

BTS – MBDS – Big Data Security

Assignment 2:

- 1.- GPG – Key generation
- 2.- Role A: Encrypt / Role B: Decrypt
- 3.- Sign

01 March, 2021

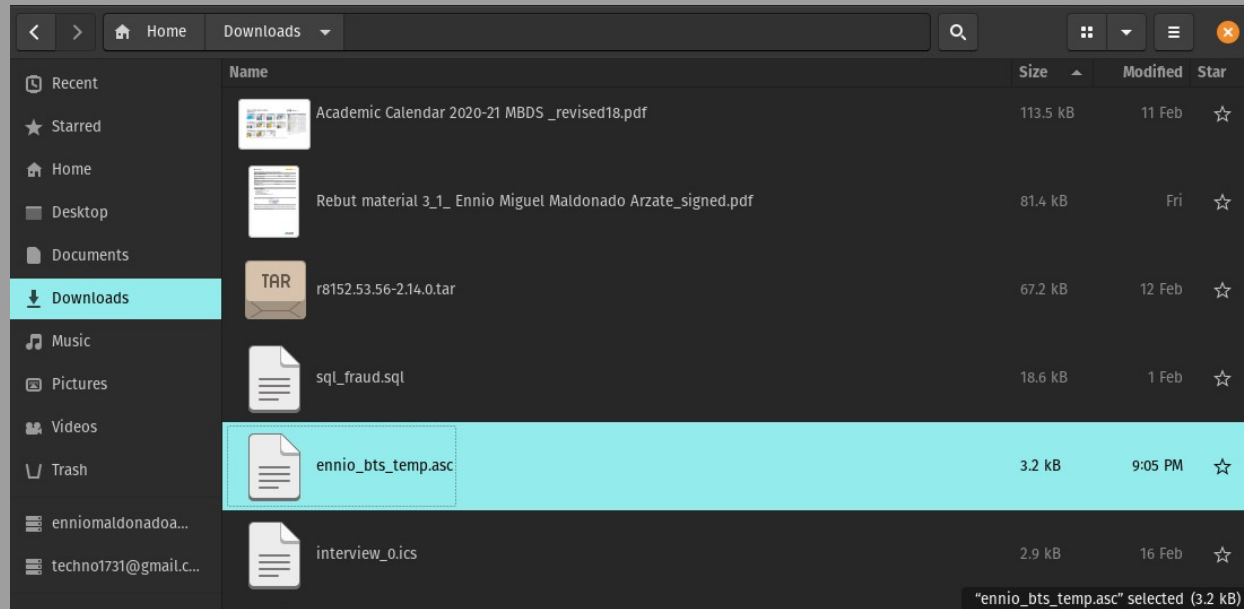
1: GPG – Key generation

```
> gpg --full-generate-key
```

```
techno@pop-os:~  
> gpg --full-generate-key  
gpg (GnuPG) 2.2.20; Copyright (C) 2020 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)  
  (14) Existing key from card  
Your selection? 1  
RSA keys may be between 1024 and 4096 bits long.  
What keysize do you want? (3072) 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 Wed 03 Mar 2021 09:03:04 PM CET  
Is this correct? (y/N) y  
  
GnuPG needs to construct a user ID to identify your key.  
  
Real name: Ennio  
Email address: ennio.maldonado@bts.tech  
Comment: assigment disposable key  
You selected this USER-ID:  
    "Ennio (assigment disposable key) <ennio.maldonado@bts.tech>"  
  
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?  
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o  
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
```

1: GPG – Key generation

```
> gpg --armor --export ennio.maldonado@bts.tech > ennio_bts_temp.asc
```



2: Role A – Encrypt - Decrypt

```
> gpg --import khalounkey.asc
```

```
> gpg -k
```

```
/home/techno/.gnupg/pubring.kbx
-----
pub  rsa4096 2019-09-18 [SC] [expires: 2023-09-17]
     FB5DB77FD5C118B80511ADA8A6310ACC4672475C
uid  [ unknown] AWS CLI Team <aws-cli@amazon.com>

pub  rsa4096 2021-03-02 [SC] [expires: 2021-03-03]
     29A09EB36C8ECEDCF7BE4B3B580FAB2056E96CB5
uid  [ultimate] Ennio (assignment disposable key) <ennio.maldonado@bts.tech>
sub  rsa4096 2021-03-02 [E] [expires: 2021-03-03]

pub  rsa3072 2021-03-02 [SC] [expires: 2023-03-02]
     68D4E7154C58AC2F12FA344014618DC86DEC178F
uid  [ unknown] khaloun <k.alnaierat@gmail.com>
sub  rsa3072 2021-03-02 [E] [expires: 2023-03-02]
```

2: Role A – Encrypt

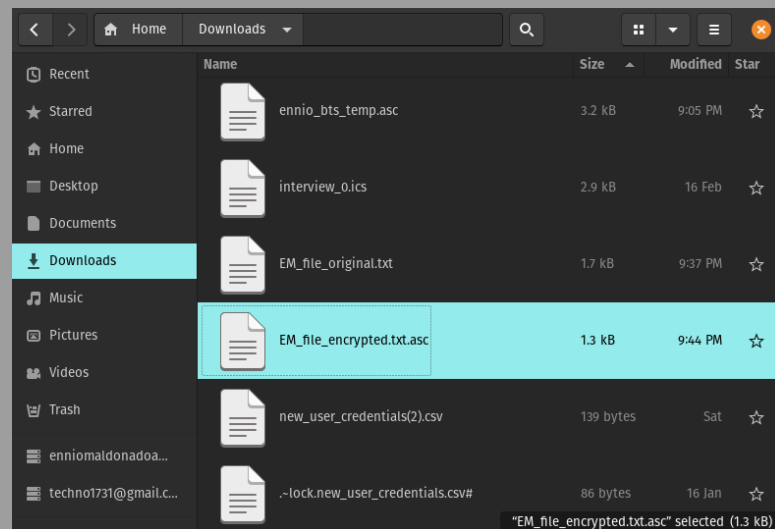
```
> gpg --encrypt --recipient 'k.alnaierat@gmail.com' --output EM_file_encrypted.txt.asc EM_file_original.txt
```

```
gpg: 930F0C2DDC81FFA9: There is no assurance this key belongs to the named user
```

```
sub rsa3072/930F0C2DDC81FFA9 2021-03-02 khaldoun <k.alnaierat@gmail.com>
Primary key fingerprint: BBA6 71C1 67DD 9DE5 1A53 D203 C401 0013 6195 AE9D
Subkey fingerprint: 275D 128A DA3E CE14 7FC9 1644 930F 0C2D DC81 FFA9
```

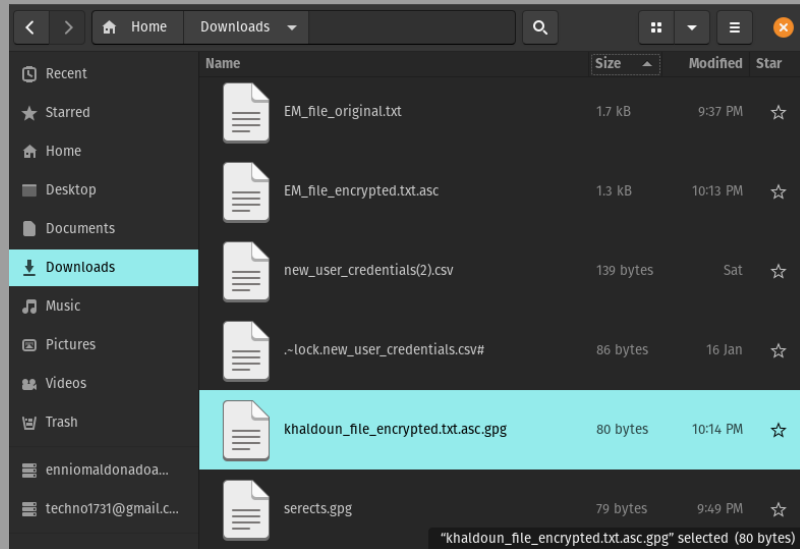
It is NOT certain that the key belongs to the person named in the user ID. If you **really** know what you are doing, you may answer the next question with yes.

```
Use this key anyway? (y/N) y
```

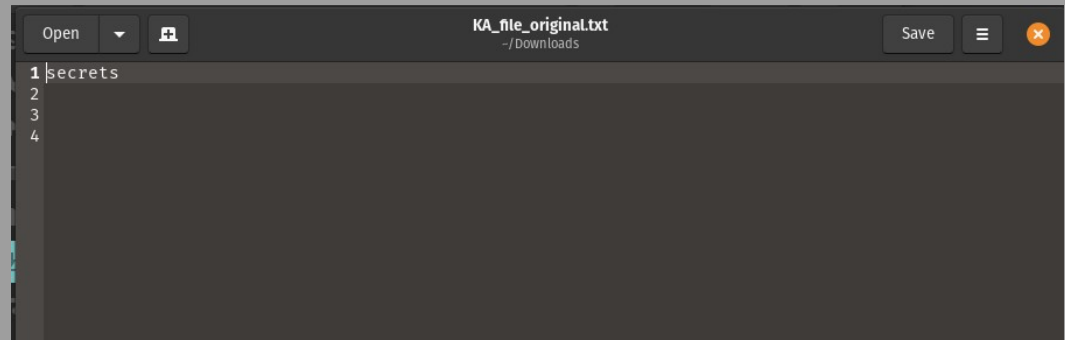


2: Role B – Decrypt

```
> gpg --decrypt khaloun_file_encrypted.txt.asc.gpg > KA_file_original.txt  
gpg: AES256 encrypted data  
gpg: encrypted with 1 passphrase
```



```
> cat KA_file_original.txt  
secrets
```



3: Sign Role A

```
> gpg --sign --default-key ennio.maldonado@bts.tech EM_file_original.txt
```

```
> mv EM_file_original.txt.gpg EM_file_SIGNED.txt.gpg
```

```
> gpg --verify EM_file_SIGNED.txt.gpg
```

```
gpg: Signature made Wed 03 Mar 2021 07:48:33 AM CET
```

```
gpg:          using RSA key 29A09EB36C8ECEDCF7BE4B3B580FAB2056E96CB5
```

```
gpg:          issuer "ennio.maldonado@bts.tech"
```

```
gpg: Good signature from "Ennio (assignment disposable key) <ennio.maldonado@bts.tech>" [ultimate]
```

3: Sign Role B

```
> gpg --verify --default-key k.alnaierat@gmail.com KA_file_original.txt.gpg
```

```
gpg: Signature made Wed 03 Mar 2021 11:35:25 AM CET
gpg:          using RSA key 42B3E427895AE141DE297C2FF2A01524B7C74C18
gpg:          issuer "k.alnaierat@gmail.com"
gpg: Good signature from "khaloun <k.alnaierat@gmail.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 42B3 E427 895A E141 DE29 7C2F F2A0 1524 B7C7 4C18
```