

The GNU Privacy Guard. Terminalkomandoer til GnuPG

Kryptering og signering av dokumenter med GnuPG

Installere GnuPG (macOS og Windows): <https://gnupg.org/>

Installere GnuPG (Linux: Debian, Ubuntu og Mint)

- `sudo apt install gpg`

Lage GPG nøkkel

- `gpg --full-generate-key`

Kontrollere nøkler

- `gpg --list-secret-keys`
- `gpg --list-public-keys`

Eksportere offentlig nøkkel

- `gpg --output key.gpg --export id (id fra nøkkel) (bytt key med et passende navn på nøkkelen)`

Importere offentlig nøkkel

- `gpg --import key.gpg (bytt key med navnet på nøkkelen)`
- `gpg --import key.asc (bytt key med navnet på nøkkelen)`

Eksportere nøkler til fil

- `gpg --export-secret-keys --armor (id fra nøkkel) > my_secret_key.asc`
- `gpg --export --armor (id fra nøkkel) > my_public_key.asc`

Importere nøkkel fra fil

- `gpg --import key.asc (bytt key med navnet på nøkkelen)`

Verifisere nøkkel

- På den andre brukerens maskin:
 - `gpg --list-public-keys`
 - `gpg --edit-key id (bytt id med den lange remsen med tall og tekst)`
 - `gpg> fpr (viser fingeravtrykket til nøkkelen, skriv ut denne)`
- På din maskin:
 - `gpg --list-public-keys`
 - `gpg --edit-key id (bytt id med den lange remsen med tall og tekst fra den importerte nøkkelen)`
 - `gpg> fpr (viser fingeravtrykket til nøkkelen, sammenlikne med den fra den andre maskinen) Hvis de er like, signer.`
 - `gpg> sign (Vil du virkelig signere? velg «j»)`

Slette offisiell nøkkel

- `gpg --delete-key id (bytt id med den lange remsen med tall og tekst fra nøkkelen)`

Slette privat nøkkel

- `gpg --delete-secret-key id (bytt id med den lange remsen med tall og tekst fra nøkkelen)`

The GNU Privacy Guard. Terminalkomandoer til GnuPG

Kryptere med passord

I dette tilfellet et txt-dokument som heter testbrev.

- `gpg -c testbrev.txt` (krypterer dokumentet med passord. Det krypterte dokumentet får en ekstra filendelse (.gpg)).
- `gpg -d testbrev.txt.gpg` (dekrypterer dokumentet, og viser innholdet i terminalen)
- `gpg -d testbrev.txt.gpg>testbrev.txt` (dekrypterer og gjenoppretter / lager et nytt dokument)

Hvis du sender den krypterte filen til noen andre, eller starter maskinen på nytt, vil GnuPG spørre om passord når du dekrypterer.

Kryptere med krypteringsnøkkel

I dette tilfellet et txt-dokument som heter testbrev.

- `gpg --encrypt --recipient [email nøkkel] testbrev.txt` (krypterer med nøkkel. Det krypterte dokumentet får en ekstra filendelse (.gpg)).
- `gpg --decrypt --output testbrev.txt testbrev.txt.gpg` (dekrypterer fil som er kryptert med nøkkel)

The GNU Privacy Guard. Terminalkomandoer til GnuPG

Signere med GnuPG

I dette tilfellet et txt-dokument som heter testbrev.

Du signerer en fil med din private nøkkel:

- `gpg --sign testbrev.txt` (Dokumentet blir signert og pakket, og får en ekstra filendelse (.gpg)).
- `gpg --clearsign testbrev.txt` (Legger en ASCII signatur rundt innholdet i filen. Filen får filendelsen (.asc). Egner seg kun for txt-filer).
- `gpg --detach-sig testbrev.txt` (lager en egen signaturfil, som er koblet til det originale dokumentet. Det originale dokumentet er urørt. Signaturfilen får filendelsen (.sig). Du må ha både originalfilen og signaturfilen for å kunne verifisere den digitale signaturen).

Verifisere en signert fil

I dette tilfellet et txt-dokument som heter testbrev.

For å kunne verifisere en signert fil, må du ha den offentlige nøkkelen fra den som signerte filen.

- `gpg --verify testbrev.txt.gpg` (verifiserer den digitale signaturen, og viser resultatet i terminalen). For å pakke ut/gjenopprette dokumentet, bruk «`gpg --decrypt --output testbrev.txt testbrev.txt.gpg`» på vanlig måte (resultatet av verifiseringen av signaturen vises i terminalen)
- `gpg --verify testbrev.txt.asc` (verifiserer den digitale signaturen, og viser resultatet i terminalen)
- `gpg --verify testbrev.txt.sig testbrev.txt` (verifiserer den digitale signaturen, og viser resultatet i terminalen)

Du kan både kryptere og signere et dokument ved å bruke: `--encrypt` og `--sign`.