Labb G summary

First I installed gpg by opening the terminal and wrote the command **brew install gpg**
I then made sure that gpg was installed by writing **gpg --version**

2.1 Creating keys

Then I generated keys by writing **gpg --full-generate-key**
The full command let me choose different properties of the key. The user identities that were assigned to my key were my name, my email and the comment DD2395. The fingerprint of my key was: 2447 B89C A3AE 1C21 EB22 4CA8 4792 4A12 C7B5 1F96.

Subkeys are keys that are bound to a primary key pair. The purpose of the subkeys is to make key management easier. You can use a subkeys for encryption and another for signing, this mean you don't have to use your primary keys which is safer.
• What subkey(s) are attached to your key?

2.2 IMPORTING KEYS

I first downloaded the coursekey.pub from canvas and imported it to my keyring by using the import command: **gpg --import coursekey.pub**. I made sure the key was imported by using: **gpg --list-keys** to see if the key was added properly.
I then signed the key by using the command: **gpg --sign-key gpg-sign@dasak-vm-lab-server.eecs.kth.se**

To check if Sonja Buchegger's signature is on the key I would use **gpg --list-sigs buc@kth.se**
If I trust this signature I can trust that the course key is valid.

The difference between a normal and a local key wsignature is that you can't export a local signature.

2.3 SUBMITTING KEYS

To export the key I used **gpg --output public.pgp --armor --export oazrak@kth.se**

I then imported the signed key with **gpg --import signed-publig.pgp**

To see all signatures on my key I use: **gpg --list-keys**

The reason I would like to export my private key is if I would like to switch machine.

2.4 AUGMENTING IDENTITIES

- list all secret keys: **gpg --list-secret-keys --keyid-format=long**
- Remember which key you want to add identity: 501BB5417FD61F04
- edit key: **gpg --edit-key 501BB5417FD61F04**

- Enter gpg> **adduid** to add the user ID details.
 a) name: oscarazrak
 b) epost: oscar@azrak.se
 c) comment: DD2395
- enter gpg> **save**

- export by using: **gpg --armor --export**

To list all identities **gpg --list-keys *keyid***

2.5 SIGNING KEYS

Låt annan signera din key

- List my keys: **gpg --list-keys**

- export the key I want signed, **gpg --output public.pgp --armor --export oazrak@kth.se**

- friend imports my key till his keyring: **gpg --import *filename***

- friend signs my key: **gpg --sign-key oazrak@kth.se**

- friend exports the key I want signed, **gpg --output public.pgp --armor --export oazrak@kth.se**

- I import key: **gpg --import *FILENAME***

The system should trust my key after it has my signature, even more so given that the system is my friend's system

I should however always compare the fingerprint before signing my friend's key because if I don't then it could mean that I sign someone else's key which could damage my worthiness.

3.1 MESSAGE SIGNING

I first copied and pasted the sign messages in different files. Then in the terminal I ran the command: **gpg --verify *FILENAME***. I then checked for the correct messages and saved the hash in different file. I ran the command: **--output cmsgout.txt --armor --sign cmsg.txt**

If a message is correctly signed, then the sender used their private key for signing the message. I can verify who the sender is since their private key is connected to their public key which is accessible to me.

I used my private key to sign the messages so the receiver can use my public key in order to verify it.

3.2 MESSAGE ENCRYPTION

I first copied and pasted the sign messages in different files. Then in the terminal I ran the command: **gpg --decrypt *FILENAME***. I then checked for the correct messages and saved the hash in different file. I ran the command: **gpg --output out.pub --armor --encrypt --recipient gpg-crypt@dasak-vm-lab-server.eecs.kth.se --recipient oazrak@kth.se corrcmsg.txt**

I used my private key to decrypt the message, this means that I am the only one that is able to read the message since I am the only one with access to my private key. The encryption was done with my public key, and decryption with my private key.

The key that I used to make the reply confidential was the receiver's public key which means that the only way I could decrypt the message is to have access to the receiver's private key paired with the public key used for encryption. The command ran is the long one above, which means I am, and the receiver, are added as recipients. This means, to decrypt the message, one must have either mine or the receiver's private key.

3.3 MESSAGE SIGNING AND ENCRYPTION

I first copied and pasted the sign messages in different files. Then in the terminal I ran the command: **gpg --decrypt *FILENAME***. I then checked for the correct messages and correct signatures and saved the hash in different file. I ran the command: **gpg --output out.pub --armor --encrypt --sign --recipient gpg-both@dasak-vm-lab-server.eecs.kth.se bcorrectmsgs.txt**

A message that is both correctly signed and encrypted has been signed by my private key and the recipients can use my public key to verify my signature. The receiver needs their public key to encrypt a message correctly and their private key is needed to decrypt the message.

I used my private key to sign the message and their public key to encrypt the message.