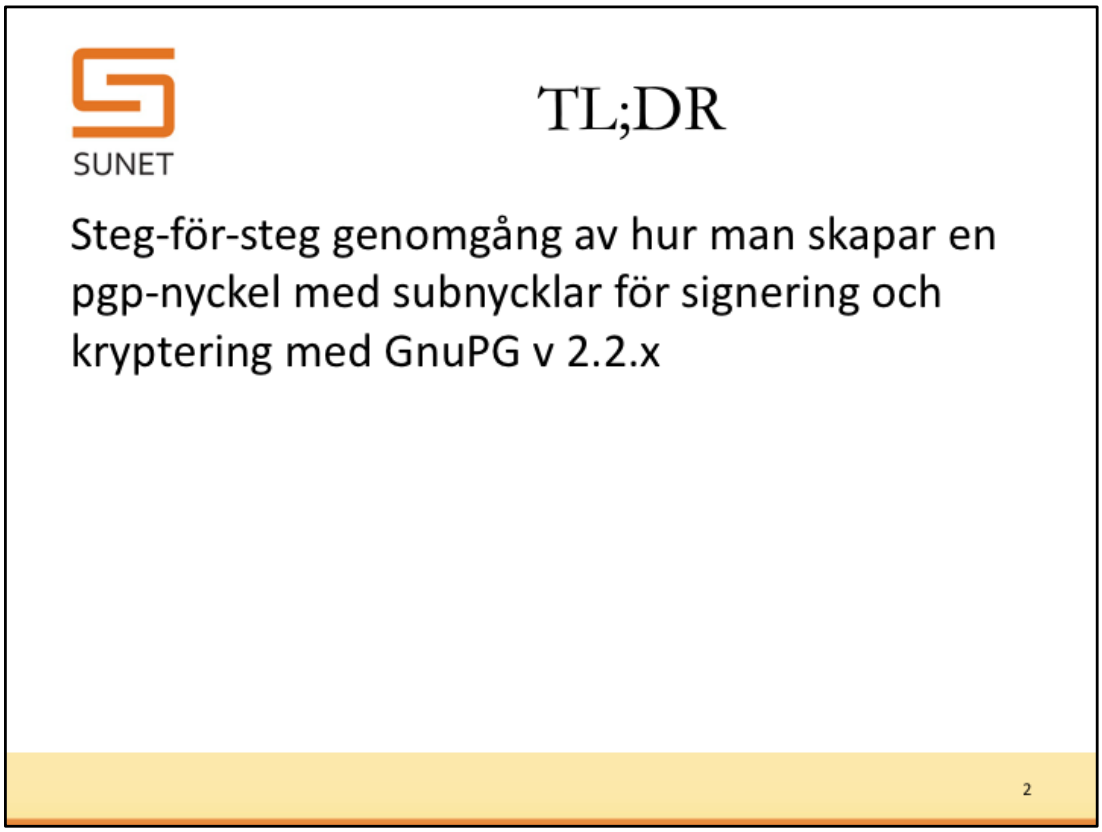
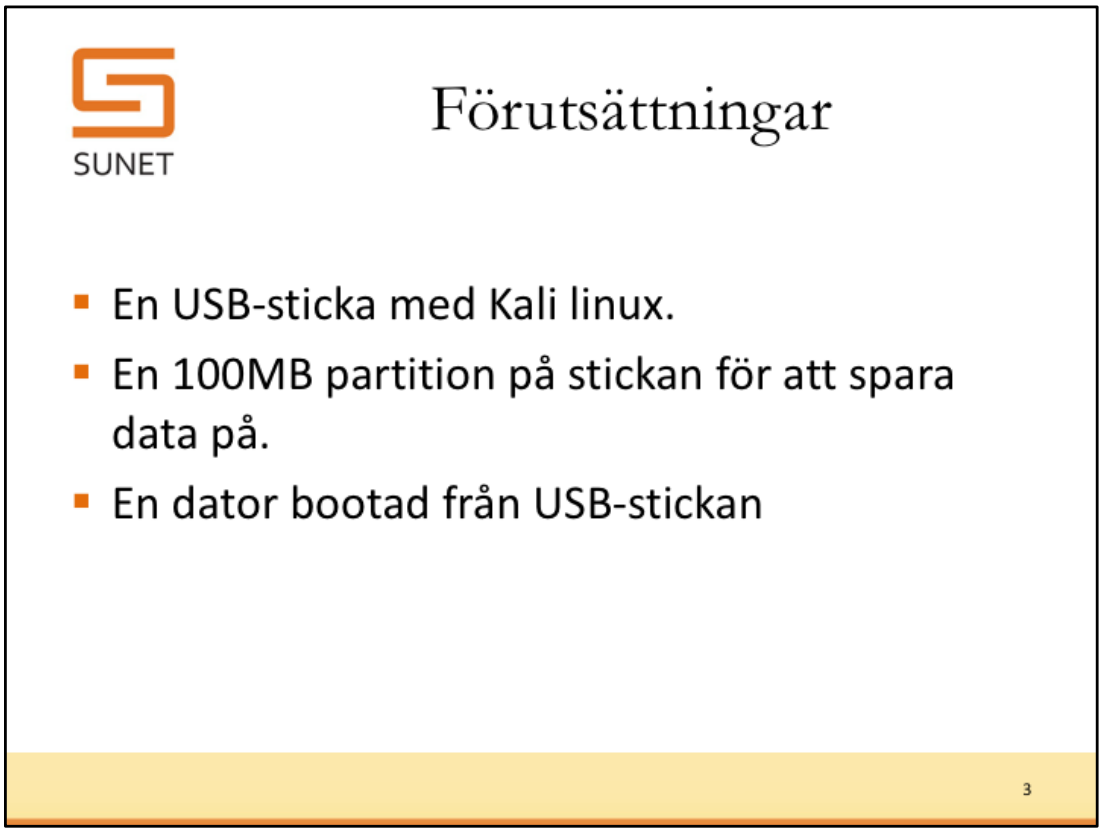

SUNET
Workshop
PGP teamnycklar


SUNET
TL;DR
Steg-för-steg genomgång av hur man skapar en pgp-nyckel med subnycklar för signering och kryptering med GnuPG v 2.2.x
2



https://cdimage.kali.org/kali-2019.1a/kali-linux-light-2019.1a-amd64.iso
https://docs.kali.org/downloading/kali-linux-live-usb-install

SUNET

Det bör dock gå bra att följa merparten av instruktionerna förutsatt att du har GPG v2.2.x på någon linux eller mac-distribution med Bash som skal. Till Workshopen finns en labmaskin

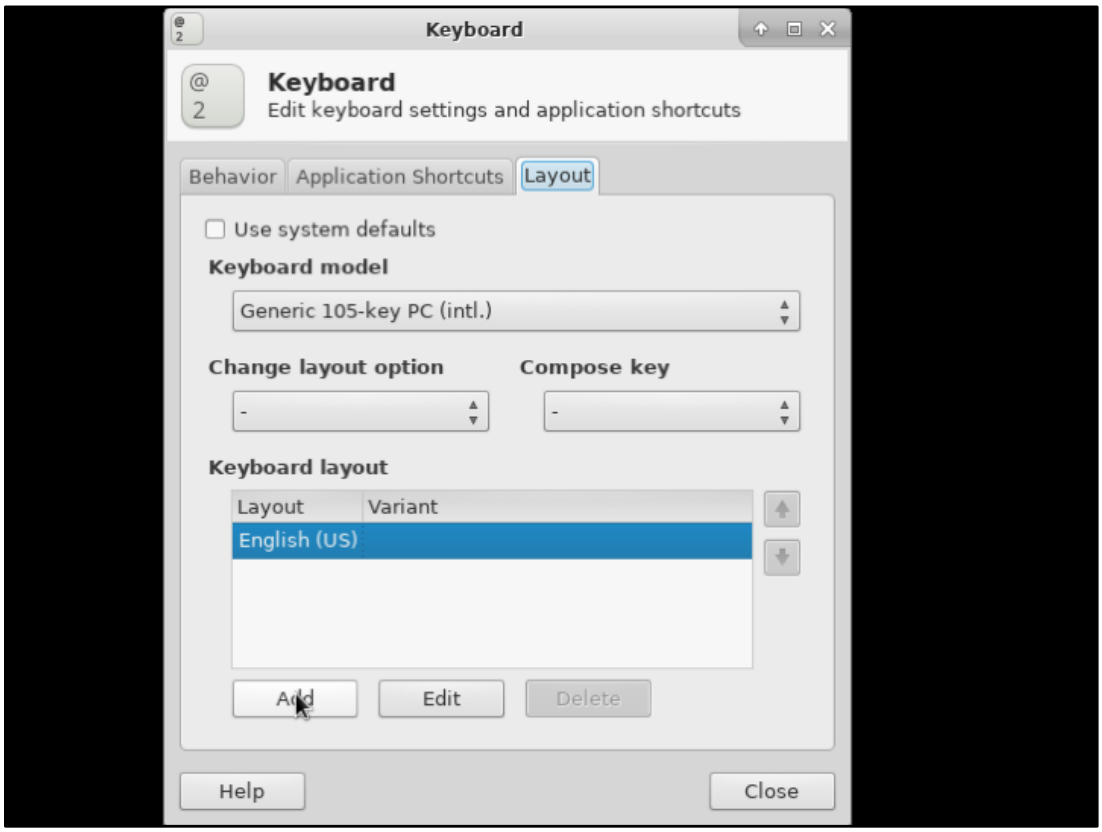
- lab.cert.sunet.se

Konton och lösenord finns på

- https://www.cert.sunet.se/ws/accounts

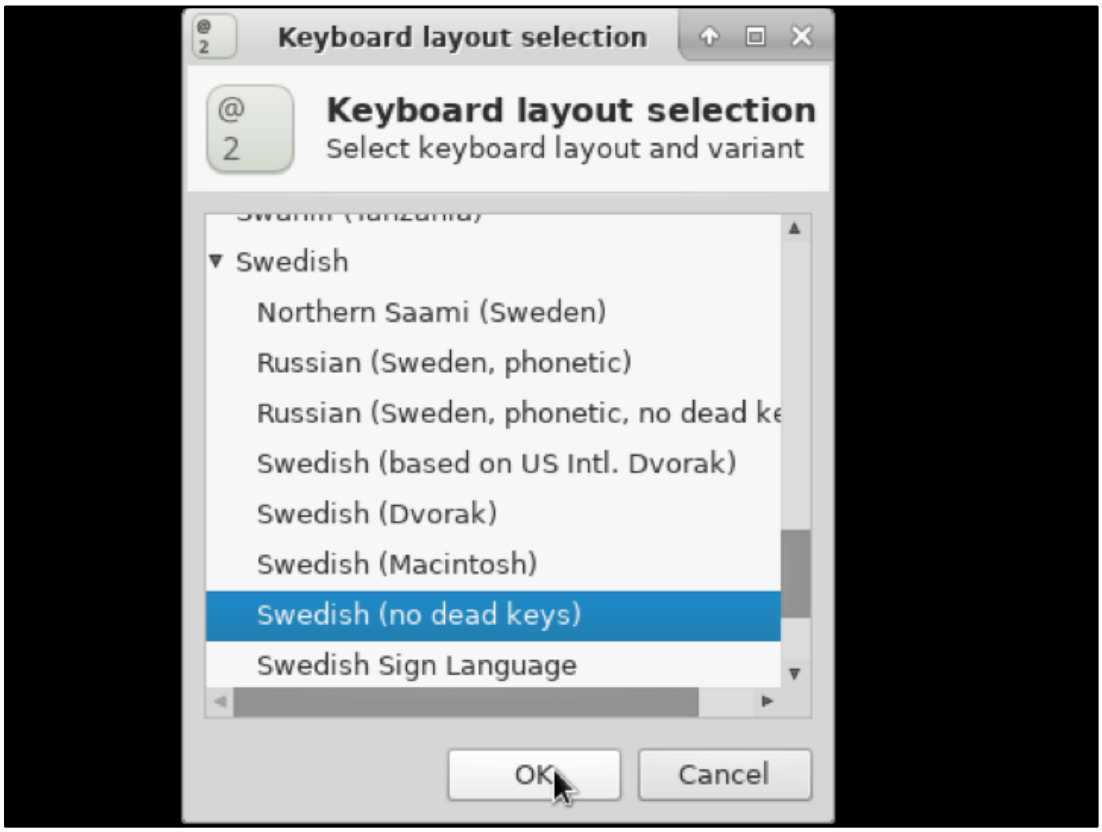
Om du vill byta tangenbordslayout

Applications -> Settings -> Keyboard

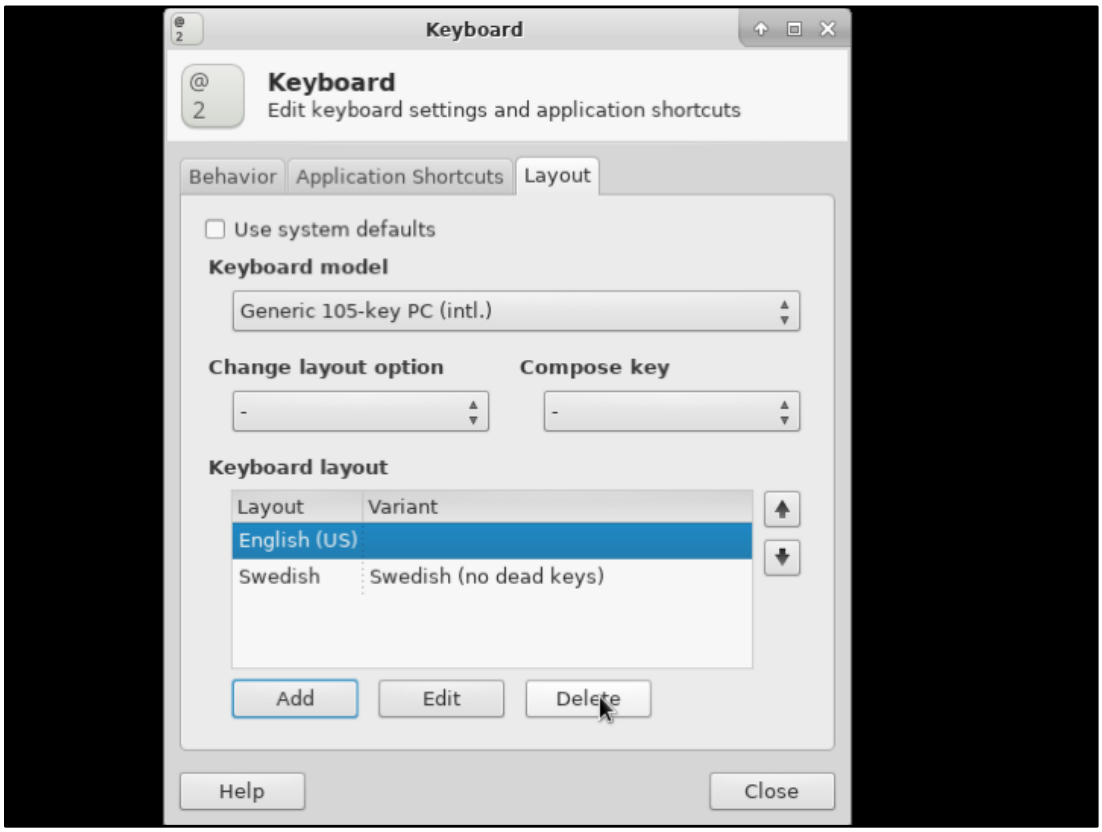


Välj fliken layout

Klicka på Add

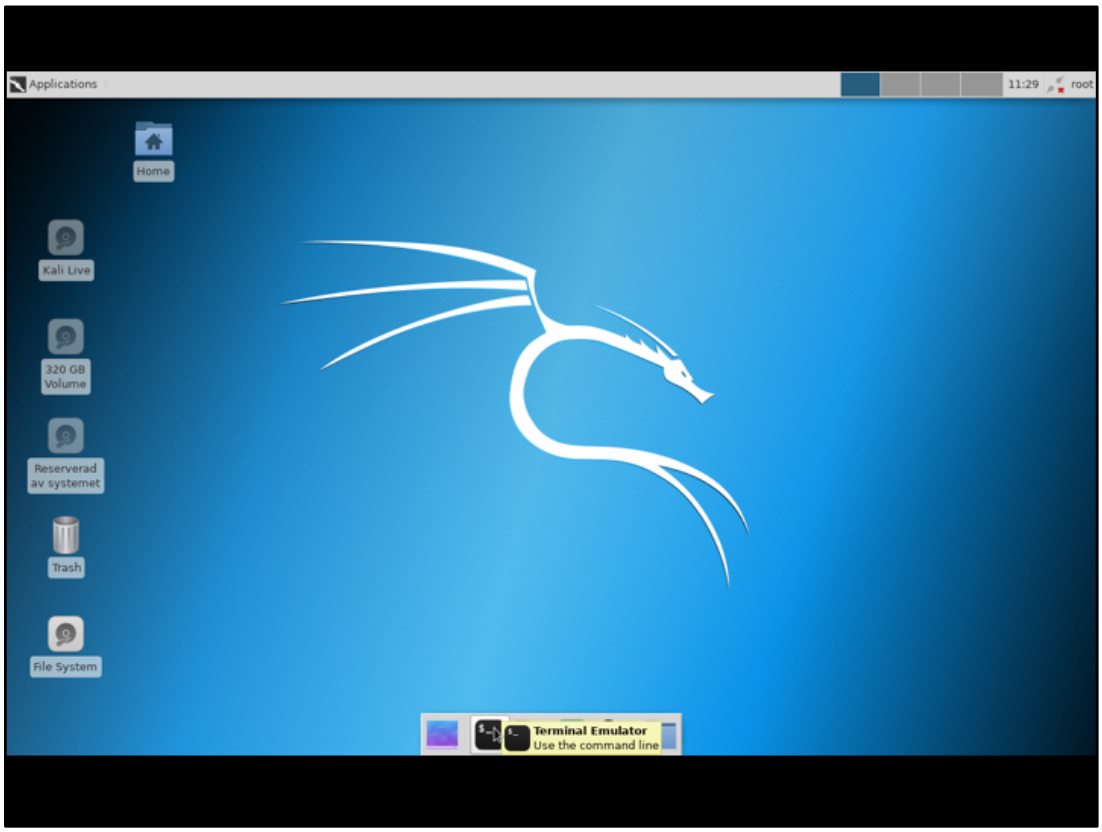


Välj önskad layout. Jag väljer Swedish utan döda tangenter.

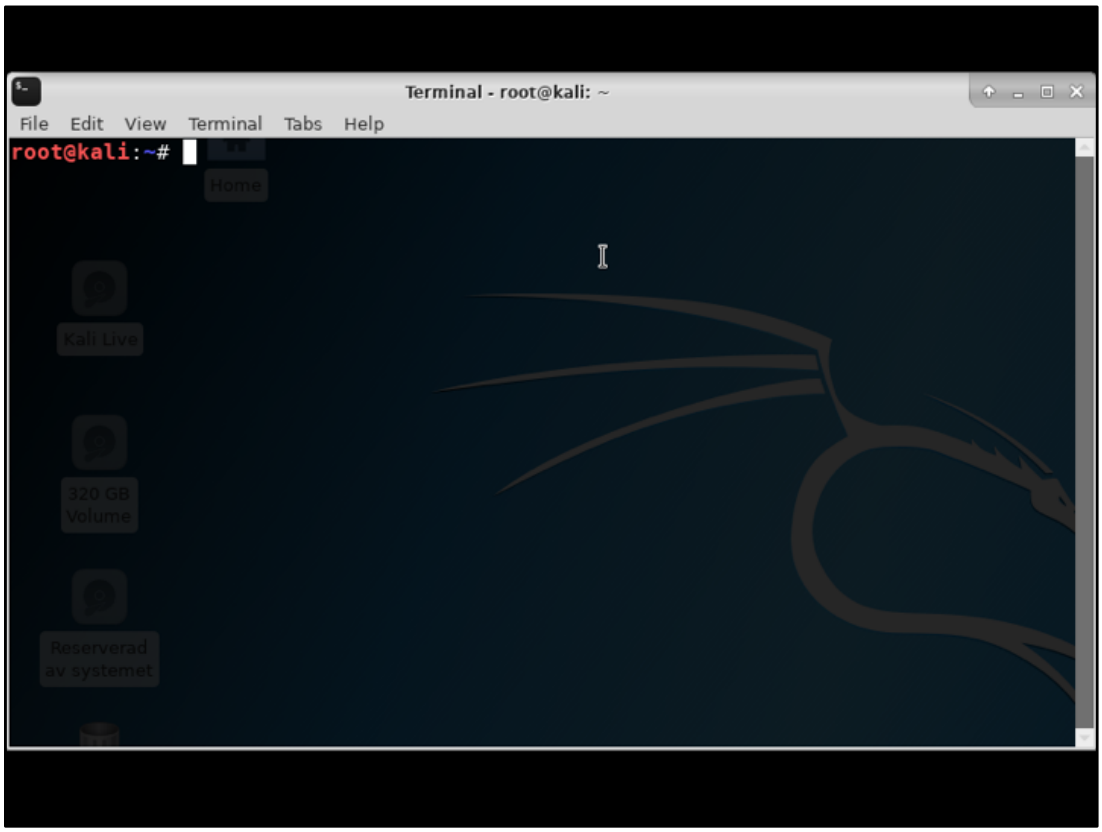


Nu borde man kunna växla layout, men jag har inte fått det att fungera. Radera därför “English (US)”

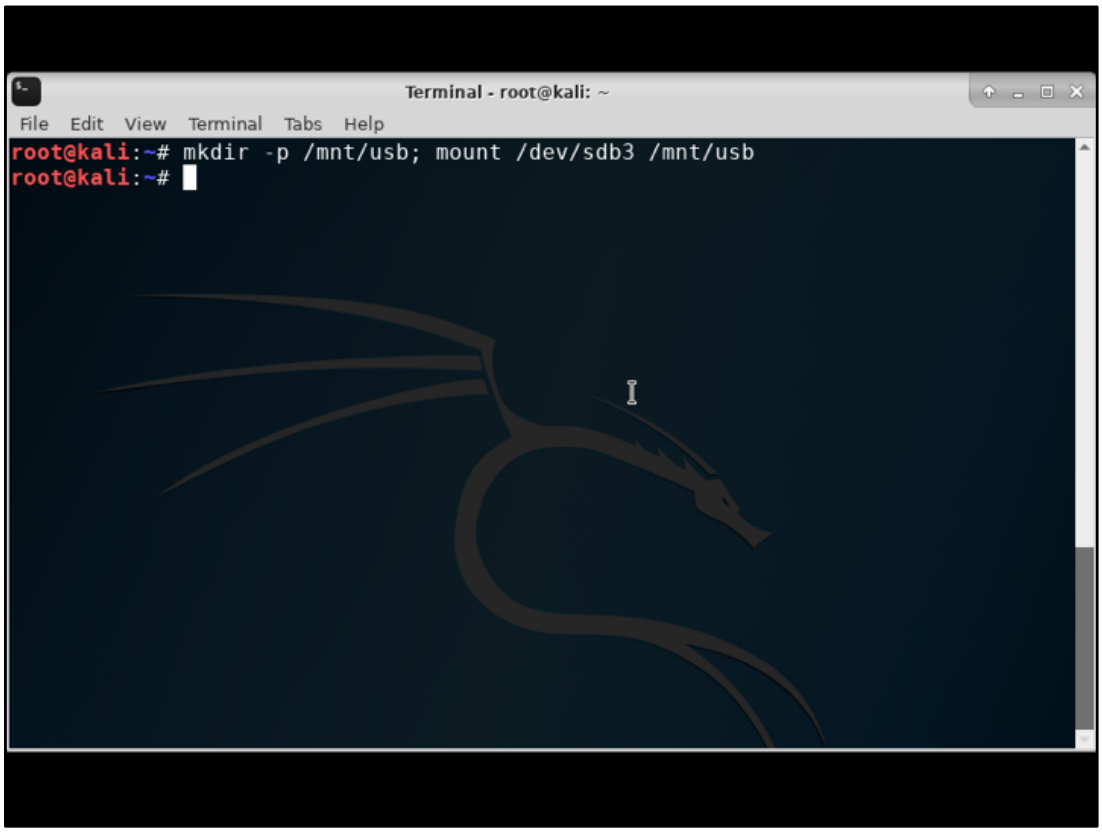
Avsluta med [Close]



Starta en terminal



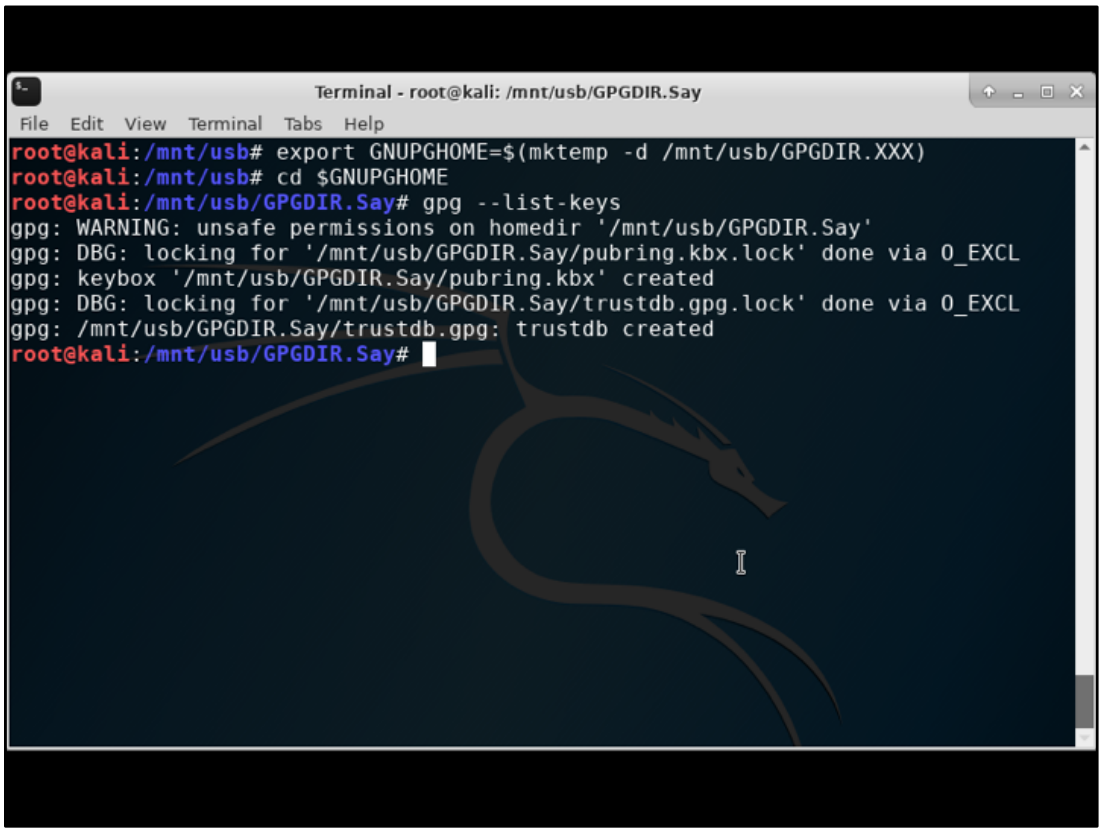
Terminal redo att börja



Förkrav är att ha skapat en partition på 100MB
För att hitta rätt device:

```
DEVICE=$(df /run/live/medium |tail -1|cut -d " " 1); DEVICE=${DEVICE%%[0-9]*}
parted $DEVICE print
```

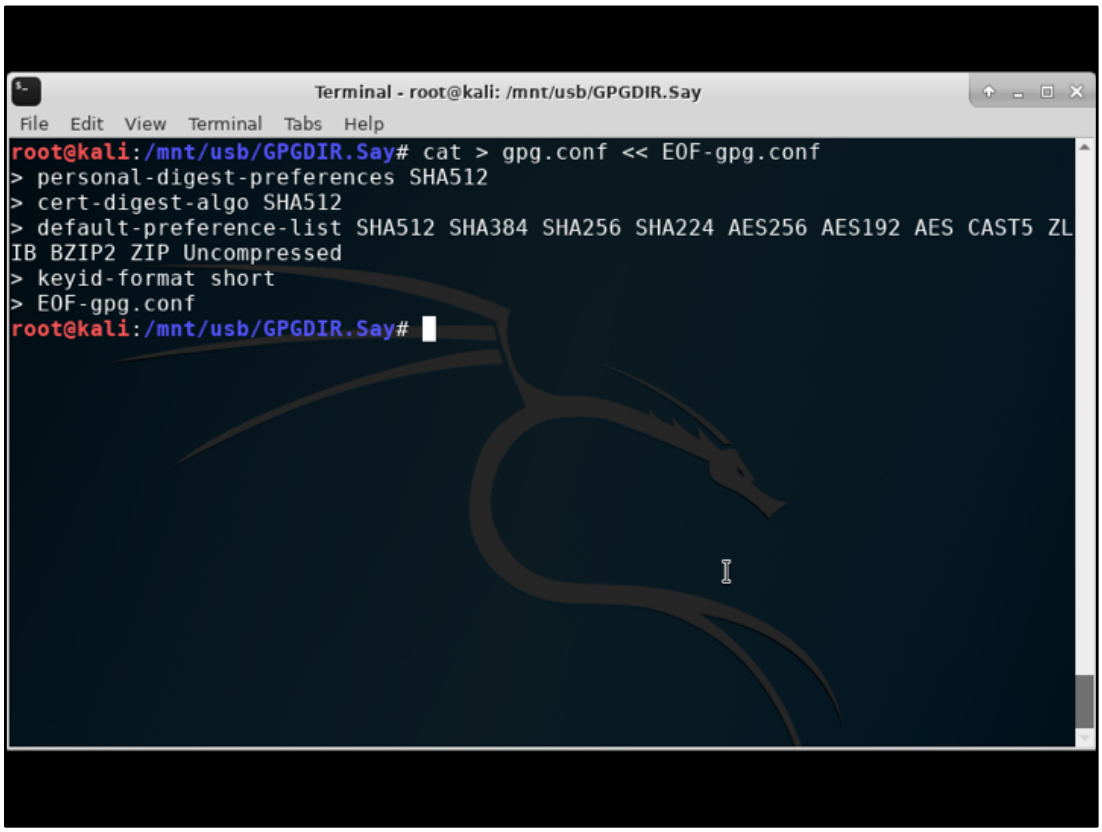
Skapa monteringspunkt /mnt/usb
Montera /dev/sdb3 på monteringspunkt /mnt/usb



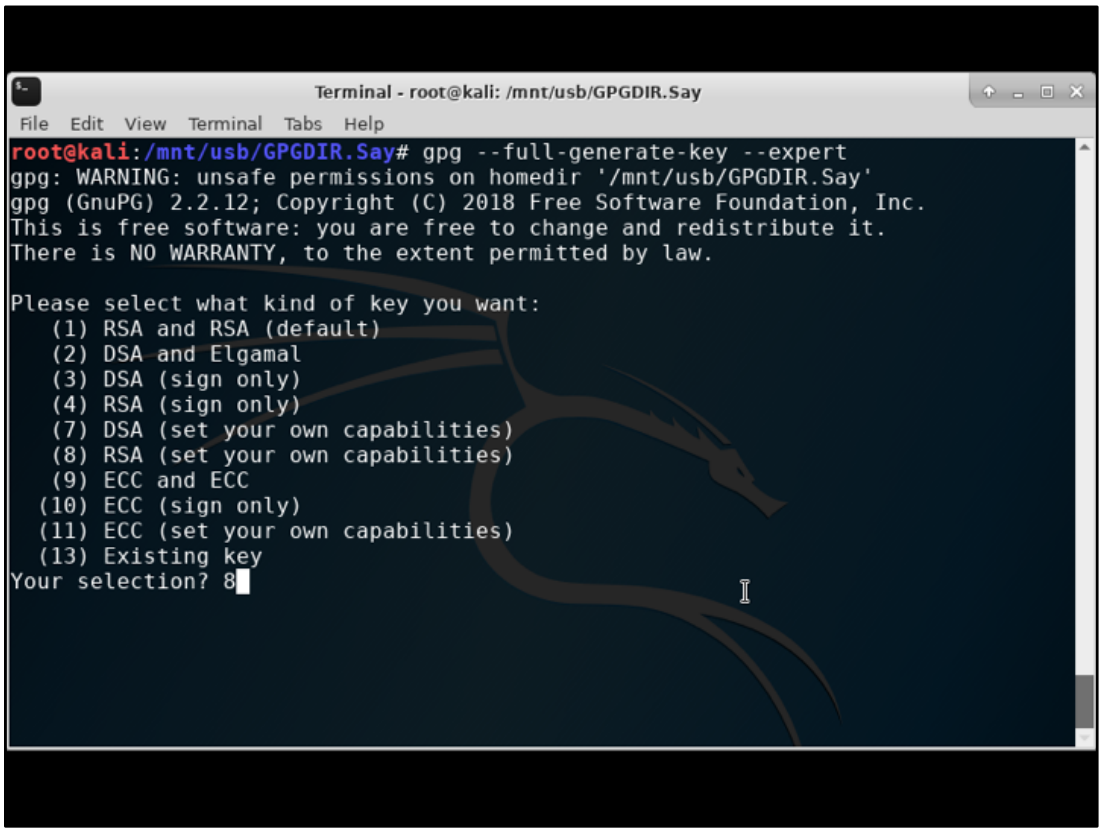
Skapa en katalog för den nya nyckeln och lagra namnet i omgivningsvariabel GNUPGHOME
gpg --list-keys

- Listar nycklar i nyckelring.
- Finns ingen nyckelring skapas en ny, tom sådan
- Viktigt test för att se att vi har en fräsch installation

Varning för att rättigheterna på GNUPGHOME är osäkra, vilket vi kan ignorera nu när vi kör i en isolerad miljö.



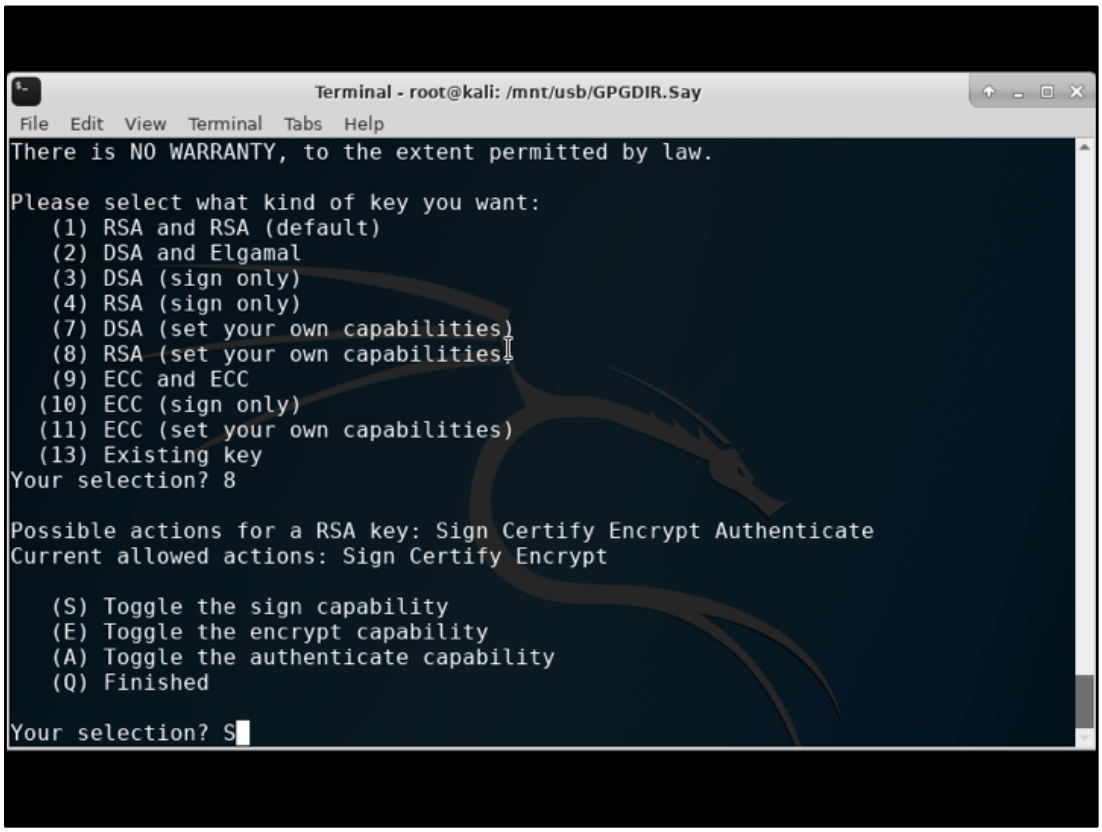
Skapa en gpg.conf med lämpliga värden så att vi får bästa möjliga nycklar
keyid-format short för att lättare kunna arbeta med nycklarna



Dags att generera vår masternyckel.

Kommando: gpg –full-generate-keys –expert

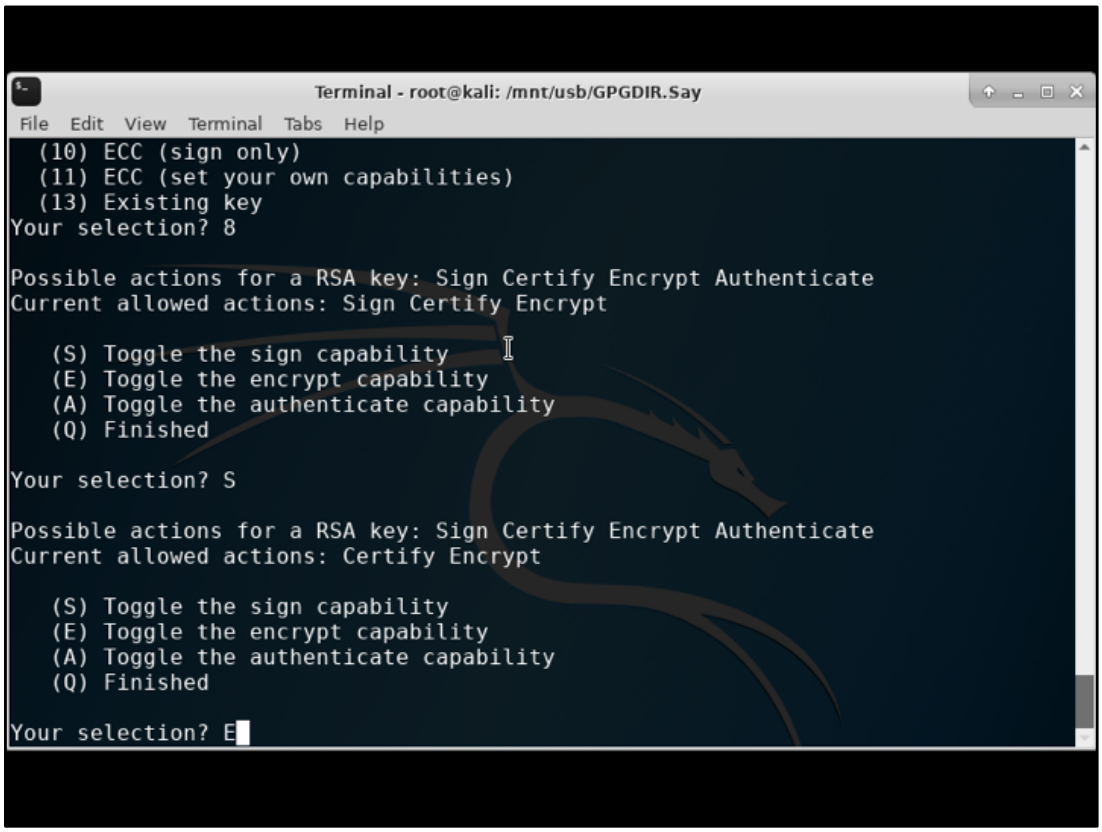
Välj RSA-nyckel (set your own capabilities). Val: 8



Primärnyckel har normalt sett tre egenskaper

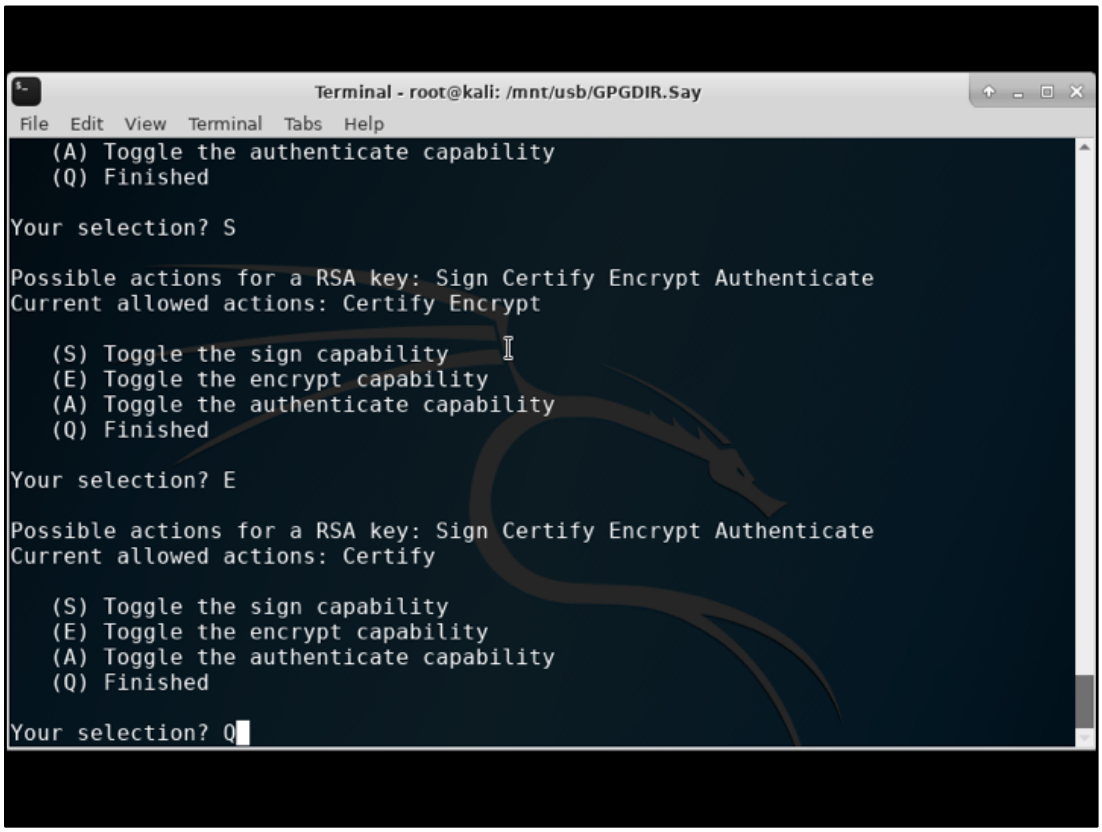
- Sign. Signera dokument
- Certify. Signera nycklar
- Encrypt. Krytpera dokument

Vi skall bara ha Certify och behöver därför ta bort S(ign) och E(ncrypt)
Val: S



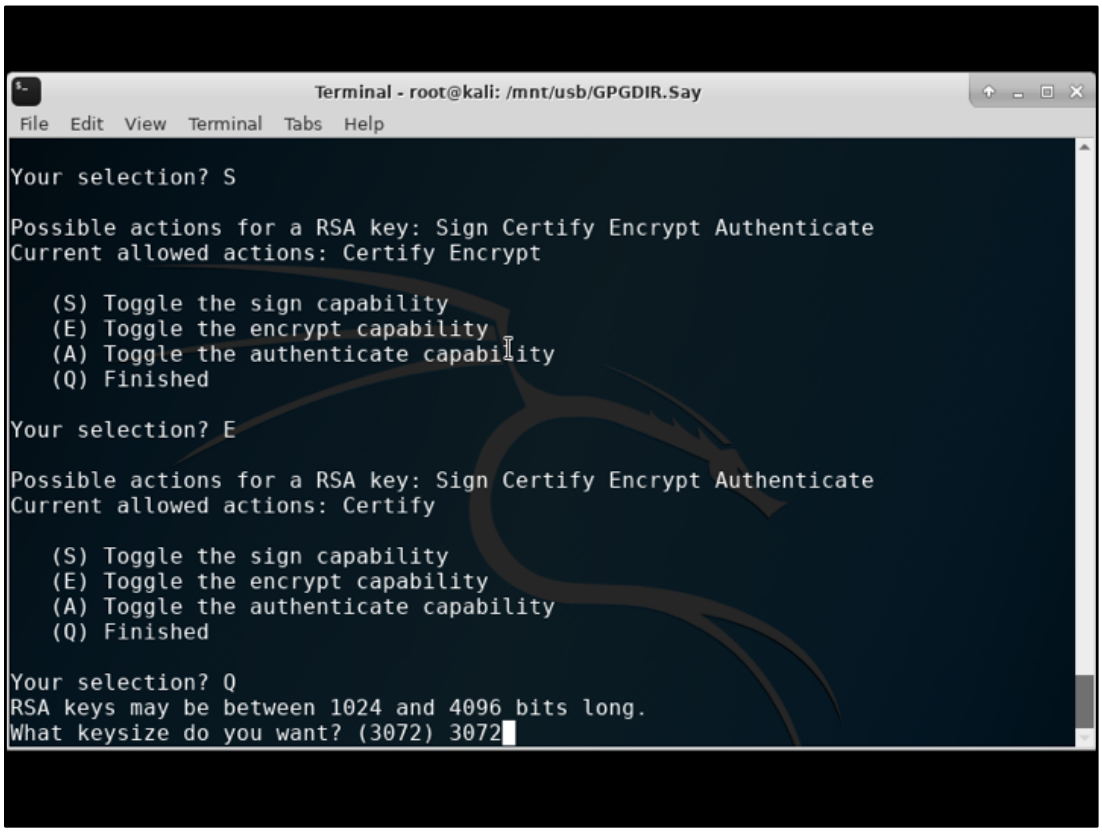
Här syns att Sign är borttaget och nu skall Encrypt bort också

Val: E

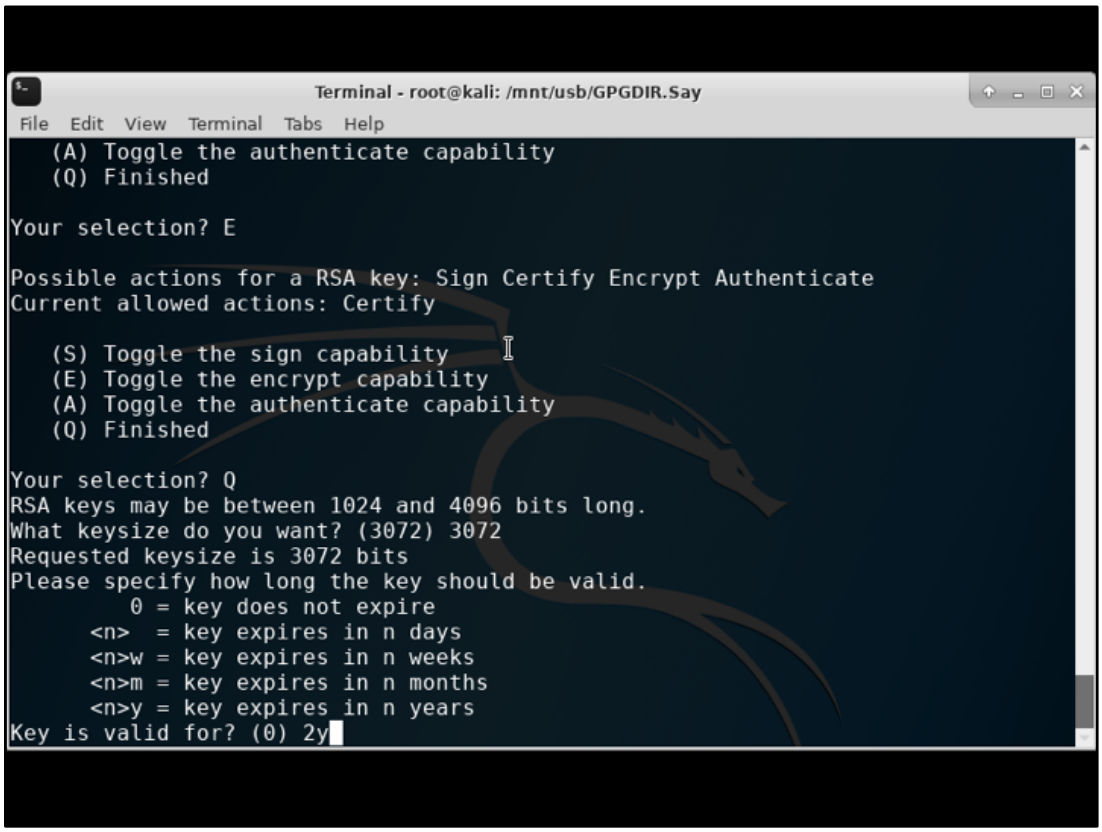


Nu ser vi att det bara är Certify kvar och vi kan avsluta valet av egenskaper.

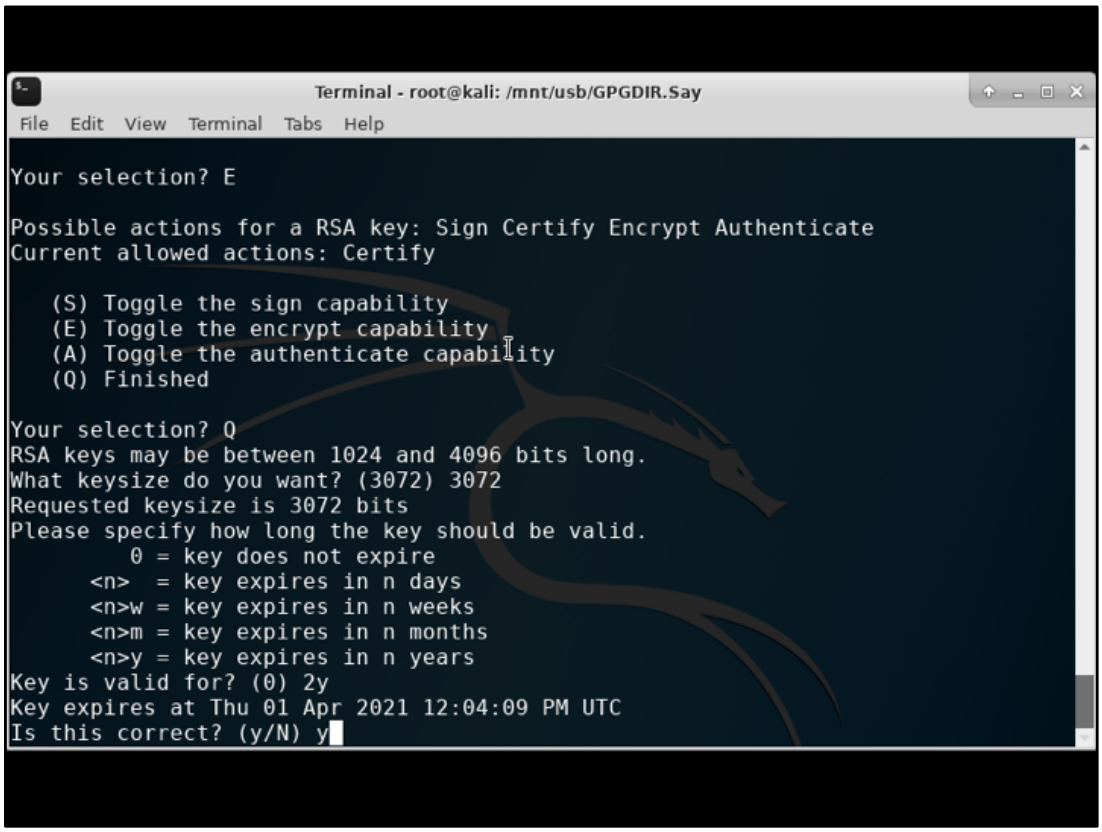
Val: Q



Nu är det dags att välja nyckellängd.
Enligt NIST är 2048 bara tillräckligt fram till 2030. 3072 skall räcka, 4096 ger inte så mycket extra skydd, men kräver mer beräkningskraft än 3072.
Val: 3072

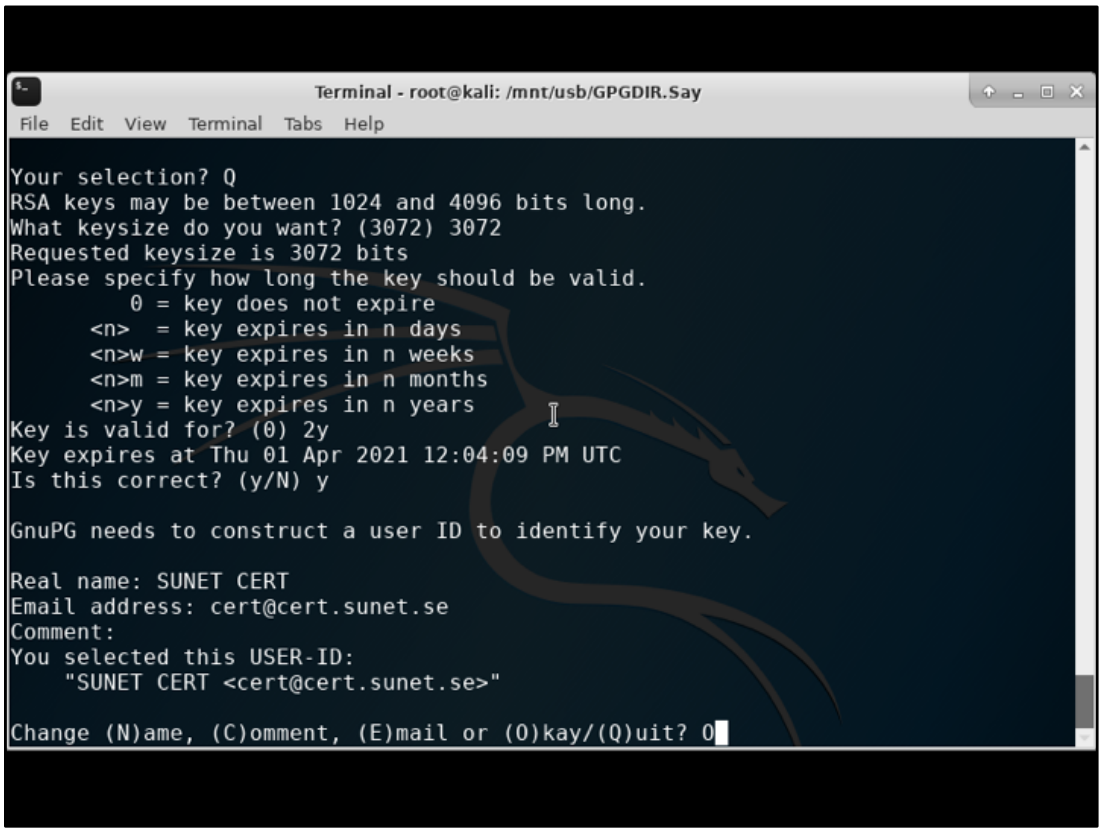


Nu skall vi välja giltighetstid.
Viktigt att veta att vi alltid kan förlänga giltighetstid, så att sätta detta är ett extra skydd om nyckel och revokeringscertifikat skulle gå förlorade. Välj ett så lågt värde som ni anser är hanterbart. Vi väljer 2 år i den här övningen.
Val: 2y



Nu skall vi välja giltighetstid.
Viktigt att veta att vi alltid kan förlänga giltighetstid, så att sätta detta är ett extra skydd om nyckel och revokeringscertifikat skulle gå förlorade. Välj ett så lågt värde som ni anser är hanterbart. Vi väljer 2 år i den här övningen.
Val1: 2y

Val2: y

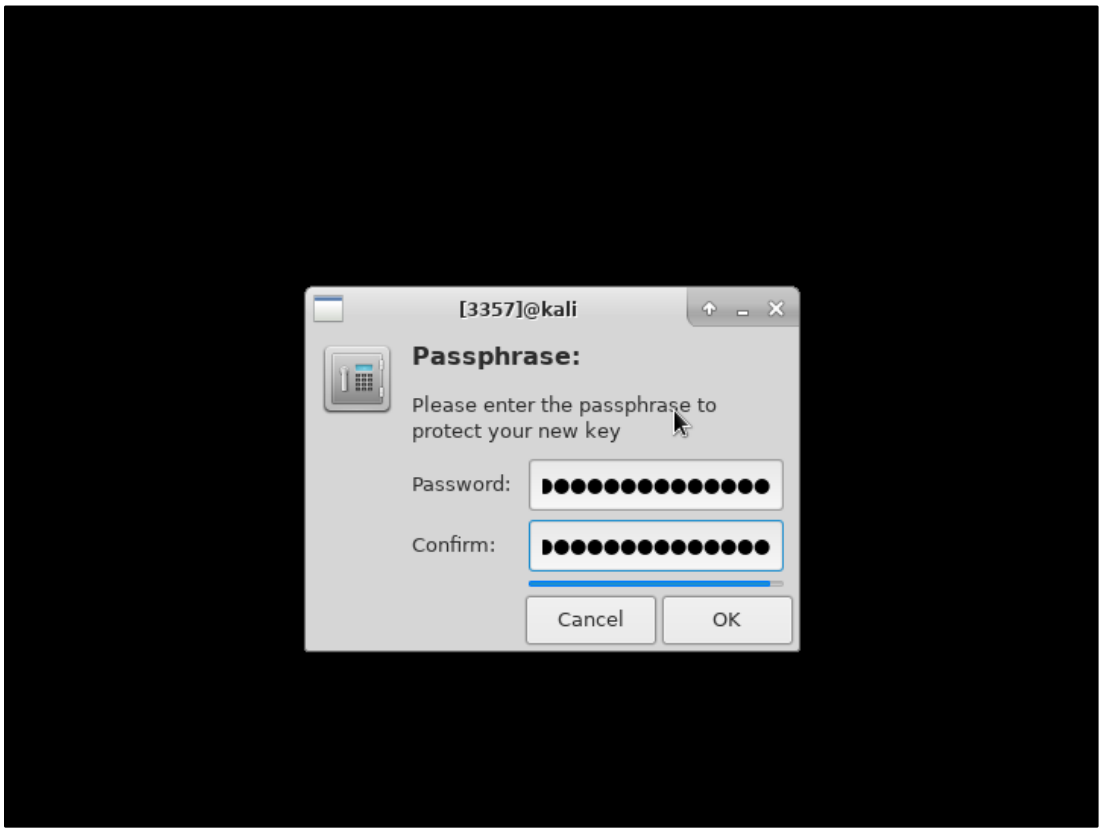


Sätt en identitet på nyckeln. Real name och Email address bör sättas även om Email address är det enda som krävs.

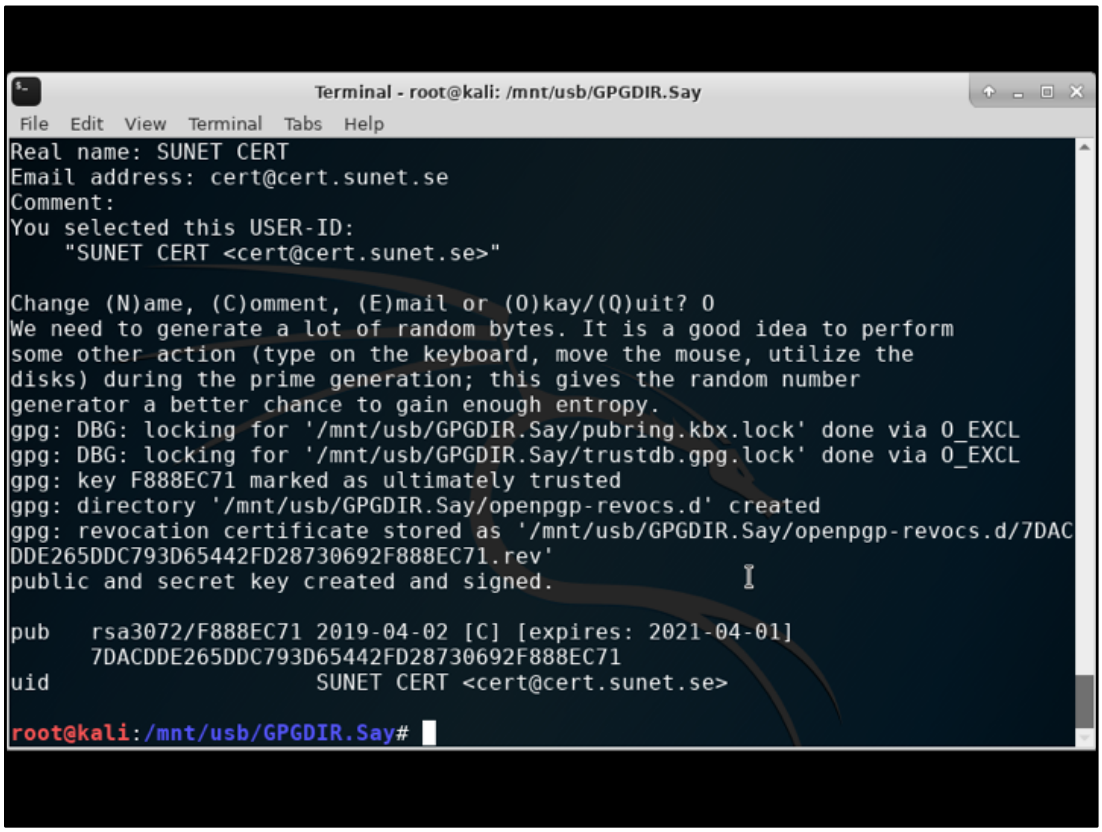
Val 1: <Namn på ditt team>
Val 2: <epost till ditt team>

Val 3: <tomt>

Val 4: O

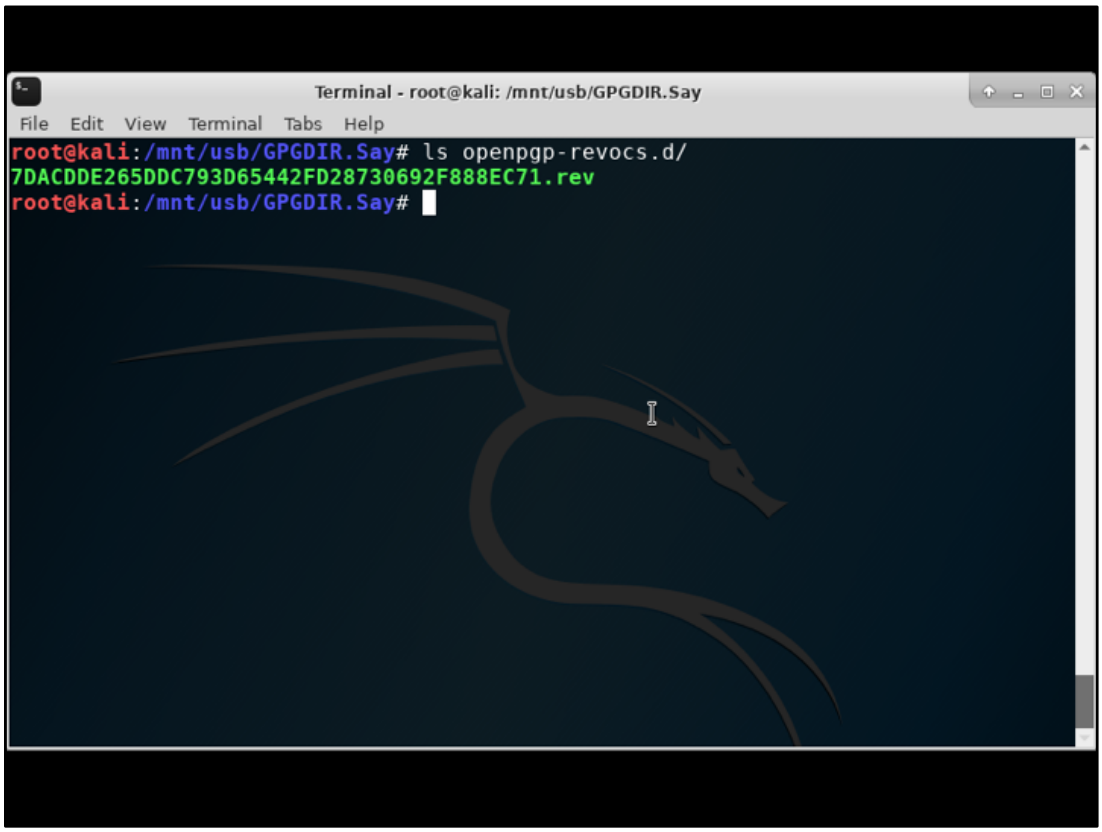


Välj en bra passphrase för din nyckel. Det här är enda skyddet som återstår om någon kommer åt din hemliga nyckel.

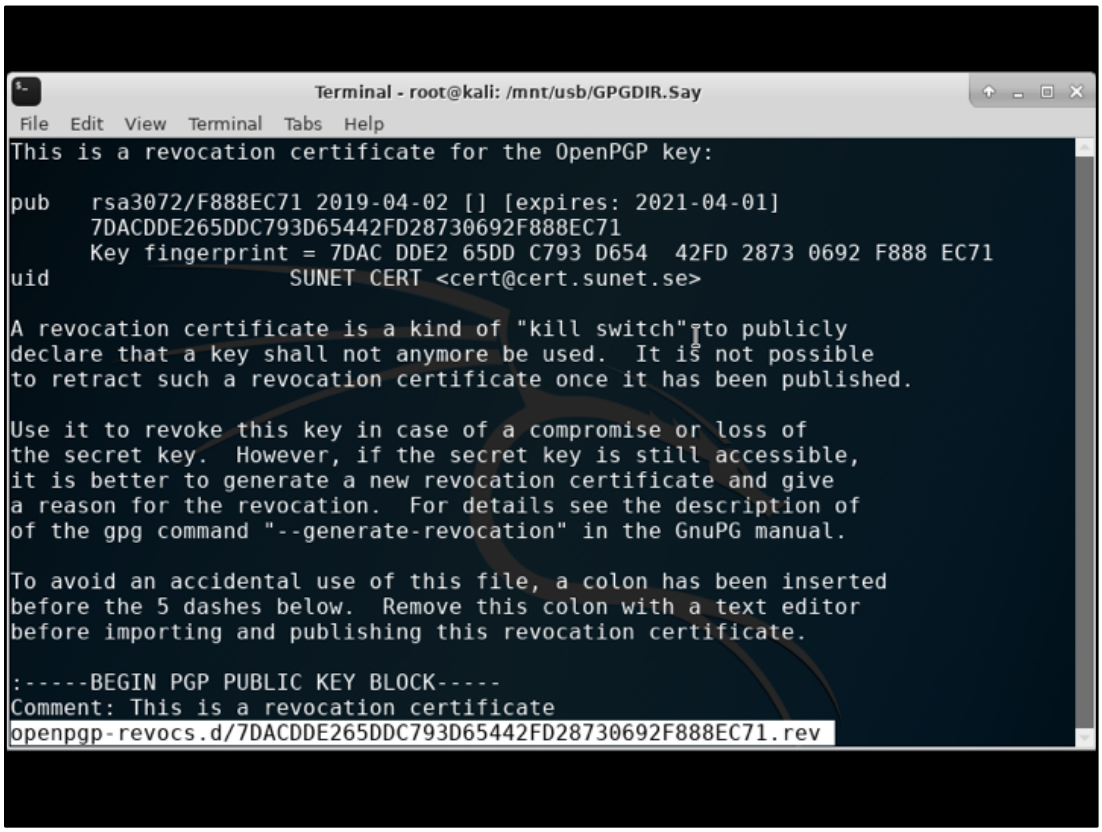


Efter en stund är nyckeln klar (rör på musen om det går långsamt)
Vi ser att vi har en nyckel utan undernycklar och att den bara har egenskapen Certify [C]

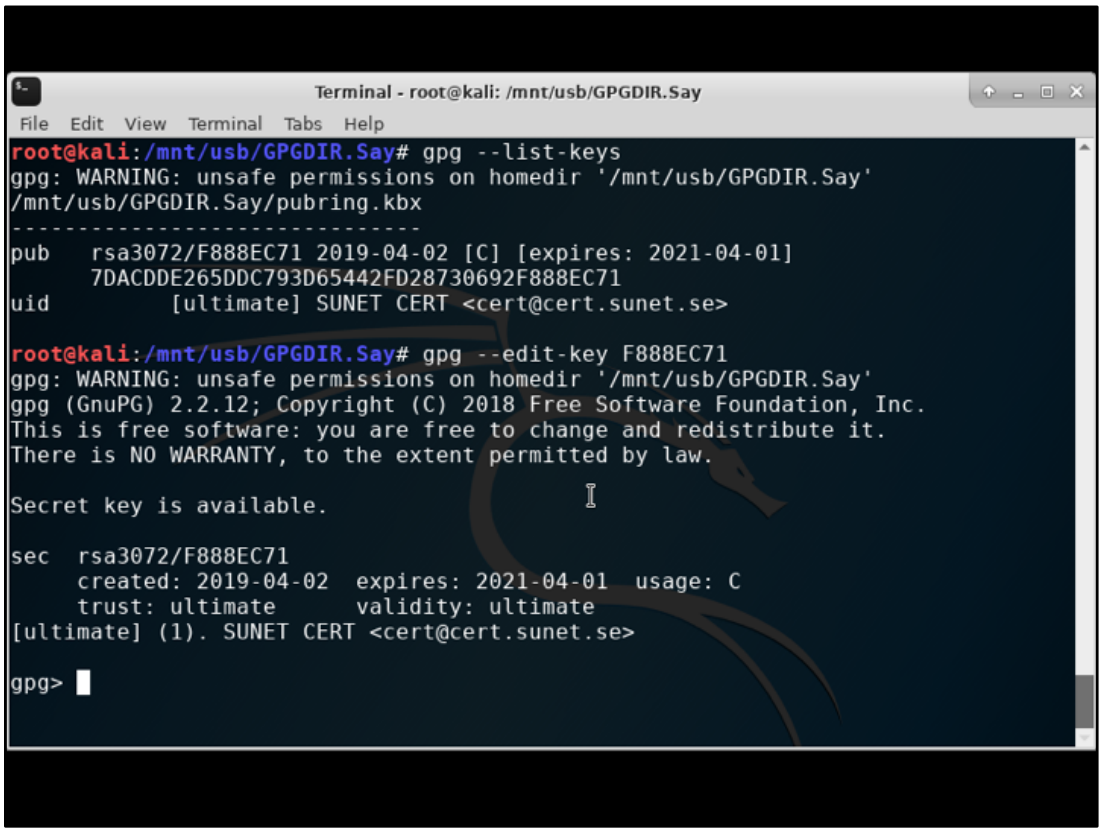
Kort nyckelid efter nyckellängden rsa3072/**F888EC71**



Vi ser här att en revokeringsnyckel har skapats automatiskt

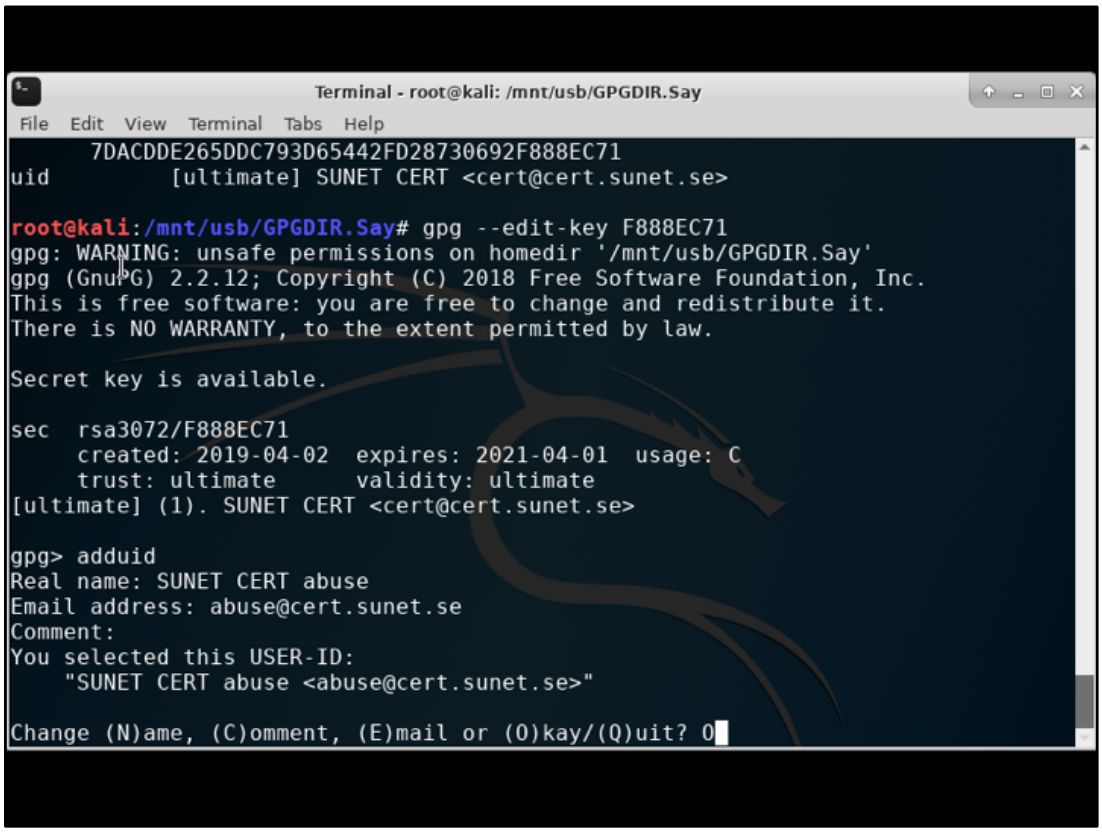


Början på revokeringsfilen. Den här kan vara bra att skriva ut på papper och förvara i kassaskåp bredvid USB-stickan med den hemliga nyckeln. Skulle alla digitala kopior vara förstörda kan man (mödosamt) återskapa den och revokera en förlorad nyckel.



Dags att lägga till fler identiteter (behövs bara om ni har fler identiteter som skall använda samma nyckel)

Kommando: gpg –edit-key <NYCKELID>



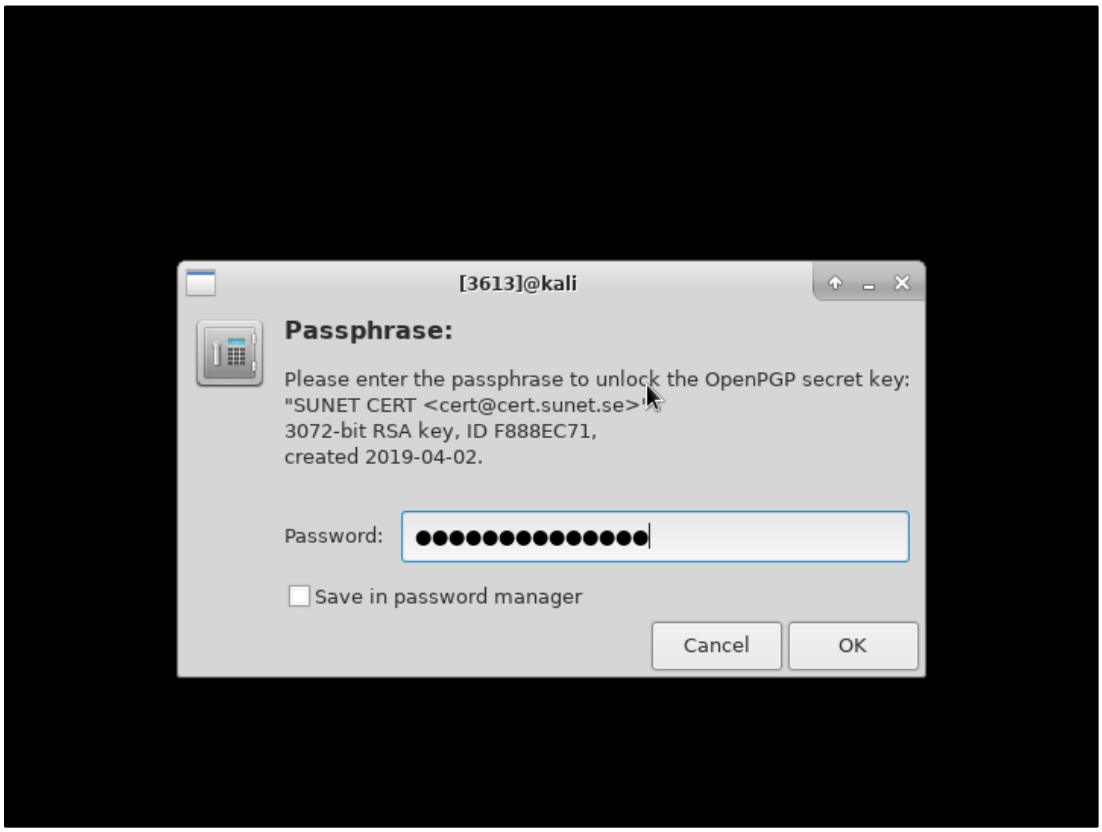
Extra identiteter läggs till med **adduid**

Val: <Namn på identiteten>

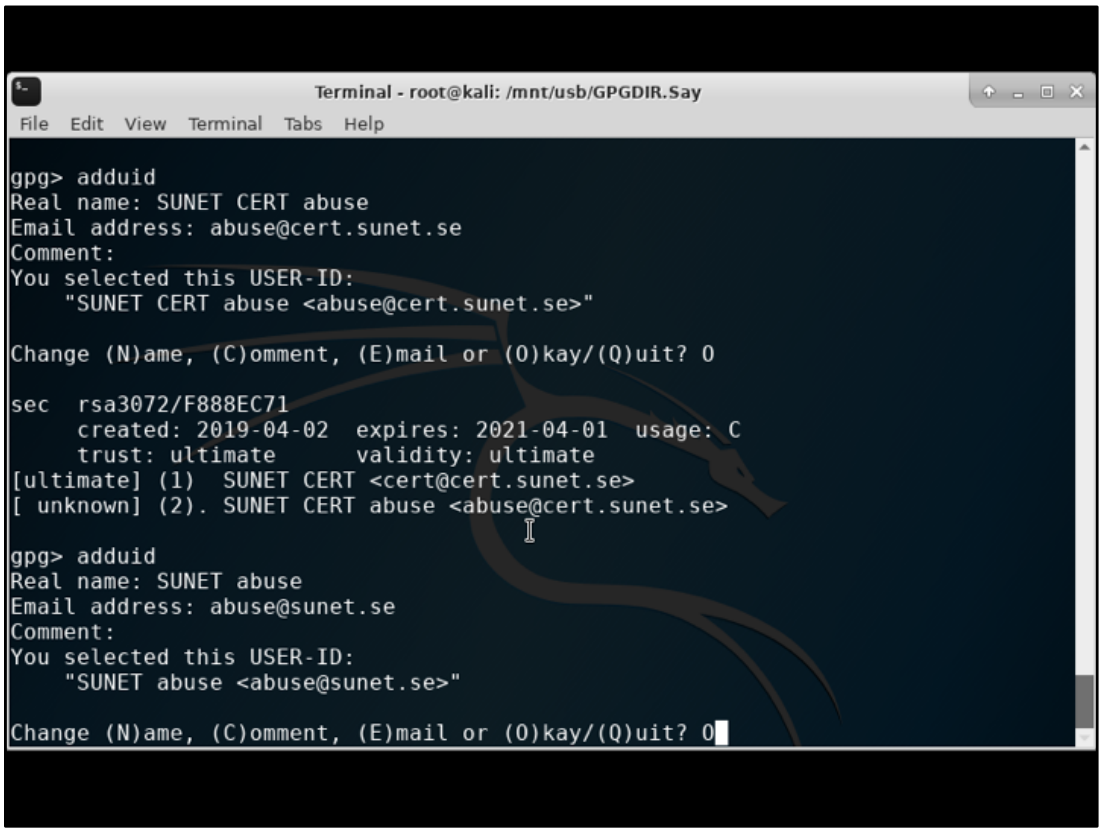
Val: <epost för identiteten>

Val: <tomt>

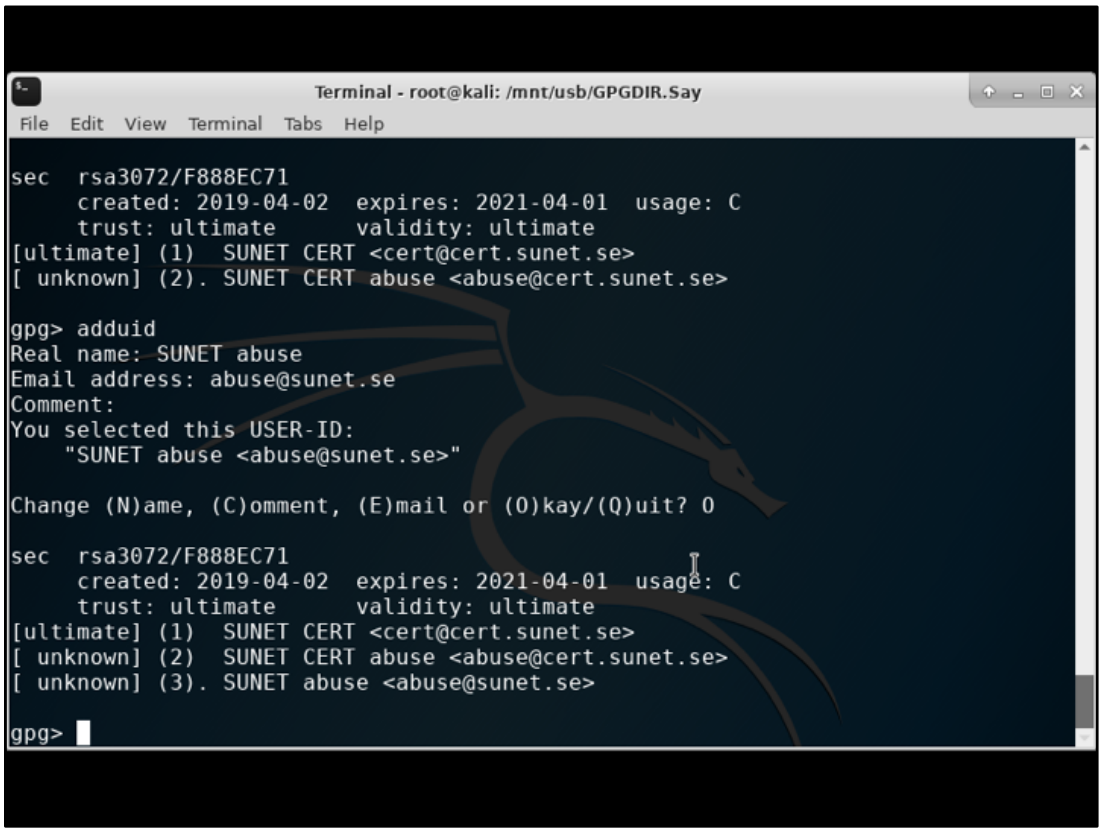
Val: O



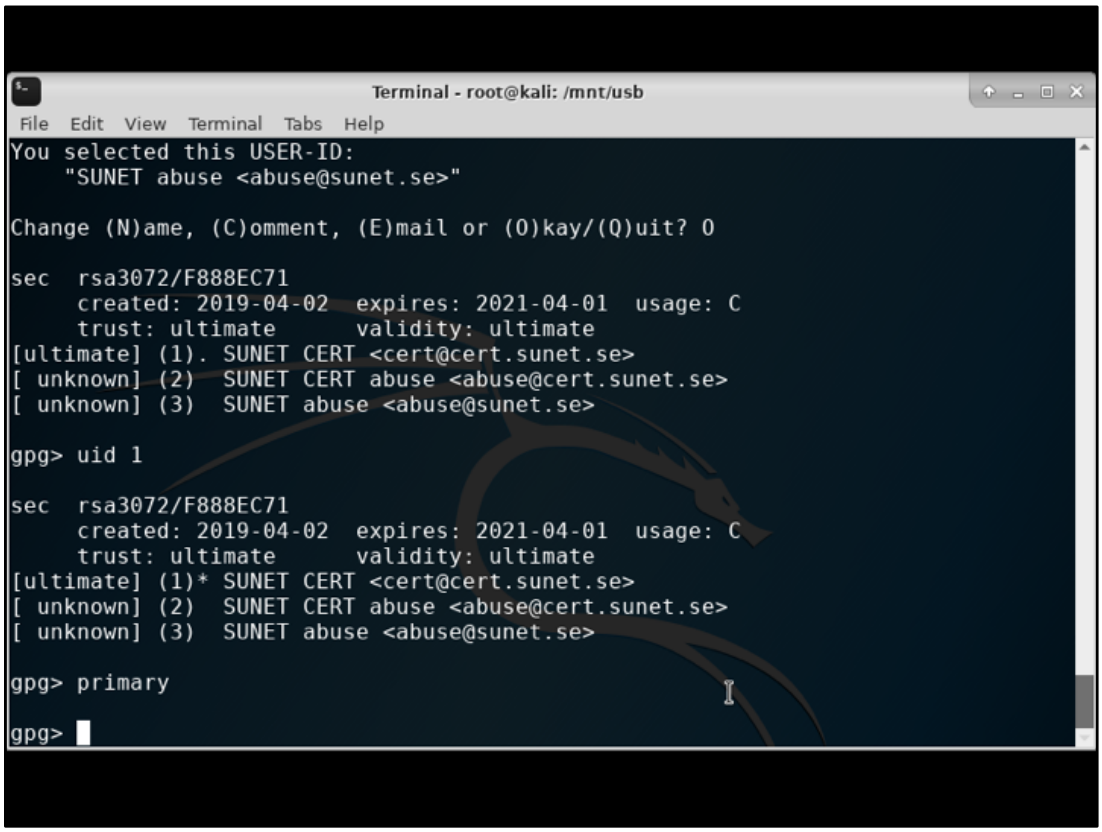
Då vi skall ändra nyckeln så krävs att vi anger lösenordsfras. Senare i exempeln visas inte denna uttryckligen, men du förutsätts fylla i den när så behövs.



Vid behov lägg till ytterligare identiteter.



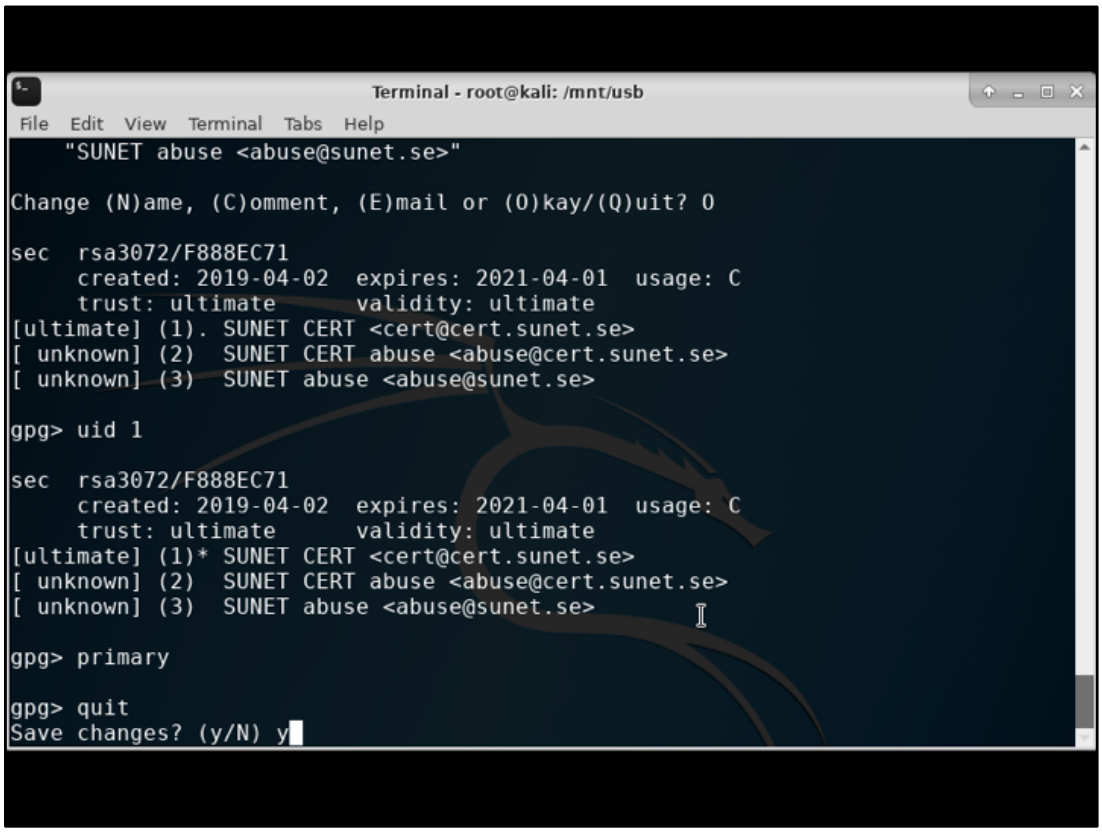
Vid behov lägg till ytterligare identiteter.



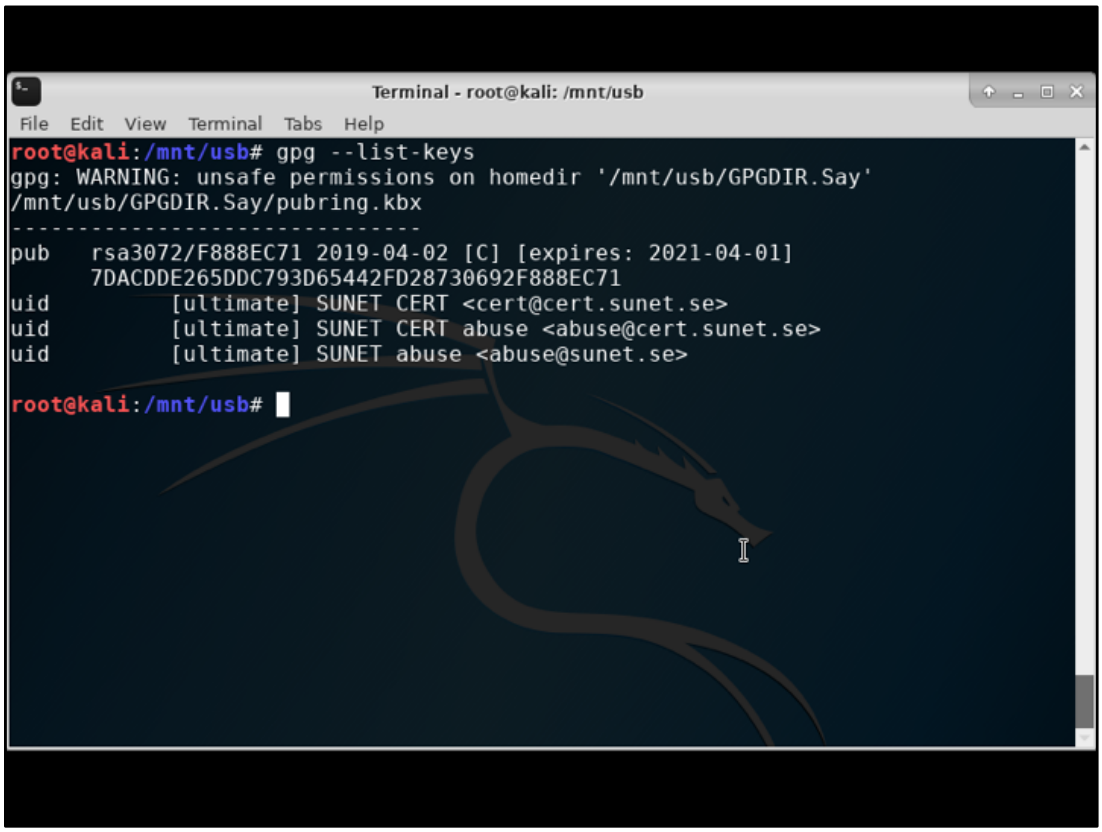
Välj den uid som skall vara primär.

Val 1: uid 1

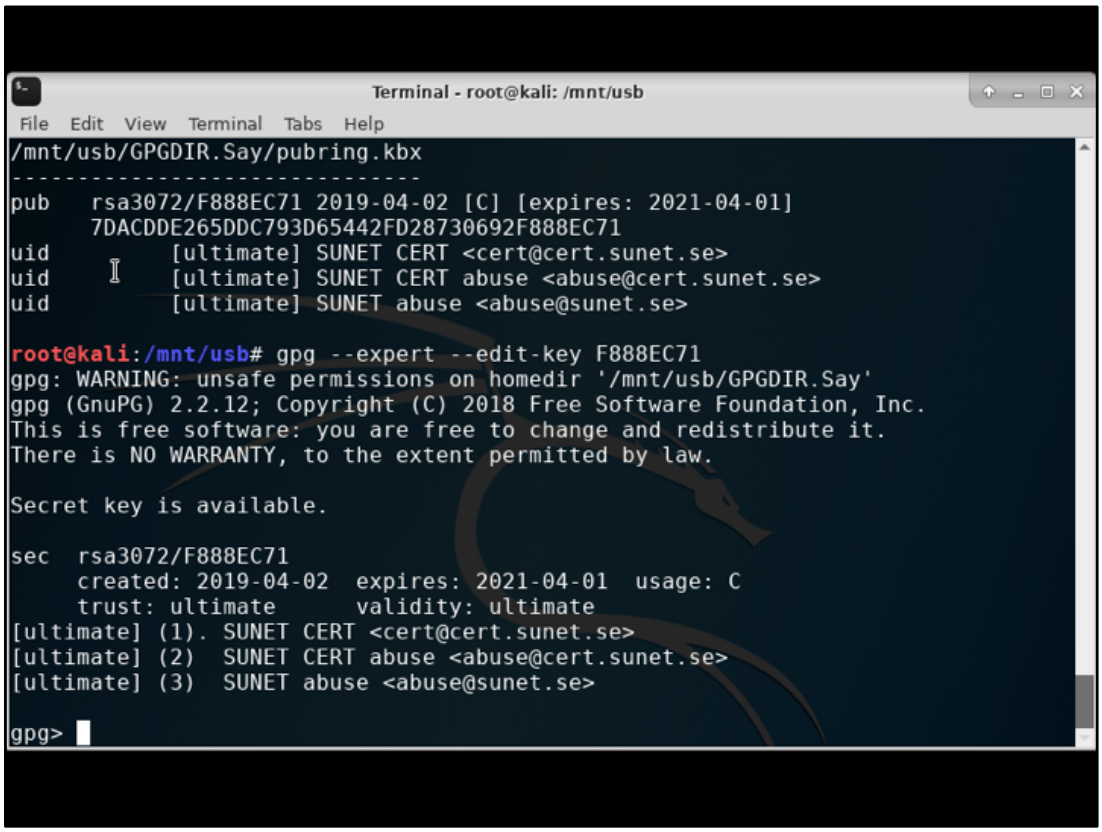
Val 2: primary



Här ser vi nu att nyckeln har tre identiteter och vi kan avsluta och spara.

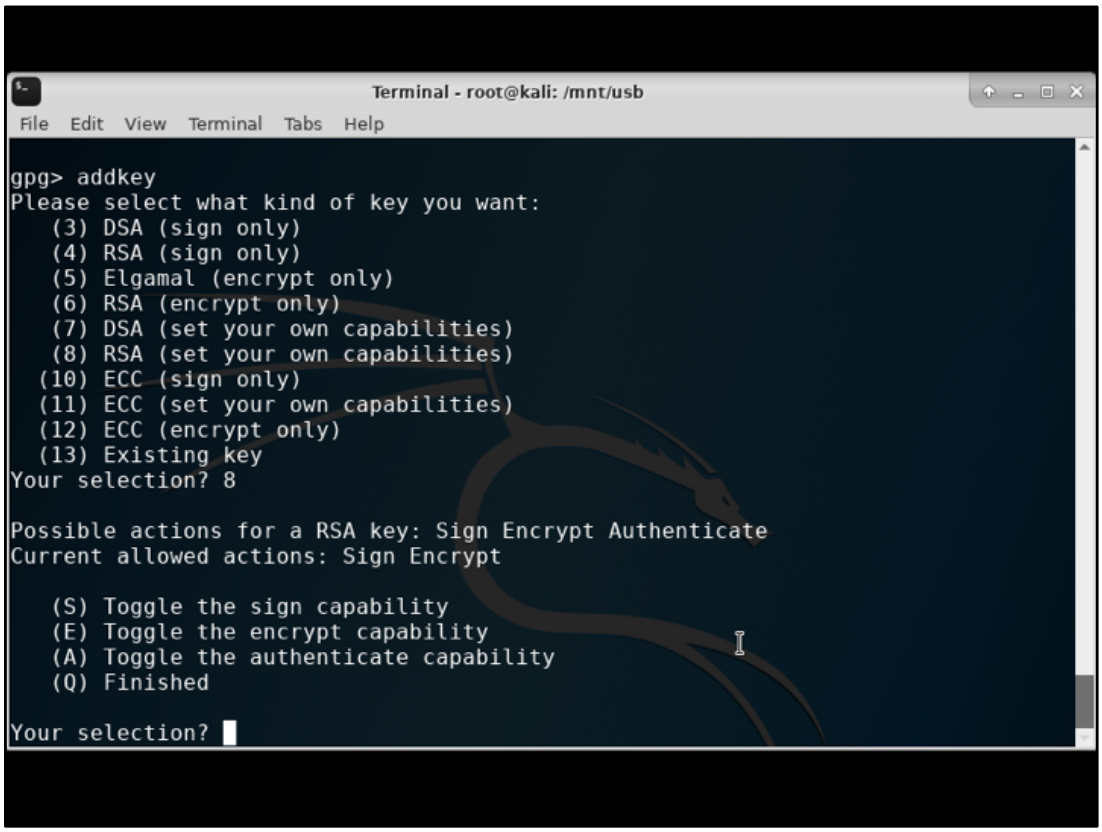


Listar vi nyckeln ser vi de nya identiterna och att vi automatiskt fått [ultimate] trust för dem.
Den vi valde som primary listas först.



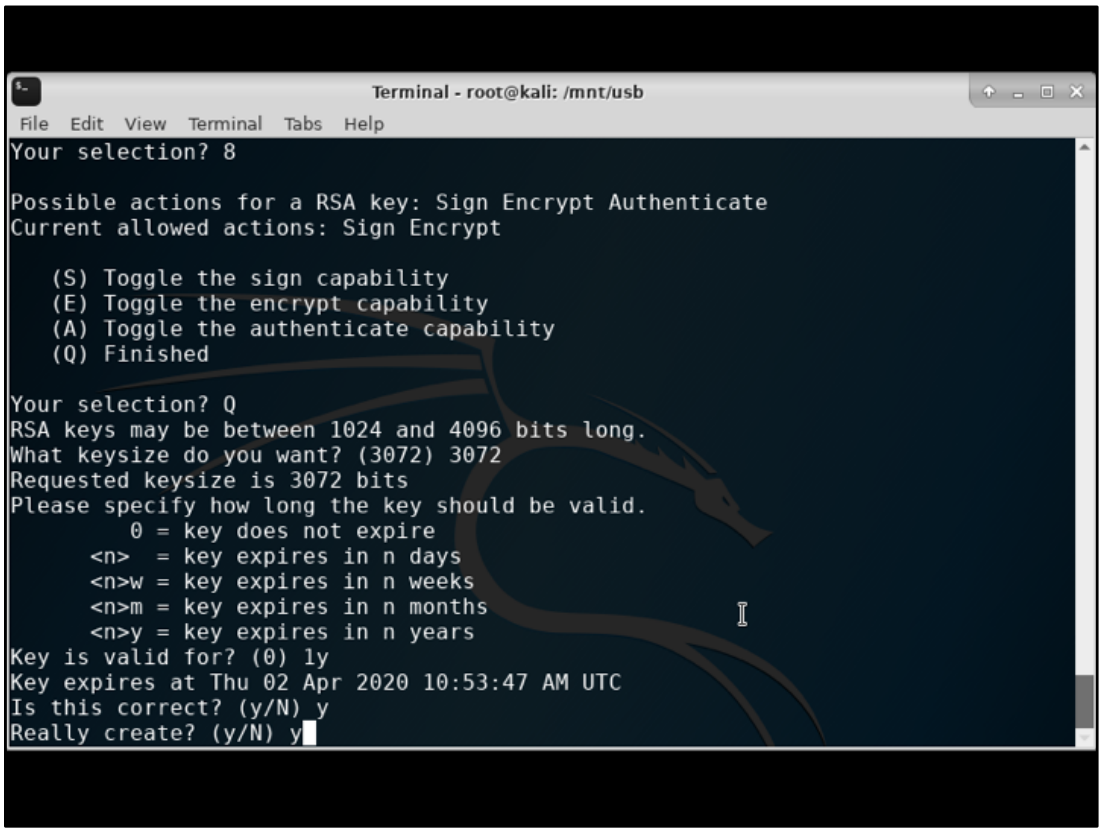
Nu är det dags att skapa subnycklar för kryptering och signering.

Kommando: gpg –expert –edit-key <KEYID>



Skapa en subnyckel med egenvalda egenskaper

Val 1: 8



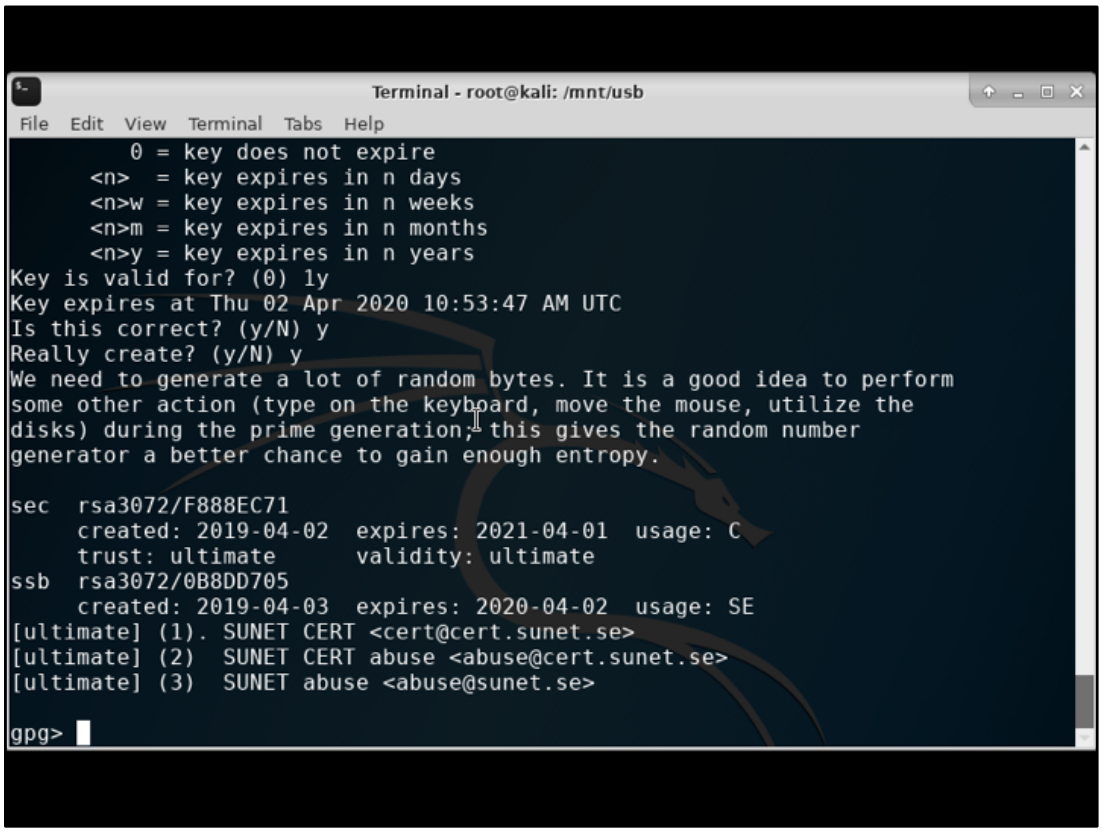
Vi är nöjda med Sign och Encrypt på nyckeln. Vi vill ha en 3072 bitar lång nyckel med 1 års giltighetstid.
Den kortarte tiden är för att detta är en nyckel som skall delas och användas på maskiner anslutna till nätet.
Val 1: Q

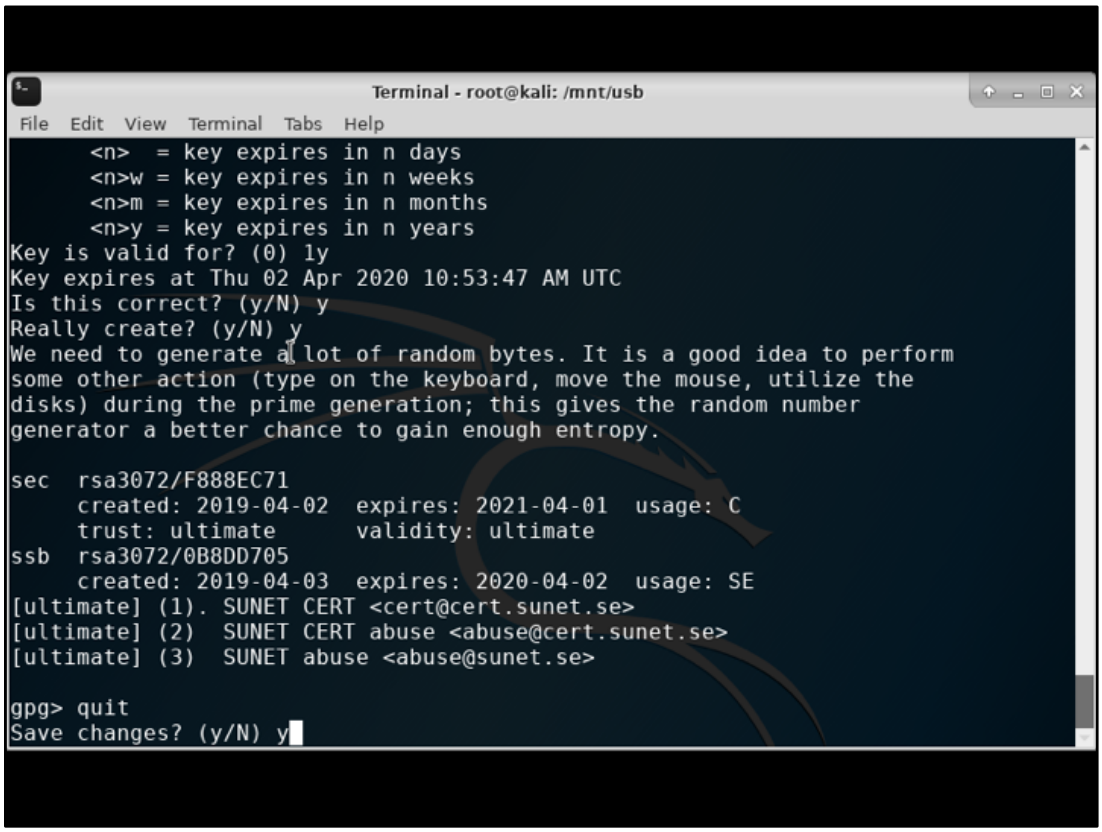
Val 2: 3072

Val 3: 1y

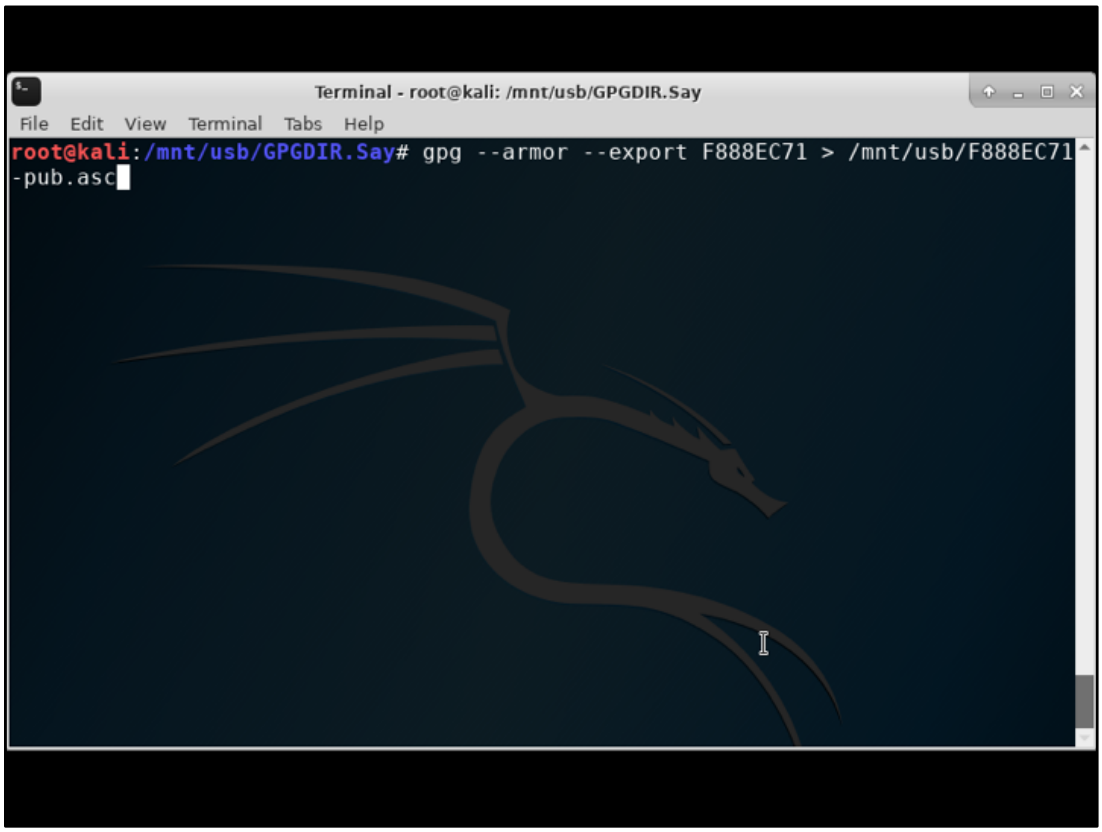
Val 4: y

Val 5: y


Terminal - root@kali: /mnt/usb
File Edit View Terminal Tabs Help
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 Thu 02 Apr 2020 10:53:47 AM UTC
Is this correct? (y/N) y
Really create? (y/N) y
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.
sec rsa3072/F888EC71
created: 2019-04-02 expires: 2021-04-01 usage: C
trust: ultimate validity: ultimate
ssb rsa3072/0B8DD705
created: 2019-04-03 expires: 2020-04-02 usage: SE
[ultimate] (1). SUNET CERT <cert@cert.sunet.se>
[ultimate] (2) SUNET CERT abuse <abuse@cert.sunet.se>
[ultimate] (3) SUNET abuse <abuse@sunet.se>
gpg>



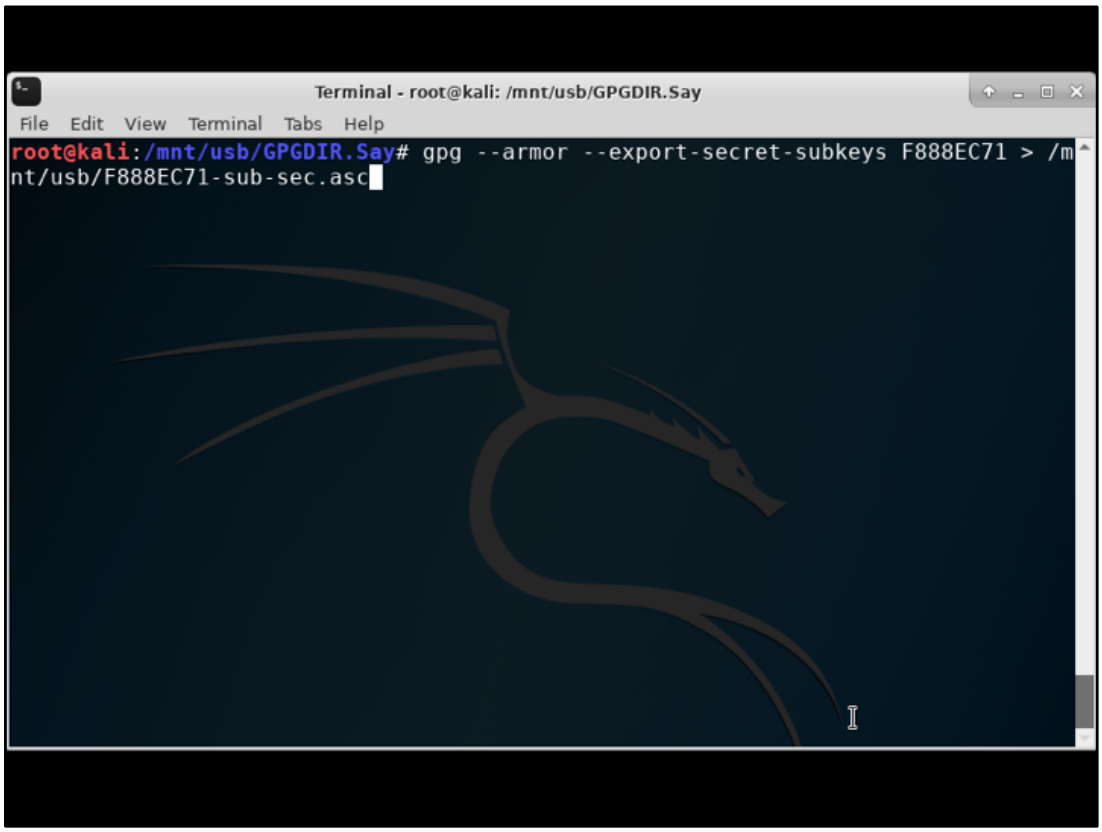
Avlsuta och spara nyckeln.



För att vi och andra skall kunna använda nycklarna måste den publika delen exporteras och publiceras. Både master och subnycklar exporteras.
Kommando: gpg –armor –export <KEYID> > /mnt/usb/<KEYID>-pub.asc

--armor : Konvertera binär nyckel till portabelt ASCII format

--export : Exportera publik nyckel

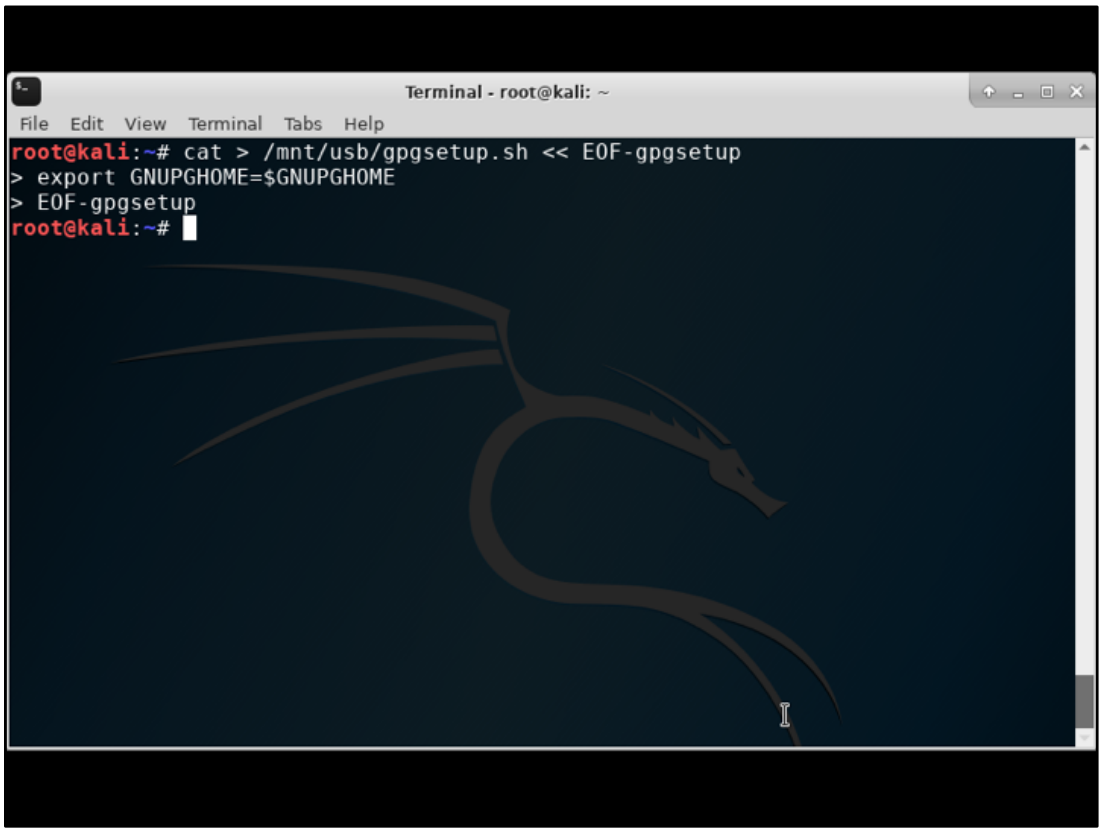


För att vi skall kunna arbeta med signering och kryptering måste vi ha tillgång till de privata subnycklarna, men inte den privata masternyckeln. Här exporterar vi endast subnycklarna.

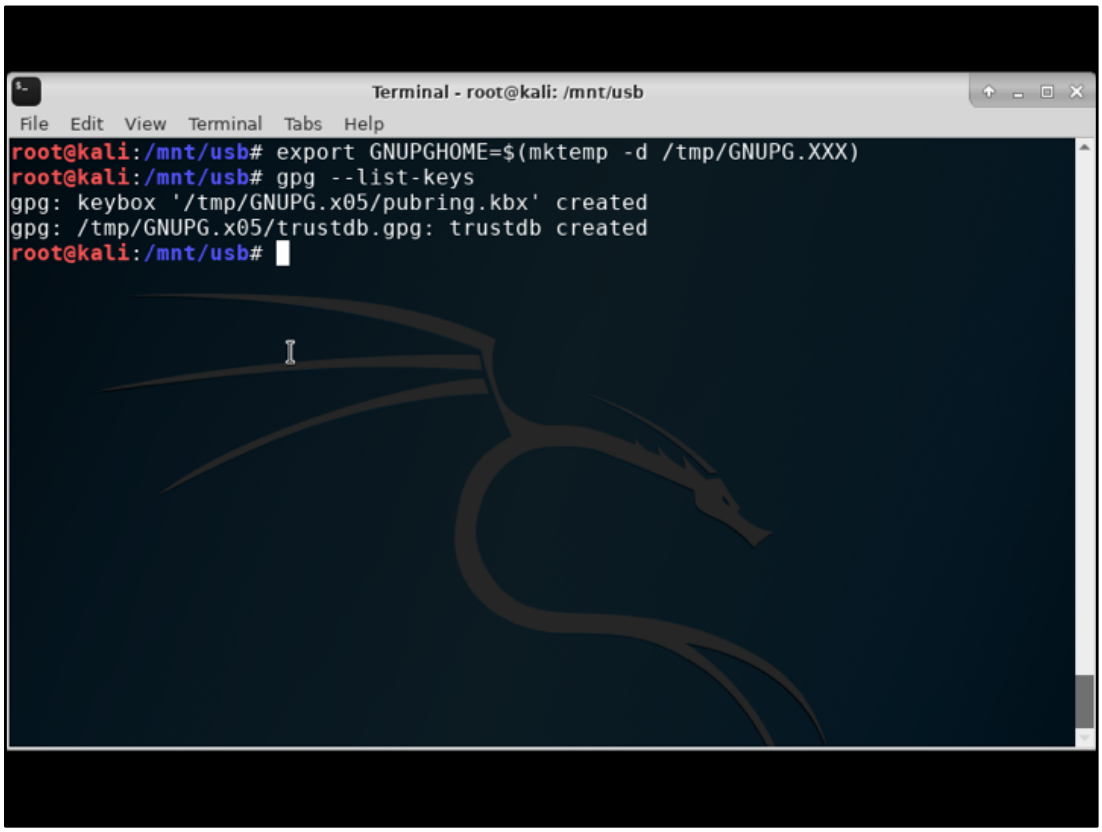
Kommando: gpg –armor –export-secret-subkeys <KEYID> > /mnt/usb/<KEYID>-sub-sec.asc

-- armor : Konvertera binär nyckel till portabelt ASCII format

-- export-secret-subkeys : Exportera bara de hemliga subnycklarna.



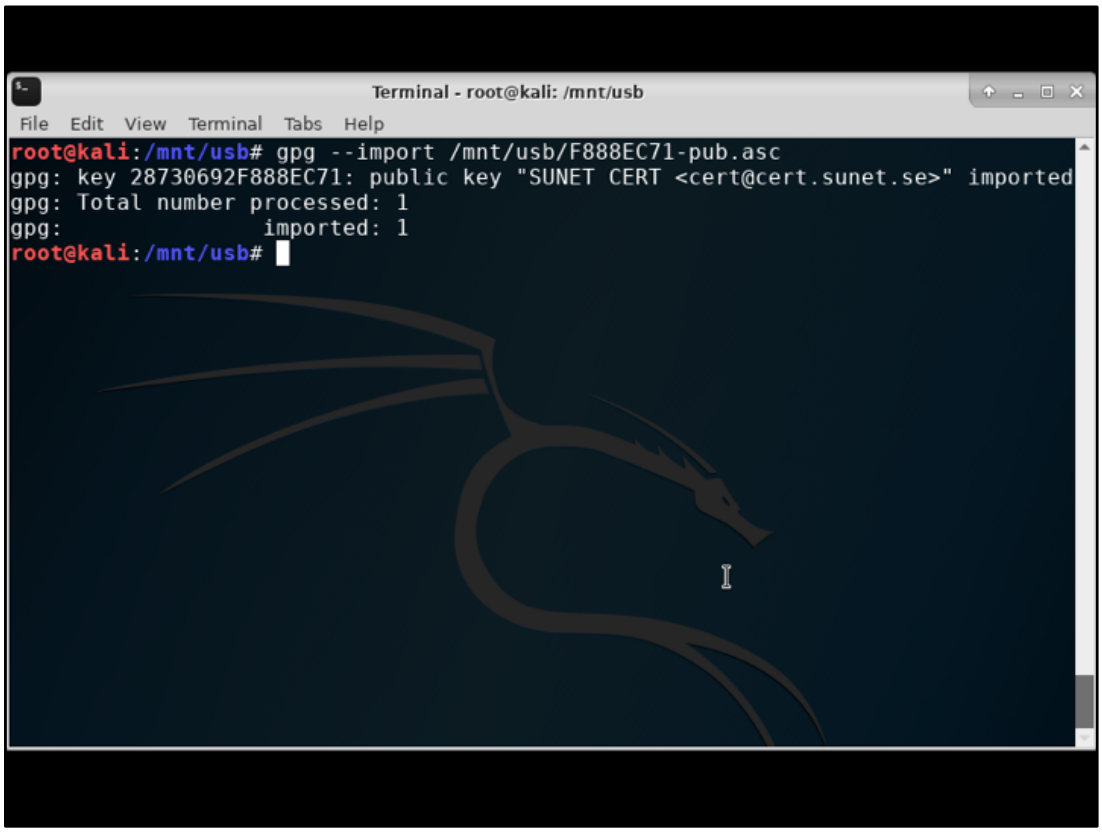
För att enkelt kunna använda USB-stickan och generera nya nycklar i framtiden skapar vi ett litet skript som kan köras för att välja rätt nyckelring.
För att använda denna nästa gång räcker det att skriva “. /mnt/usb/gpgsetup.sh”



Dags för en demonstration av hur man kan arbeta med endast subnycklar.
Sätt upp en ny GNUPG-katalog

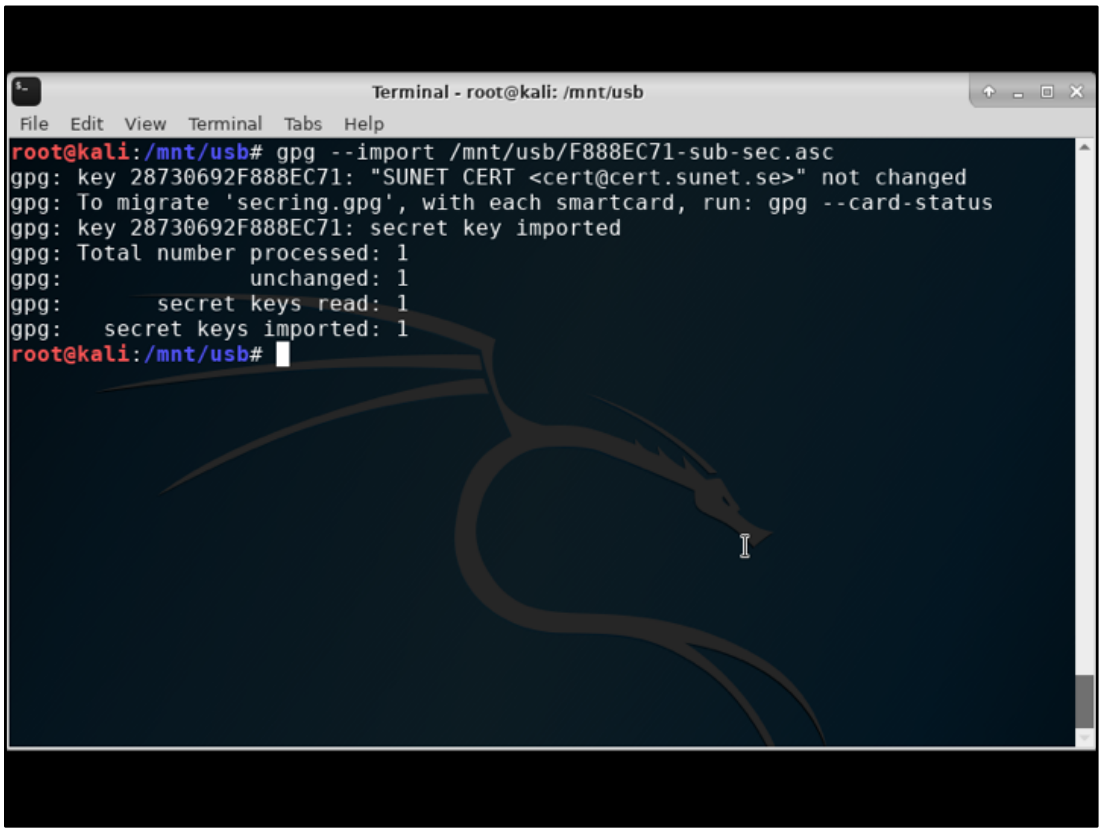
Kommando: export GNUPGHOME=$(mktemp –d /tmp/GNUPG.XXX)

Kommando: gpg –list-keys



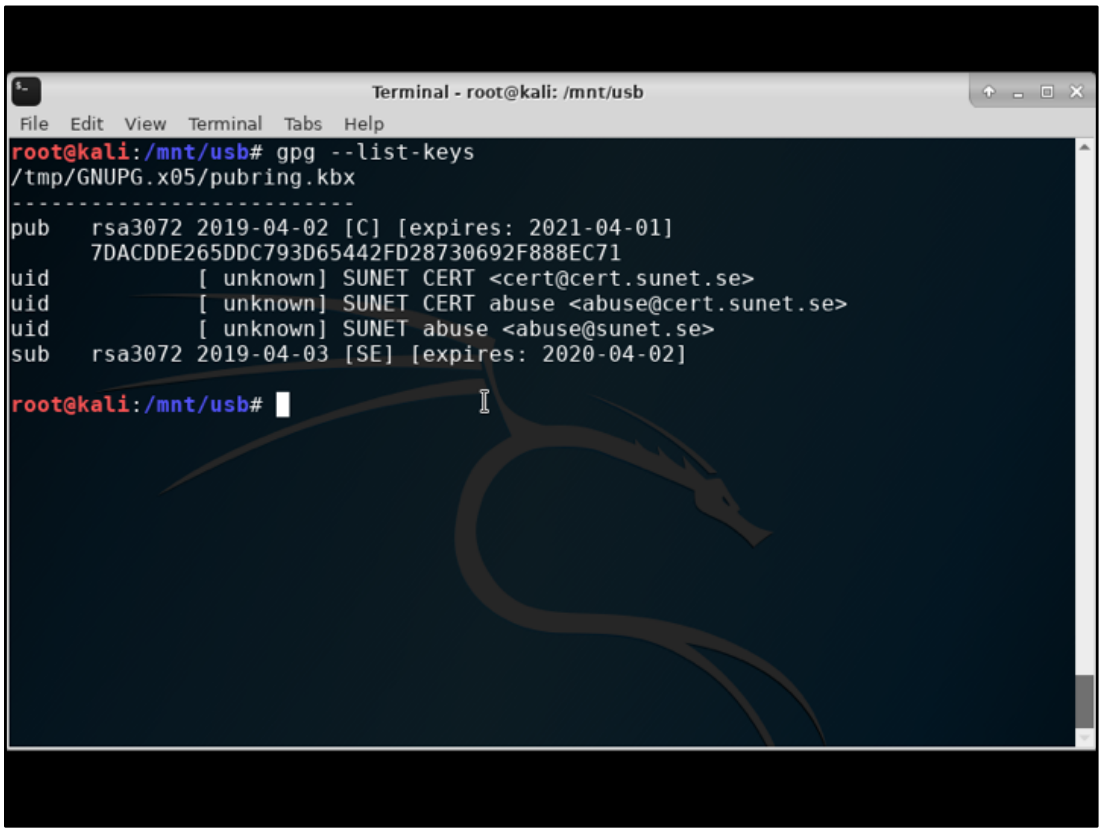
Importera den publika nyckeln

Kommando: gpg –import <path-to-public-key>

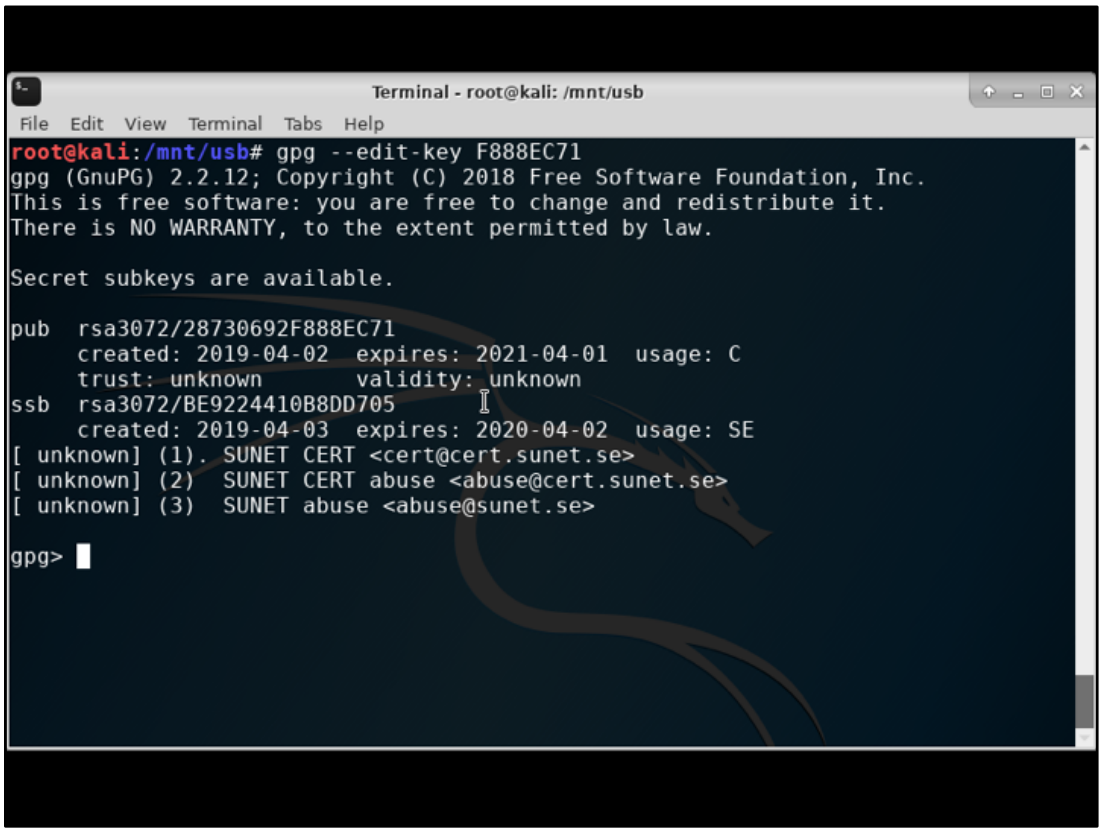


Importera de privata subnycklarna

Kommando: gpg –import <path-to-secret-subkeys>

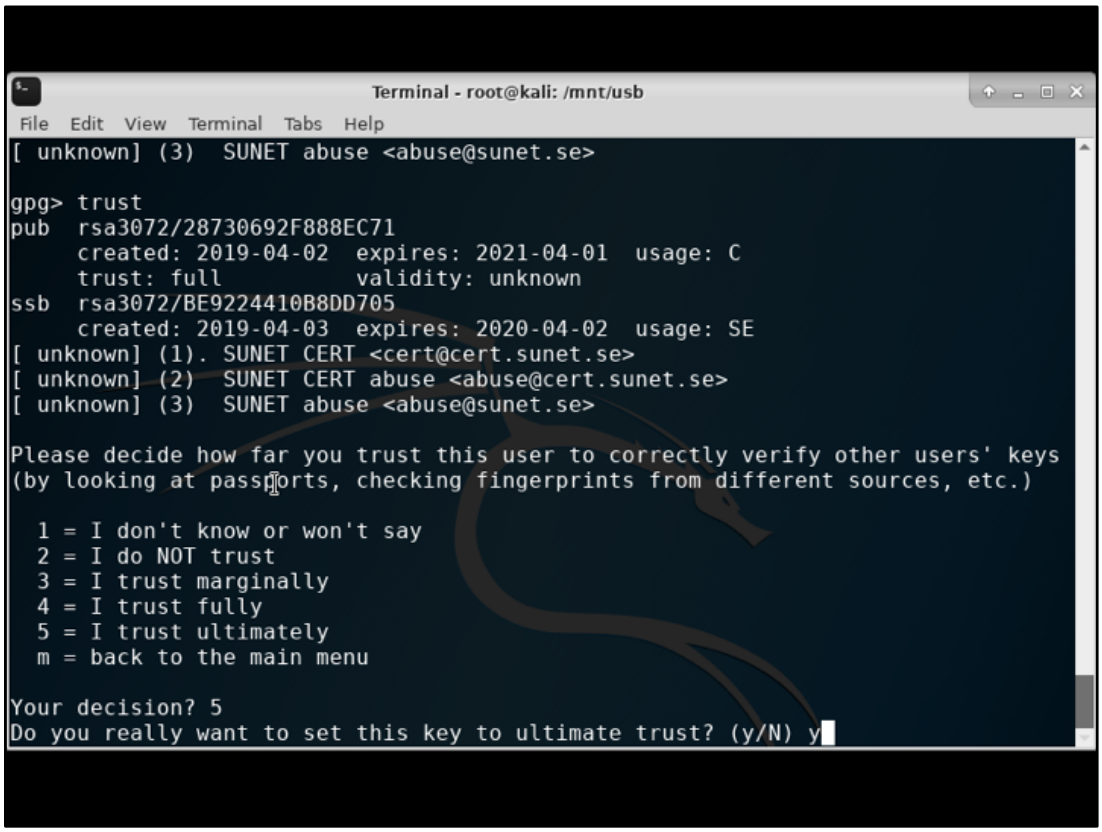


Här ser vi att de importerade nycklarna. [unknown] betyder att vi inte sagt att vi litar på dessa.



Först måste vi se till att vi litar på nycklarna.

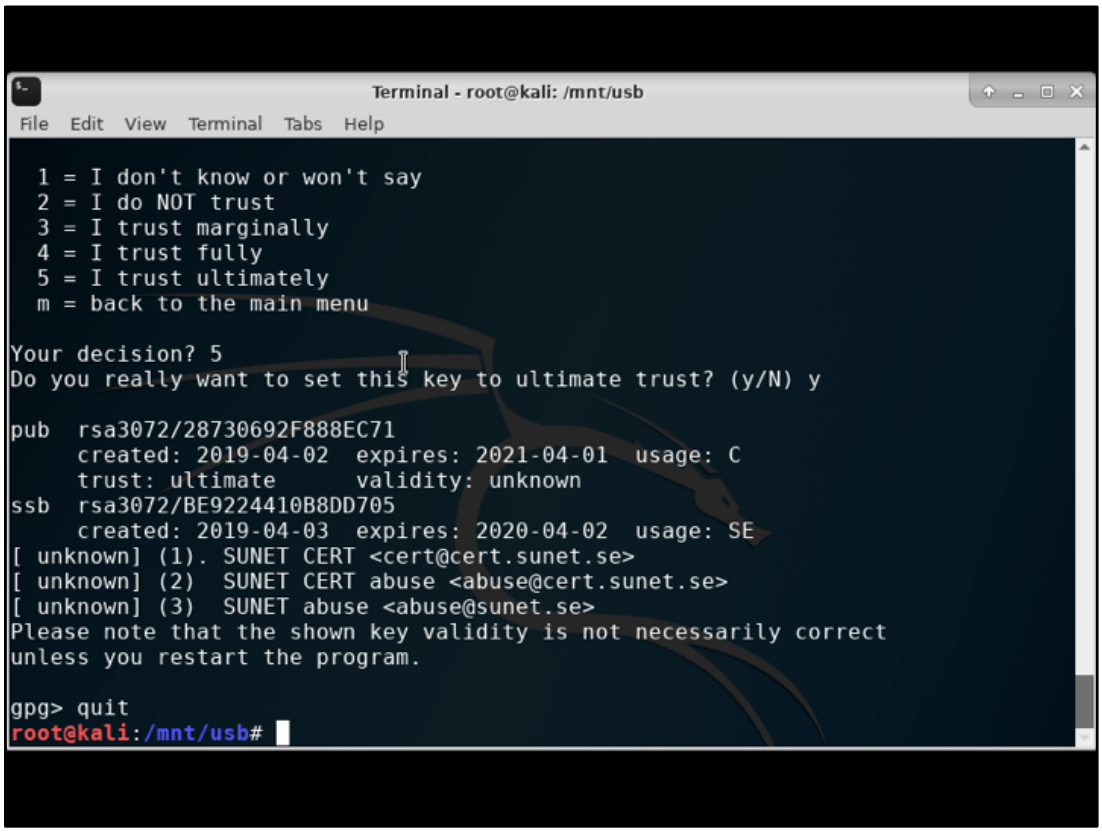
Kommando: gpg –edit-key <KEYID>



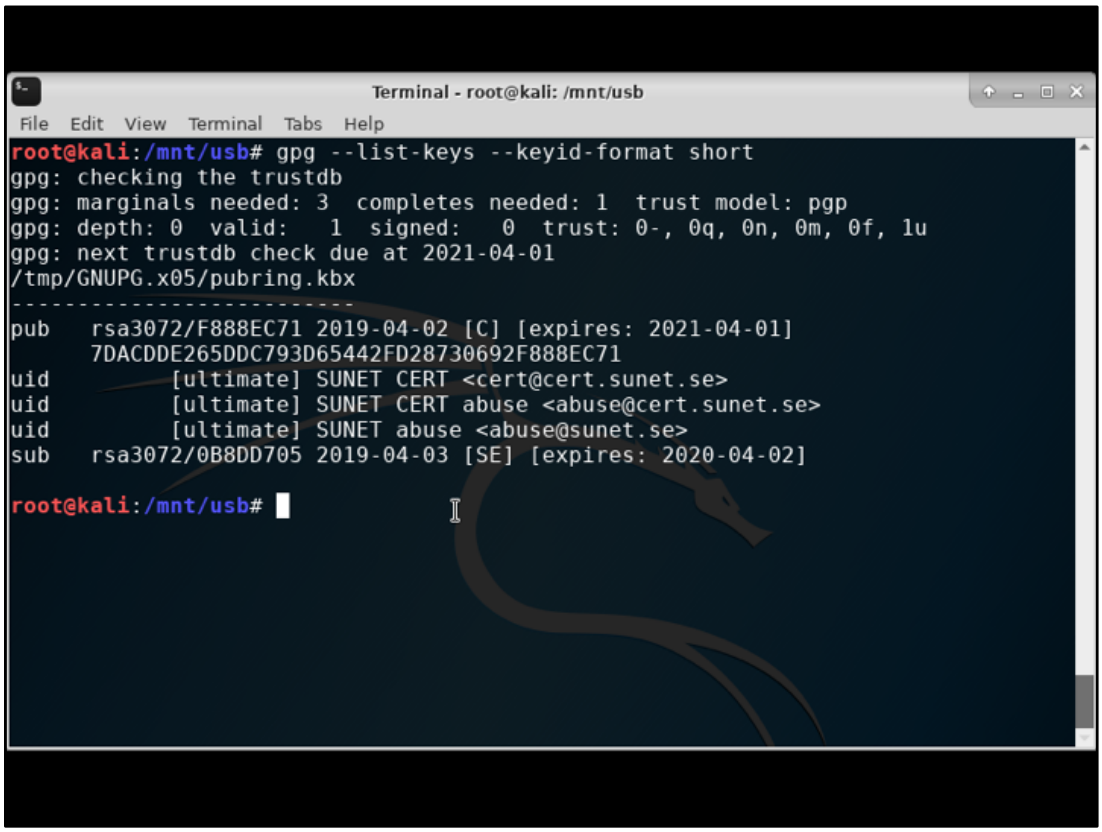
Sätt ultimate trust . Ultimate används normalt bara för dina egna nycklar så i praktiken kanske det skall vara “I trust fully” då det kan vara någon av dina kollegor som signerat.
gpg kommando: trust

Val 1: 5

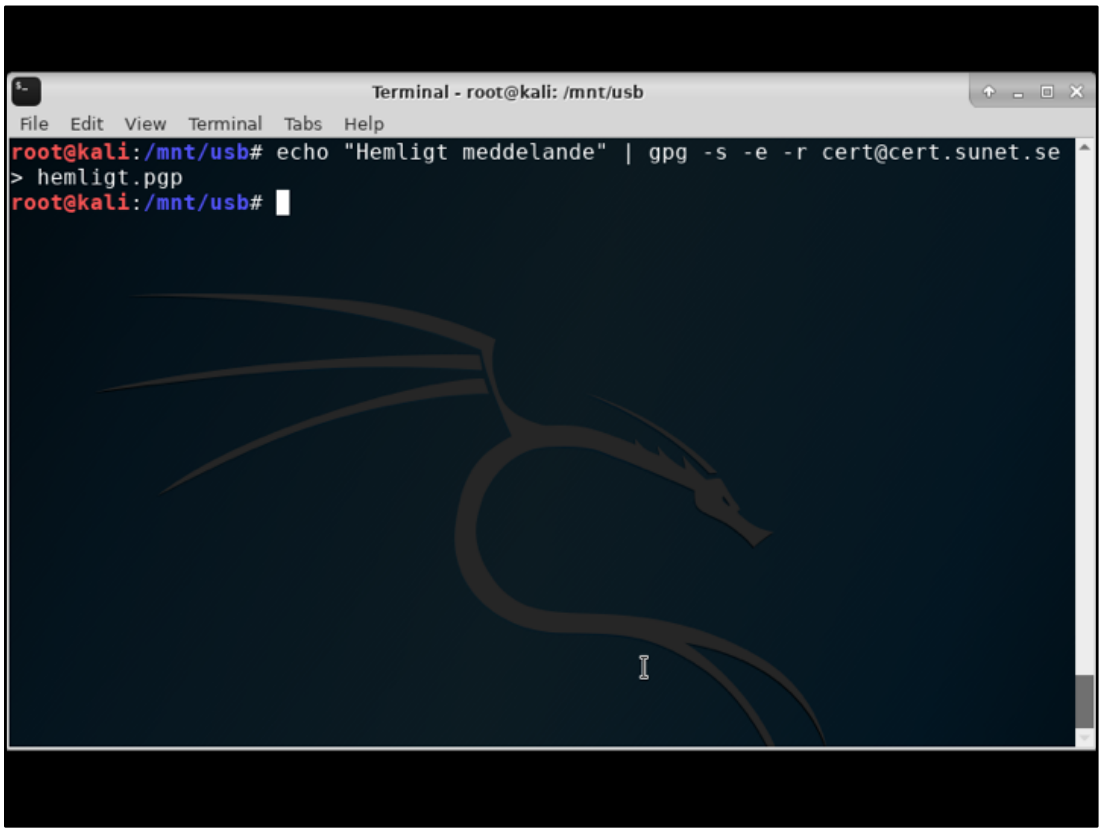
Val 2: y



Avsluta. Notera att du inte behöver spara. Anledningen är att trust sparas i en separat fil och inte påverkar själva nyckeln.



Nu ser vi att vi har [ultimate] trust på identiteterna.



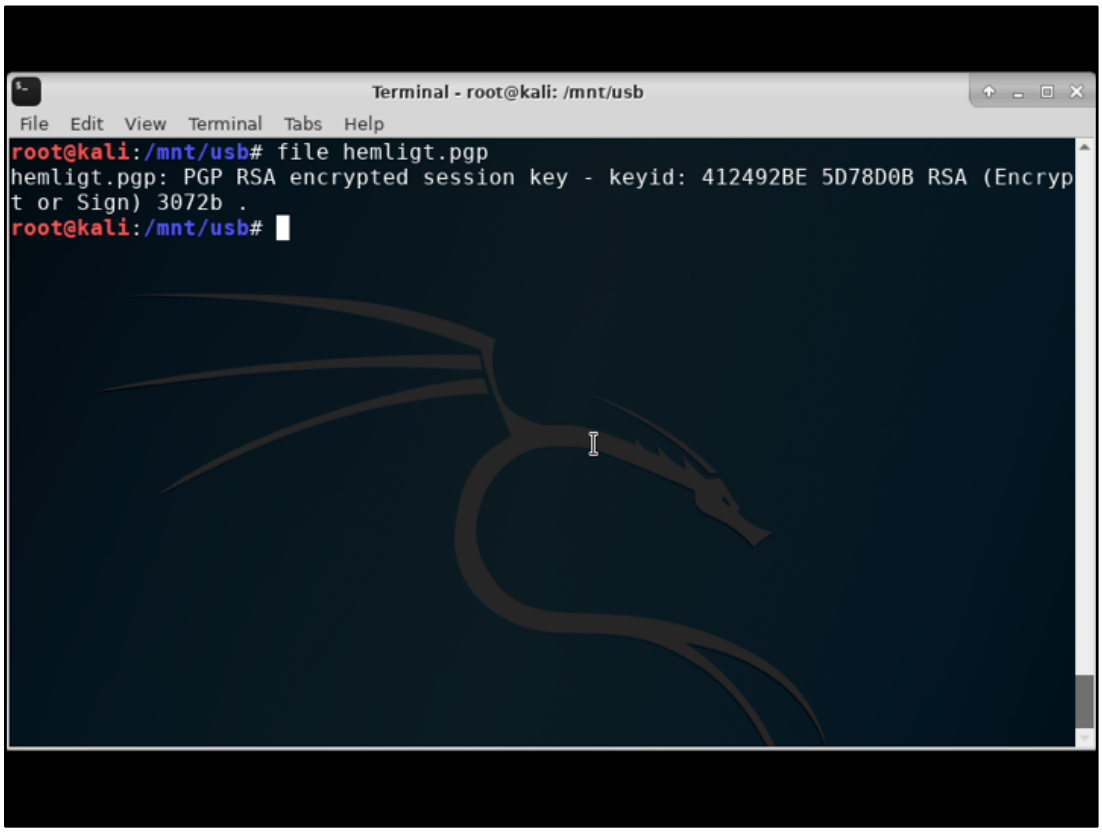
Dags att kryptera ett hemligt meddelande och spara det i filen hemligt.pgp

Ni får byta cert@cert.sunet.se mot en av de identiteter ni valt för er teamnyckel.
Kommando: gpg –s –e –r cert@cert.sunet.se

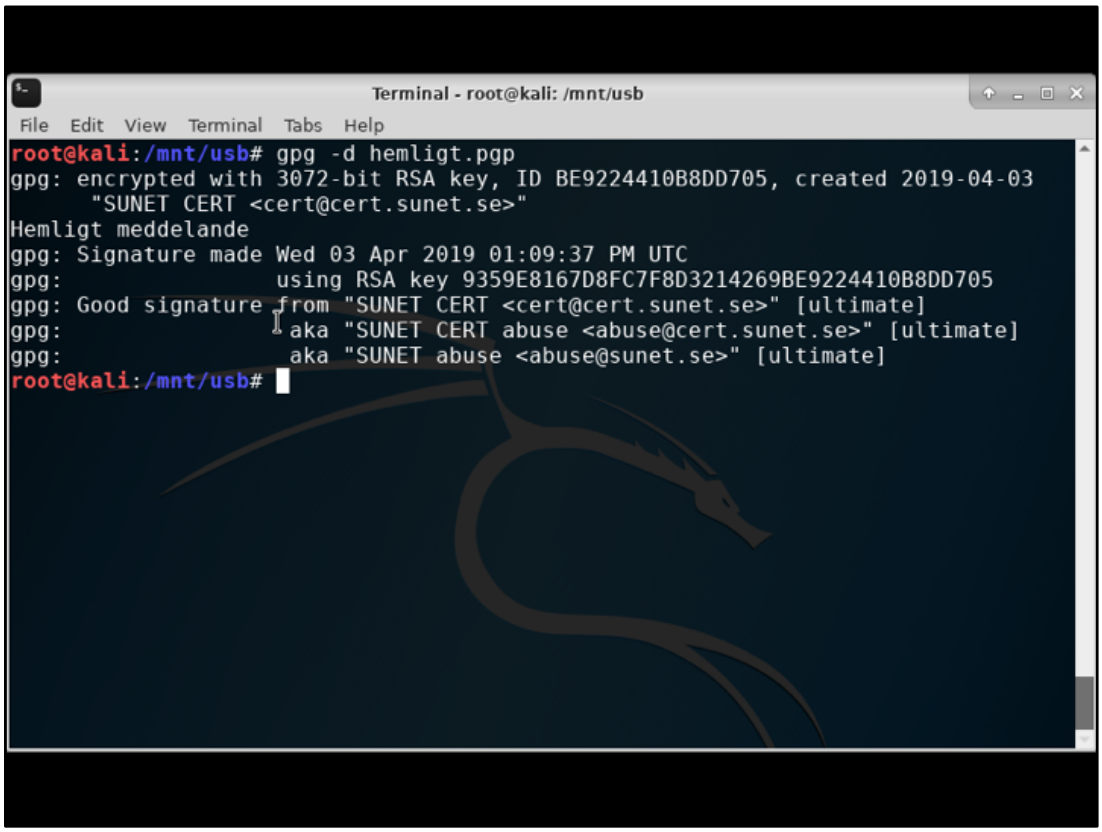
-s : Signera meddelandet

-e : Kryptera meddelandet

-r abuse@cert.sunet.se : Kryptera för denna mottagare



Här ser vi att det är en PGP RSA krypterad fil



Här ser vi att vi kan avkryptera filen med nyckeln för <abuse@sunet.se> och att den är signerad av <abuse@sunet.se>

Kommando: gpg –d hemligt.php

-d : Dekryptera