cat
gpg --armor --output  pere.gpg.pem --export pere@edt.cat
cat pere.gpg.pem

 file pere.gpg pere.gpg.pem

pere.gpg:      GPG key public ring, created Wed Feb 13 08:41:24
2019
pere.gpg.pem: PGP public key block Public-Key (old)
```

marta importa la clau de pere i la signa (validant-la)

```
[marta] gpg--gen-keys
[marta] gpg --import
/tmp/pere.gpg
[marta] gpg --list-keys
[marta] gpg --edit-key 8BA06141
[marta]     gpg> fpr
[pere] gpg --fingerprint
[marta]     gpg> sign
[marta]     gpg> check
[marta]     gpg> quit
[marta] gpg --edit-key 8BA06141
```

- Observar el validity un cop signada la clau.
- Observar que el procés és que marta signa amb la seva clau privada la clau pública de pere per donar-li validesa.

marta encripta un document que només pere pot veure

```
[marta] gpg --output passwd.gpg --encrypt --recipient pere@edt.cat passwd.txt
[marta] cat passwd.gpg
[marta] gpg --armor --output passwd.gpg.pem --encrypt --recipient pere@edt.cat passwd.txt
[marta] cat passwd.gpg.pem
[marta]mv passwd.gpg* /tmp
[marta]  file /tmp/passwd.gpg*

/tmp/passwd.gpg:      PGP RSA encrypted session key - keyid: E819AEF6 8A547F27 RSA (Encrypt or Sign) 2048b .
/tmp/passwd.gpg.pem: PGP message Public-Key Encrypted Session Key (old)
```

- l'emisor encripta amb la clau pública del destinatari.
- es pot encriptar generant binari o pem (o stdout).
- només destinatari (cal indicar-ho a recipient) el pot desencriptar.
- [pendent enviar un missatge xifrat a més d'un destinatari]

```
pere desencriptar el missatge
[pere] gpg --decrypt /tmp/passwd.gpg
[pere] gpg --output passwd.txt  --decrypt
/tmp/passwd.gp
[pere] cat passwd.txt
[pere] gpg --decrypt /tmp/passwd.gpg.pem
```

Pràctica2: SIGNAR

- pere signa un document i marta valida

```
pere signa un missatge
            [pere] gpg -sign group.txt
            [pere] cat group.txt.gpg
            [pere] gpg --out group.gpg --sign group.txt
            [pere] cat group.gpg
            [pere] file group.gpg
            [pere] gpg --armor --out group.gpg.pem --
            sign group.txt
            [pere] cat group.gpg.pem
```

- cal signar el missatge amb la clau privada de l'emissor. Qualsevol pot verificar la signatura si disposa de la clau pública del emisor.
- es pot signar tot generant binari o pem, o fer un clearsign o fer un detach.

- en un clearsign hi ha el missatge en text pla seguit de la signatura.
- en un detach només es genera el fitxer de signatura. Cal enviar el fitxer de dades i el de la signatura.
- el receptar pot extreure el missatge (si tot embolcallat) i verificar la signatura. O només verificar la signatura.
- observar si diu Good signature o no i amb quines condicions.

marta verifica la signatura del document de pere

```
[marta] gpg --decrypt /tmp/group.gpg
[marta] gpg --output group.txt --decrypt
/tmp/group.gpg.pem
[marta] cat group.txt
[marta] gpg --output group.txt --verify
/tmp/group.gpg.pem
```

pere signa amb clear text (clear text + signature pem)

```
[pere] gpg --output group.gpg --clear-
sign group.txt
[marta]  gpg --decrypt /tmp/group.gpg
[marta] gpg --verify /tmp/group.gpg
```

pere fa un detach signature

```
[pere] gpg --output group.gpg --detach-
sign group.txt
[marta]  gpg --verify /tmp/group.pem
/tmp/group.txt
```

- es verifica la signatura usant el fitxer de dades i el de signatura.
- [aquest format és el que tenen els fitxer de signatura de software: https://archives.]

Pràctrica 3: Managing Keys

pere es crea un altre joc de claus amb una altra identitat

```
[pere/perico] gpg  --gen-
key
[pere/perico] gpg  --list-
keys
[pere]  gpg --edit-key perico
          gpg> list
          gpg> togggle
          gpg> check
```

- observar pub (public) mida, tipus (dsa, rdsa, g elgamal), id. Validity, trust. Toggle per observar les sec (provades).
- observar que la de pere és una self-signature.
- trust levels: *unknown, none, marginal, full*
- no és el mateix *validate* que *trust*.

marta modifica el trust de la clau de pere

```
[marta] gpg --edit-key pere (validate: full, trust:
unknown)
[marta]    gpg> trust  (establir el 3)
```

Regla per validar una clau: a través de signar-la o de el trust:

The web of trust allows a more elaborate algorithm to be used to validate a key. Formerly, a key was considered valid only if you signed it personally. A more flexible algorithm can now be used: a key K is considered valid if it meets two conditions:

1. it is signed by enough valid keys, meaning you have signed it personally,

- it has been signed by one fully trusted key, or
- it has been signed by three marginally trusted keys; and

2. the path of signed keys leading from K back to your own key is five steps or shorter.

The path length, number of marginally trusted keys required, and number of fully trusted keys required may be adjusted. The numbers given above are the default values used by GnuPG.

anna importarà el clauer de marta:

```
[anna] gpg --gen-key
[marta] gpg --armor --output /tmp/marta.pem --export
marta
[marta] cat /tmp/marta.pem
[anna] gpg --import /tmp/marta.pem
[anna] gpg --list-keys
[anna] gpg --edit-key marta@edt.cat
```

- marta ha exportat una única clau del seu clauer (la de marta@edt.cat) i anna ha importat només aquesta (no la signa encara). validity: unknown, trust: unknown.

```
[marta] gpg --armor --output /tmp/marta.pem --
export
[anna] gpg --imprt /tmp/marta.pem
[anna] gpg --list-keys
[anna] gpg --edit-key marta@edt.cat
[anna] gpg --edit-key pere@edt.cat
```

- ara anna té importades les claus de marta i pere, les dues amb unknown de validity i trust.

anna verifica un document signat per pere i un signat per marta

```
[pere ] gpg --armor --output /tmp/group.pem  --clearsign
group.txt
[anna] gpg --decrypt /tmp/group.pem
```

- Obtenim *Good* signature però *Warning* no tenim prova de que pere sigui qui diu ser.

```
[marta]  gpg --armor --output /tmp/passwd.pem --clearsign
passwd.txt
[anna]    gpg --decrypt /tmp/passwd.pem
```

- Obtenim *Good* signature però *Warning* no tenim prova de que marta sigui qui diu ser.

anna signa la clau de marta

```
[anna] gpg --edit-key marta
[anna]      gpg> fpr
[anna]      gpg> sign
[anna]      gpg> chck
[anna] gpg --edit-key marta
[anna]      gpg> chck
[anna] gpg --decrypt
/tmp/passwd.pem
```

- Ara anna en verificar la signatura de marta obtenim un *Good* signature.

```
[anna] gpg --decrypt
/tmp/group.pem
```

- en verificar la signatura de pere continuem obtenint un *Good* signature però un warning de que no sabem qui és pere.

```
[anna] gpg --decrypt
/tmp/group.pem
```

anna posa a trust la clau de marta. Observar que el validity de pere ara ja és full. Ara quan anna valida la firma de pere ja diu Good Signature i autentica que l'usuari és qui diu ser.

- anna edita la clau de marta i activa el trust:ultimate
- el llistar ara la clau de marta observem el trust ultimate.

```
[anna] $ gpg --edit-key marta
[anna] gpg> trust
[anna ]$ gpg --edit-key marta
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   2  signed:   1  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: depth: 1  valid:   1  signed:   0  trust: 1-, 0q, 0n, 0m, 0f, 0u
pub  2048R/2A7CEDBD  created: 2021-01-20  expires: never       usage: SC
                     trust: ultimate      validity: ultimate
sub  2048R/1C6B806F  created: 2021-01-20  expires: never       usage: E
[ultimate] (1). marta mas moles (la marta) <marta@edt.org>
```

- en llistar la clau de pere observer que el validity ha canviat passant de unknown a full. Això és perquè ha recalculat els validity en fer el canvi del trust de marta.

```
[anna] $ gpg --edit-key pere
gpg (GnuPG) 1.4.23; Copyright (C) 2015 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  2048R/C899CCBF  created: 2021-01-20  expires: never       usage: SC
```

```
                    trust: unknown       validity: full
sub  2048R/C3A91E7D  created: 2021-01-20  expires: never       usage: E
[  full  ] (1). pere pou prat (lo pere) <pere@edt.org>
```

- ara si anna torna a validar el missatge signat per pere ja no es mostra el *warning* la signatura és correcte: *Good Signature*.

pere signa amb una de varies claus que un usuari té

- l'usuari pere signa amb una altra de les seves claus, en aquest cas la de perico

```
[pere] $ gpg --output /tmp/group.gpg --local-user perico --sign /etc/group
[anna] $ gpg --verify /tmp/group.gpg
gpg: Signature made Tue 26 Jan 2021 09:34:03 AM CET using RSA key ID 52F30FF3
gpg: Good signature from "perico pebrots pardals (lo perico) <perico@edt.org>"
```

pere fa el mateix usant Clearsign

- pere signa amb clearsign i encripta tot de cop.
- anna desencripta i li solicita el fitxer de dades amb el que ha de verificar la signatura. Aquest és un exemple estúpid perquè l'únic que s'encripta és la firma!.
- el contingut del fitxer de dades no forma part del missatge, el destinatari el rep a part.

```
[pere] $ gpg -a --output /tmp/final.gpg --recipient anna --detach-sign --encrypt /tmp/passwd.txt
[anna] $ gpg --output /tmp/decript.gpg --decrypt  /tmp/final.gpg
You need a passphrase to unlock the secret key for
user: "anna andreu aregall (la anna) <anna@edt.org>"
2048-bit RSA key, ID A2AA4D2C, created 2021-01-26 (main key ID 57BFB02E)
gpg: encrypted with 2048-bit RSA key, ID A2AA4D2C, created 2021-01-26
      "anna andreu aregall (la anna) <anna@edt.org>"
Detached signature.
Please enter name of data file: /tmp/passwd.txt
gpg: Signature made Tue 26 Jan 2021 12:44:11 PM CET using RSA key ID C899CCBF
gpg: Good signature from "pere pou prat (lo pere) <pere@edt.org>"
```