ΥΣ13 Computer Security
PGP

Intro

How does cryptography work?

A *cryptographic algorithm,* or cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a *key* — a word, number, or phrase — to encrypt the plaintext. The same plaintext encrypts to different ciphertext with different keys.

The security of encrypted data is entirely dependent on two things:

- the strength of the cryptographic algorithm
- the secrecy of the key

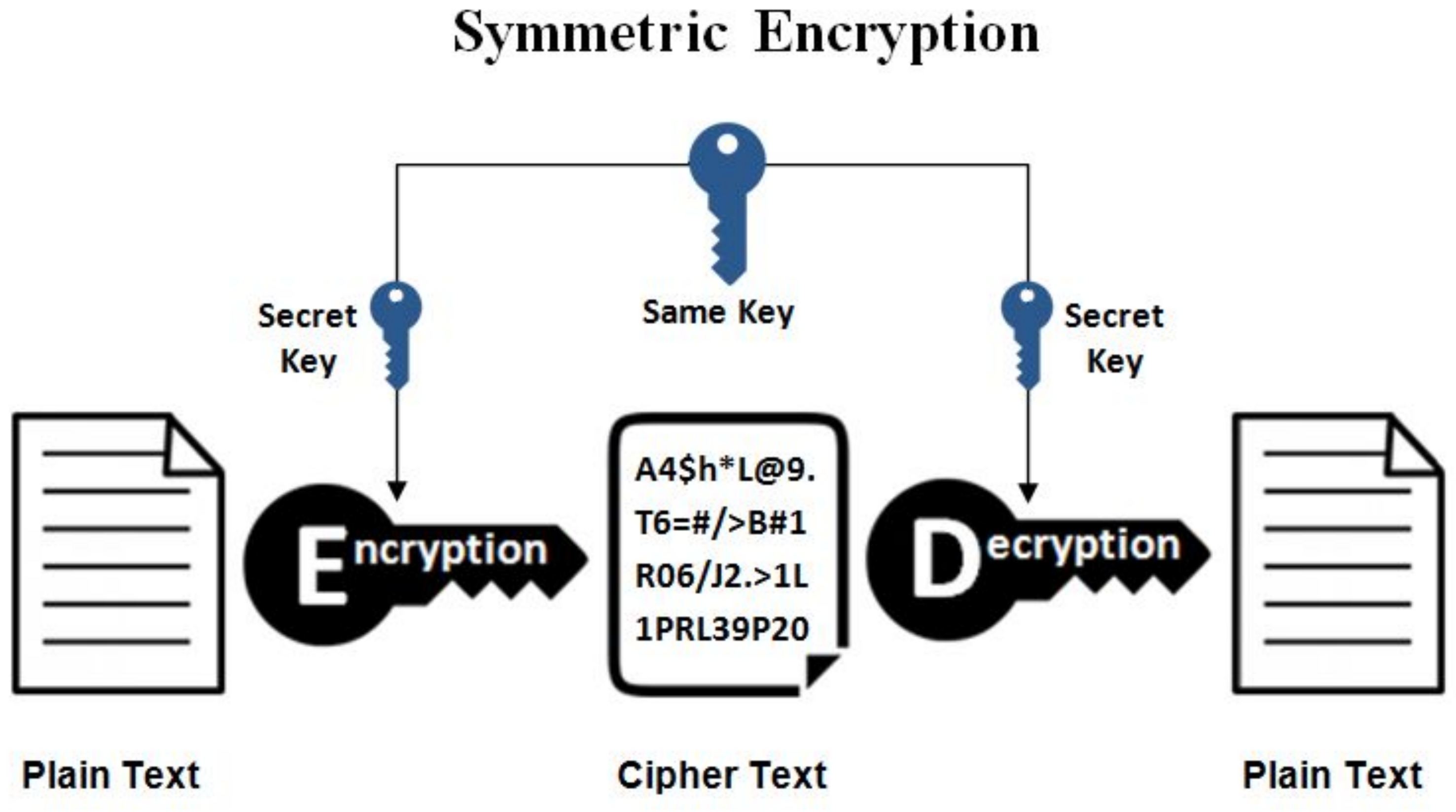
Symmetric vs Asymmetric Encryption

- Symmetric encryption uses a single key that needs to be shared among the people who need to receive the message while asymmetrical encryption uses a pair of public key and a private key to encrypt and decrypt messages when communicating.

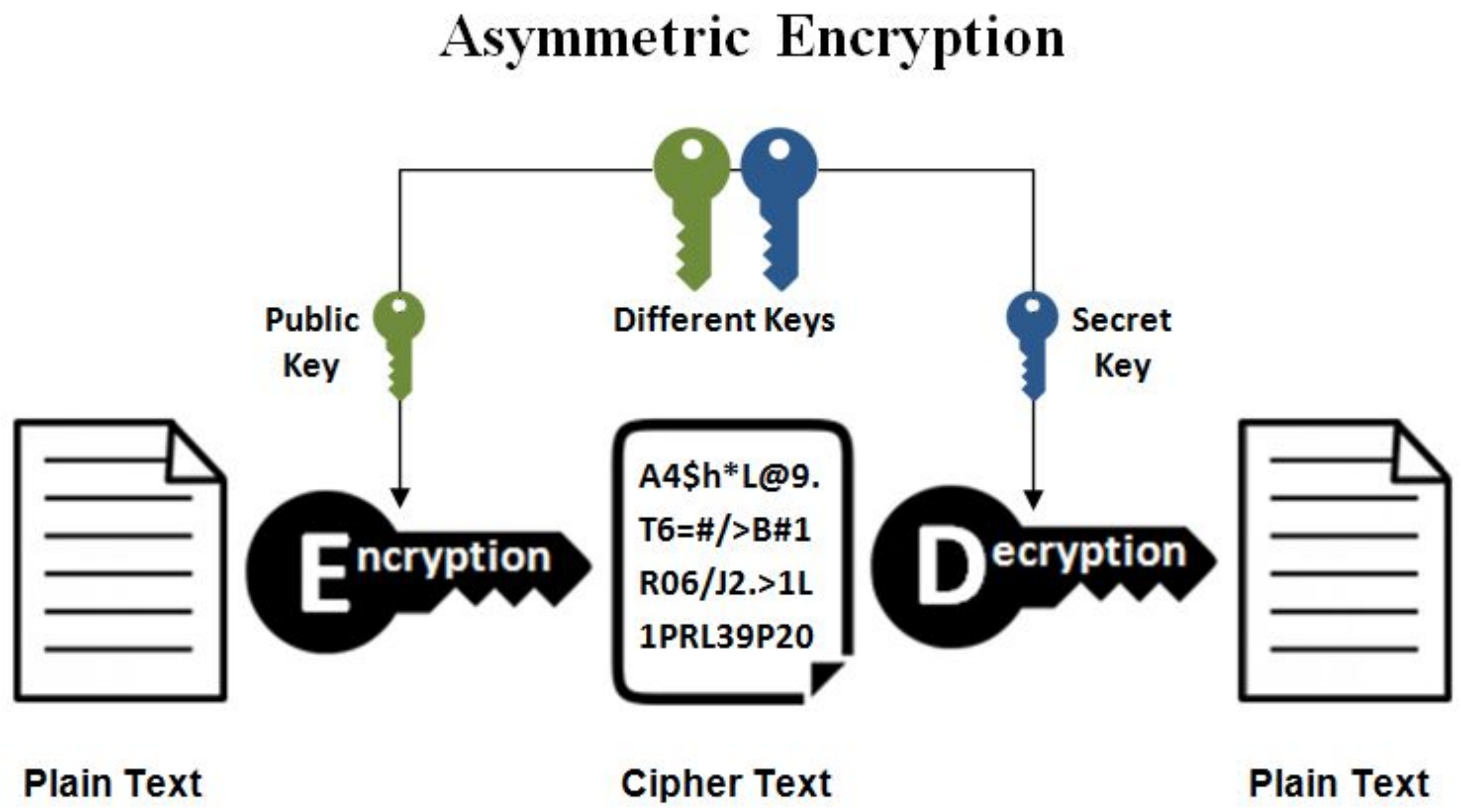
- Asymmetric encryption was introduced to complement the inherent problem of the need to share the key in symmetrical encryption model, eliminating the need to share the key by using a pair of public-private keys.

- Asymmetric encryption takes relatively more time than the symmetric encryption.

Symmetric vs Asymmetric Encryption



Symmetric vs Asymmetric Encryption



Symmetric vs Asymmetric Encryption

Symmetric Encryption Algorithms

- Blowfish, AES, RC4, DES, RC5, RC6
- Most commonly used AES-128, AES-192, and AES-256

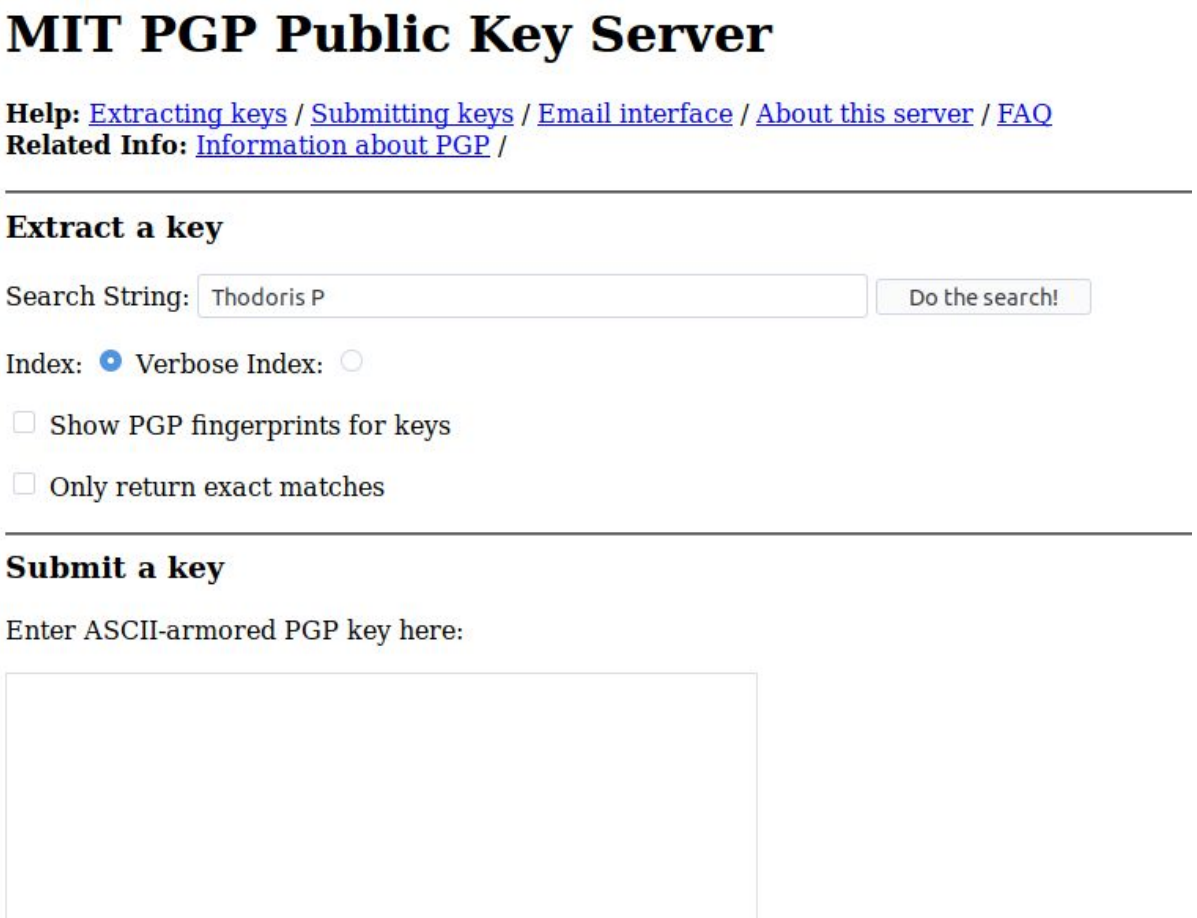
Asymmetric Encryption Algorithms

- ElGamal, RSA, DSA, Elliptic curve techniques, PKCS, Diffie-Hellman

What is PGP ?

- Created by Phil Zimmermann on 1991.
- PGP uses a **private-key** that must be kept secret and a **public-key** that sender and receiver must share.
- **GPG** (**Gnu Privacy Guard**) is an independent implementation of the OpenPGP standards.
- Stores pubic-keys on public key servers (https://pgp.mit.edu/)
- Other uses:
  - Web of trust
  - Digital signatures
  - A digital certificate contains the user's identifying information, their public key and one or more digital signatures.
  - Digital Certificates

PGP



MIT PGP Public Key Server

Help: Extracting keys / Submitting keys / Email interface / About this server / FAQ
Related Info: Information about PGP /

Extract a key

Search String: Thodoris P   Do the search!

Index: [✓ ●] Verbose Index: ○

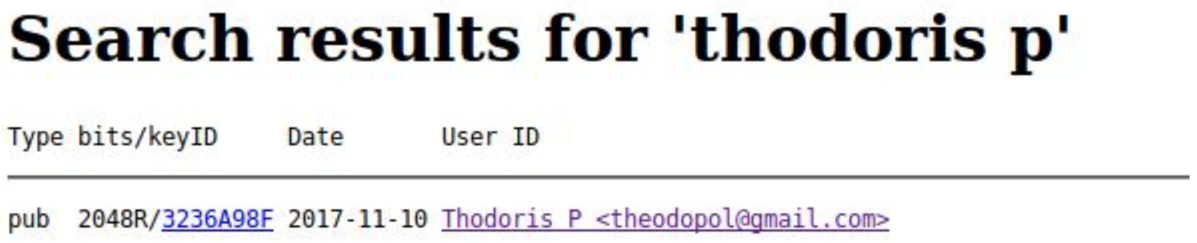
☐ Show PGP fingerprints for keys

☐ Only return exact matches

Submit a key

Enter ASCII-armored PGP key here:

PGP



Search results for 'thodoris p'

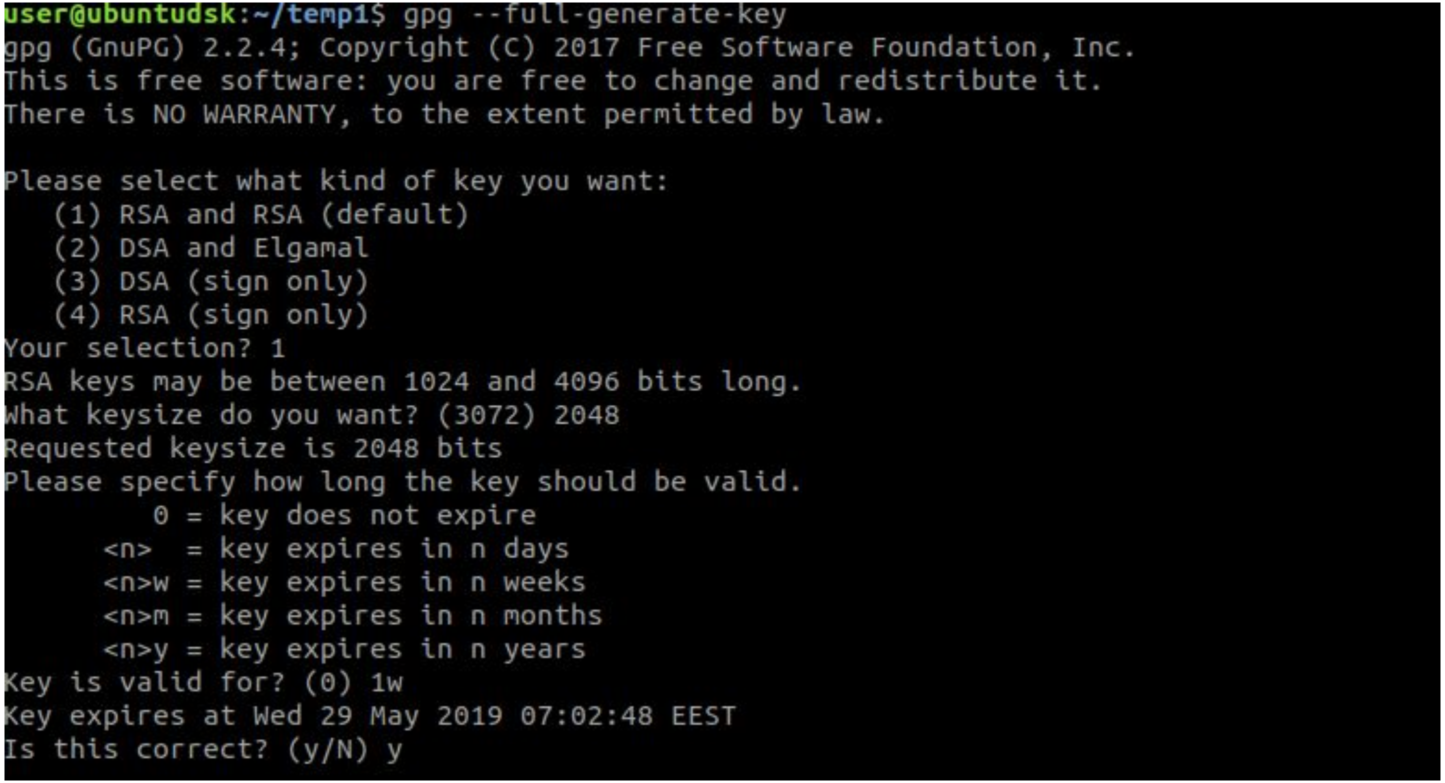
```
Type bits/keyID     Date       User ID
pub  2048R/3236A98F 2017-11-10 Thodoris P <theodopol@gmail.com>
```

PGP

```
-----BEGIN PGP MESSAGE-----

hQEMA1uJ1V0RFqsgAQf+MilDkqcrn4Ay+ik+GjV05K6ohBKrk4gX89cJ9N53quMJ
Ly299Ti7h9wqM/dSVQkzxm9/TQdTU6FbCm/kn6JyyGuMigV98NSrPfXGwPp5pCuW
HPtkJjbM/ZqoAKwY0tGkXWIJvZ4j8NNX0zket9v/Ncf+JaEf+CdZllcz1Qmwl87q
FcQ6Rud2A2jpJlkfMNTJ1ylXflE59PYiQNm90q7RgeHQcAzV2yZYzfFOwVDmKUe1
bMPo/pV0Ienhnix9zouAHhISXJZs0ZYpuSp9pZi98eHnXJLZcR+pMI5fn/t9H1kN
yJmMF4VW2SjNxfAzD3eYq7BPhITmA7kpH0R3cZyz/4UBDAOK7moiOSK7qQEH/33N
jP9Mrt2F5QkiiitzsMsWA0DvFopvyKVRX3lZTnvPBYyvSsfUVz3sMhHJ0gf/6EPX
NU0r1JohMMZ+NKRaIN1iRmg3bktzR/crkTFTtRO4r3H3aFKjxNv+bT14t47ArUes
ZUb6fQfi1iGiyrN3ScYC6rG6yQhhw8QaMNqbheeazga5ECr9xDx8s5UTZS46RGjV
c/RFTUqwt+4U4ZkbCHSrWnovs/gh3JV9g3Cot0YBje4ArhlK7d0JDST0KIKdfU5E
35OmFVSG/z4PmftB8kb2CCTf80dvBXR9+kU3p+QgHP/ePEpgL7/m0Znz7pvjRxIM
TcCqsfbCKvPcjPiGwRLS6QGpJj+rb9Mkv5CmSme07EBrI0mm7aofQwpf+szbVZvS
by0q8jOgfjnxefDOFNo0U0uGEOroY81Qebl3hbt0dVg6Zb+eKjeH1g1oA3QrQRbx
lfsl98d8JNYVShf5MBcLLwP8gNpZSn1LcpQ5x6cCa5jLpU6xqClrIqj9bmERcBBP
0ZyFPKYvaODn/058enr+zM2slIJMXfyvEtX+cgEJ44z+cP65BBZnpUEfOT5txjNv
gJj7qt0PKBv2sUaKTtz3p8xO6LOmtqUKhRCV3k5K7OWROOSTkLm17TuUgd1wcjSY
zg9B4HIXBa2ZsJ0QdkWZEeTAMO+dwENBxrxGUCX9Zr2yzueP7rKalLUqDwA0+12w
NpN42aFcKdjwbFAUBPPSXRXqxAP2Th9xyQ5cGmGwglmYrnrj10+4nXRMQ1E0maGi
Fw9Y2jzmQactcDGsQ6QT2dz/FuTsNKIOVV1OdKQyTTdk53YsgceQzWjIE8+k/lYD
TNe2tvHpjlLURUbM1CRZGqRQ3rN02I+b0GbCt7WkQh99ebdCYsSaS8ta2FvFO1On
jz1wMaJjZhbxRrs5/gC8lbWm9vBOPxyMzSJkv8lCQSB8T/xO0HQ1i8rVK4lmjQ7t
U+qIg/oBqQCbtiFRfTL+CX7SOYC2YaKfKzV75mliBFrmo/mqGJ6YRJiFH1EqecHh
wEoePdzzxYtkYzILWiQQO53VQpTaqVkSSdEKNqIrs9+ZMY4KVWmRO73SUCOBzyA0
t/iKPanmP3vQFPBxysKjpU046B4JOs9ocvL8pd0f9kWu2Hp+PXvkYIAYXCgNuTzW
yfceGa1JieUzfR5tfHqSwRTWIJyJfbUSyUswPrHMIkFCGoMZQ5a6C6djhwo0KP45
0p8QvYuKNofyFnZ5/NK74zWx5/UkwSLm3qIa9ggCUpxfNKu/zabeaISn1TzbTNtW
5RMfEFOdC/t3ROI//NX4G3YPAMQUg1YtKy2y/oD5jcdHfZu12iywjmYJFdectmDK
JWKK2cvyBOwYUkjC9woN2+WVdh2U/w/fJIc/Aw==
=KrYw
-----END PGP MESSAGE-----
```

PGP

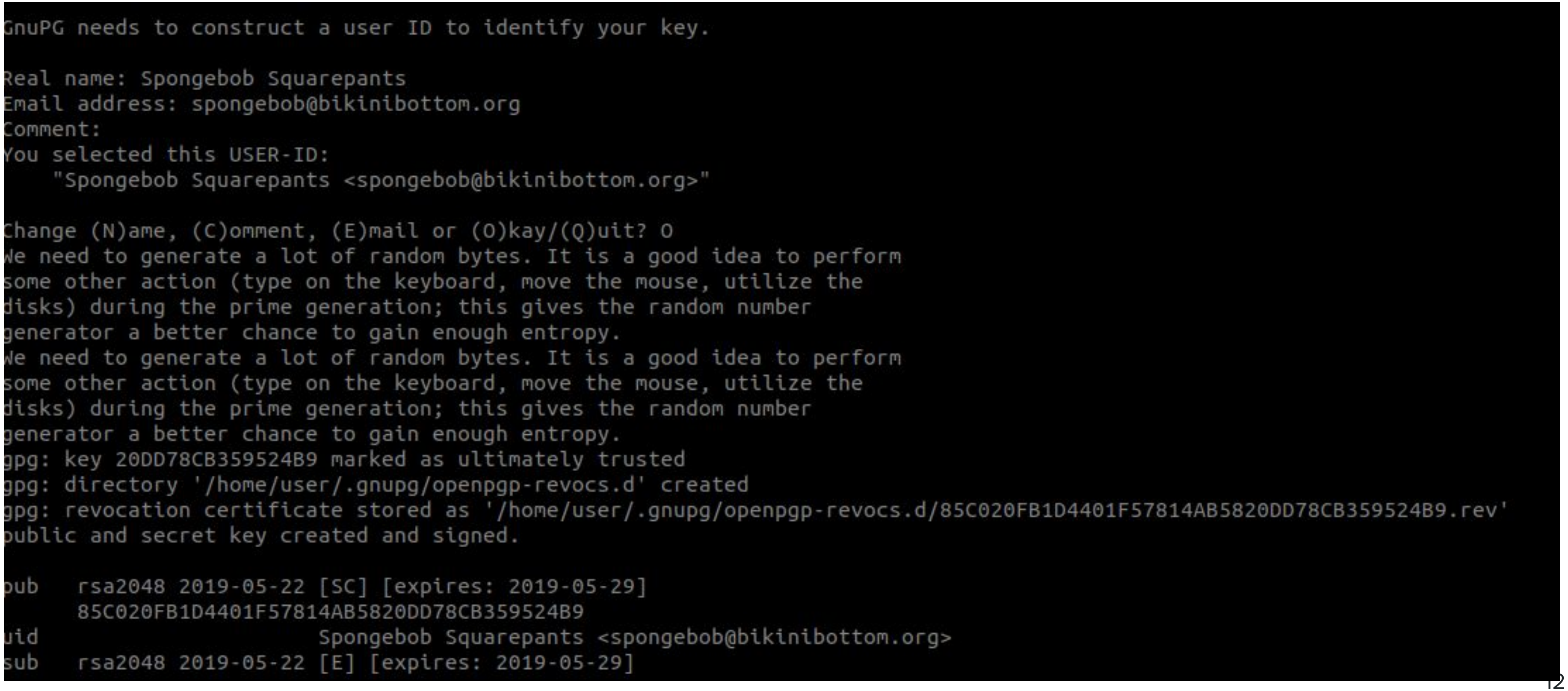
gpg --full-generate-key



```
user@ubuntudsk:~/temp1$ gpg --full-generate-key
gpg (GnuPG) 2.2.4; Copyright (C) 2017 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)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 2048
Requested keysize is 2048 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) 1w
Key expires at Wed 29 May 2019 07:02:48 EEST
Is this correct? (y/N) y
```

PGP

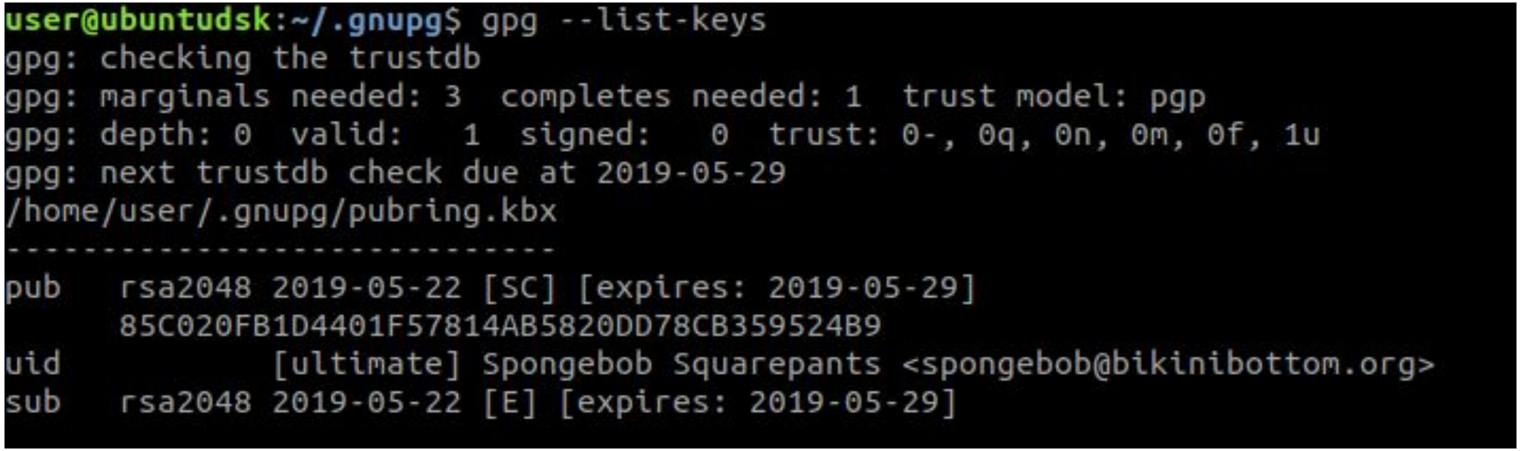
gpg --full-generate-key



```
GnuPG needs to construct a user ID to identify your key.

Real name: Spongebob Squarepants
Email address: spongebob@bikinibottom.org
Comment:
You selected this USER-ID:
    "Spongebob Squarepants <spongebob@bikinibottom.org>"

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
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
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.
gpg: key 20DD78CB359524B9 marked as ultimately trusted
gpg: directory '/home/user/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/user/.gnupg/openpgp-revocs.d/85C020FB1D4401F57814AB5820DD78CB359524B9.rev'
public and secret key created and signed.

pub   rsa2048 2019-05-22 [SC] [expires: 2019-05-29]
      85C020FB1D4401F57814AB5820DD78CB359524B9
uid                      Spongebob Squarepants <spongebob@bikinibottom.org>
sub   rsa2048 2019-05-22 [E] [expires: 2019-05-29]
```

PGP

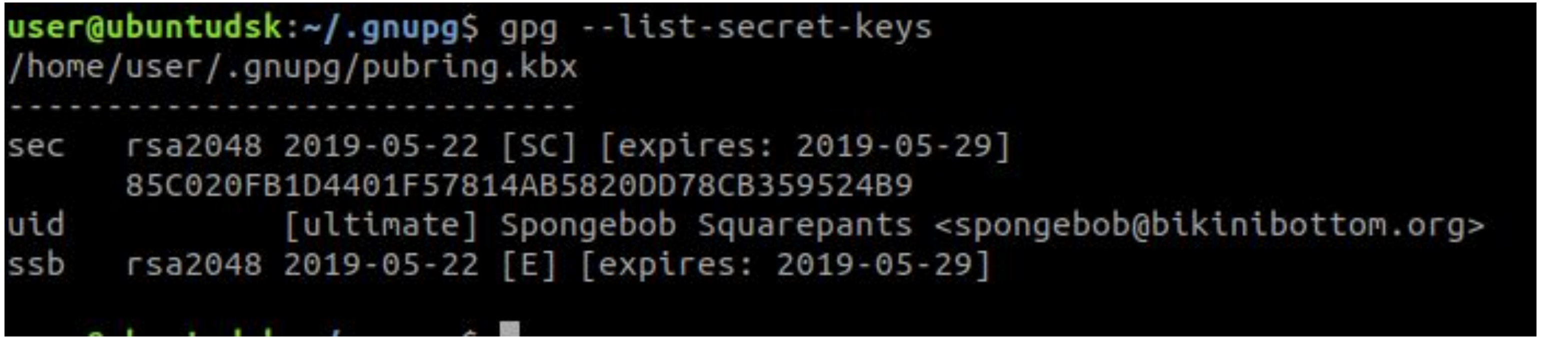
gpg --list-keys



```
user@ubuntudsk:~/.gnupg$ gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2019-05-29
/home/user/.gnupg/pubring.kbx
-----------------------------
pub   rsa2048 2019-05-22 [SC] [expires: 2019-05-29]
      85C020FB1D4401F57814AB5820DD78CB359524B9
uid           [ultimate] Spongebob Squarepants <spongebob@bikinibottom.org>
sub   rsa2048 2019-05-22 [E] [expires: 2019-05-29]
```

PGP

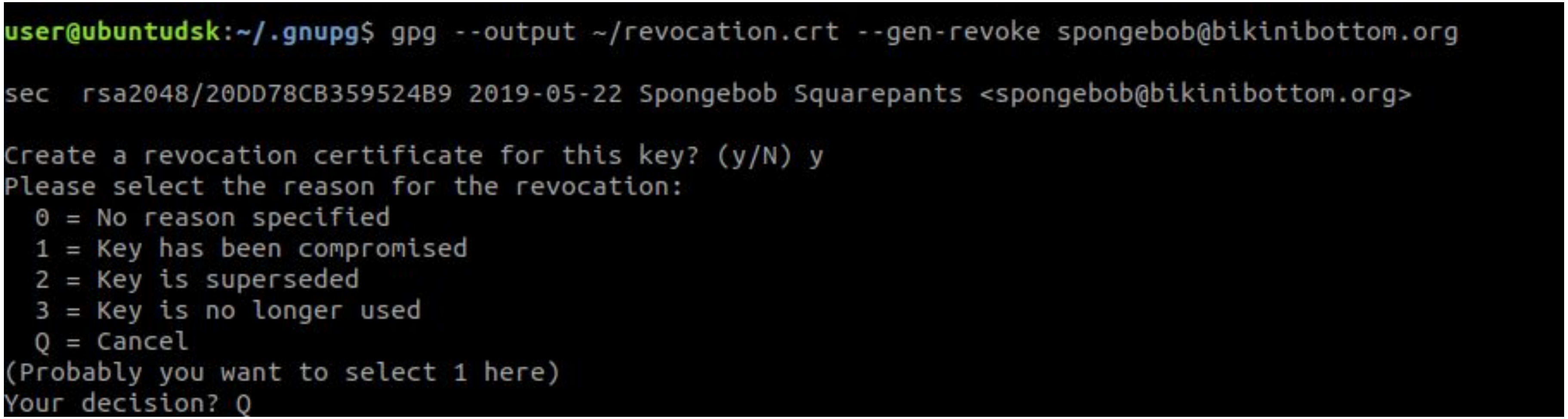
gpg --list-secret-keys



```
user@ubuntudsk:~/.gnupg$ gpg --list-secret-keys
/home/user/.gnupg/pubring.kbx
------------------------------
sec   rsa2048 2019-05-22 [SC] [expires: 2019-05-29]
      85C020FB1D4401F57814AB5820DD78CB359524B9
uid           [ultimate] Spongebob Squarepants <spongebob@bikinibottom.org>
ssb   rsa2048 2019-05-22 [E] [expires: 2019-05-29]
```

PGP

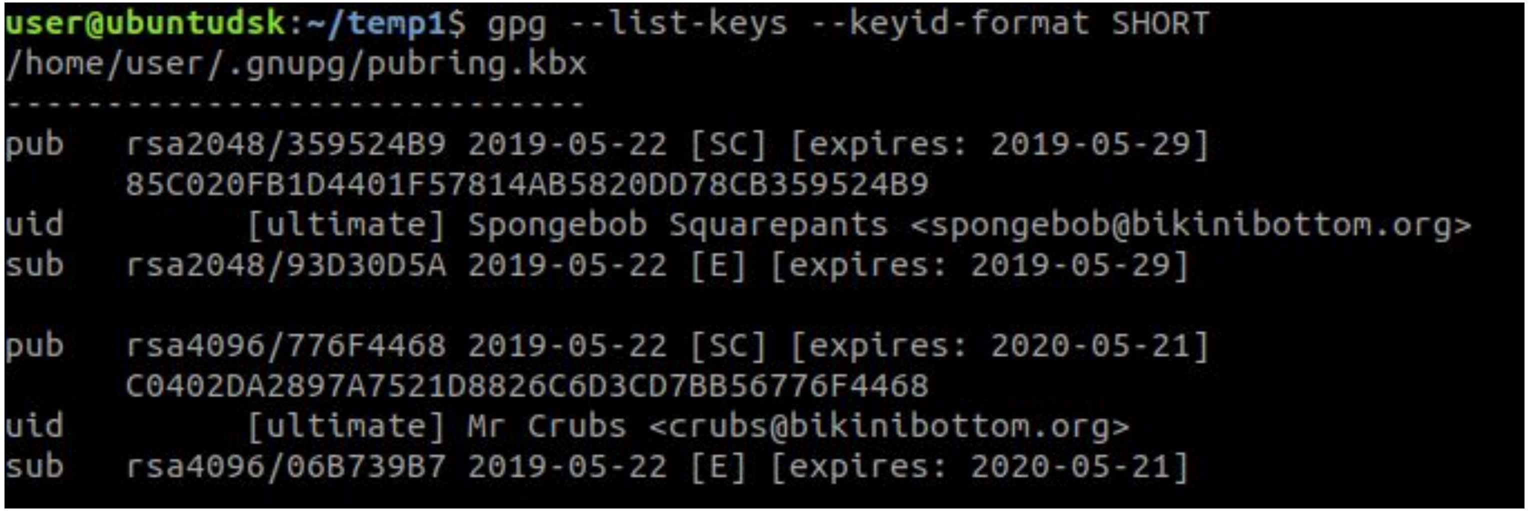
gpg --output ~/revocation.crt --gen-revoke spongebob@bikinibottom.org



```
user@ubuntudsk:~/.gnupg$ gpg --output ~/revocation.crt --gen-revoke spongebob@bikinibottom.org

sec  rsa2048/20DD78CB359524B9 2019-05-22 Spongebob Squarepants <spongebob@bikinibottom.org>

Create a revocation certificate for this key? (y/N) y
Please select the reason for the revocation:
  0 = No reason specified
  1 = Key has been compromised
  2 = Key is superseded
  3 = Key is no longer used
  Q = Cancel
(Probably you want to select 1 here)
Your decision? Q
```

PGP

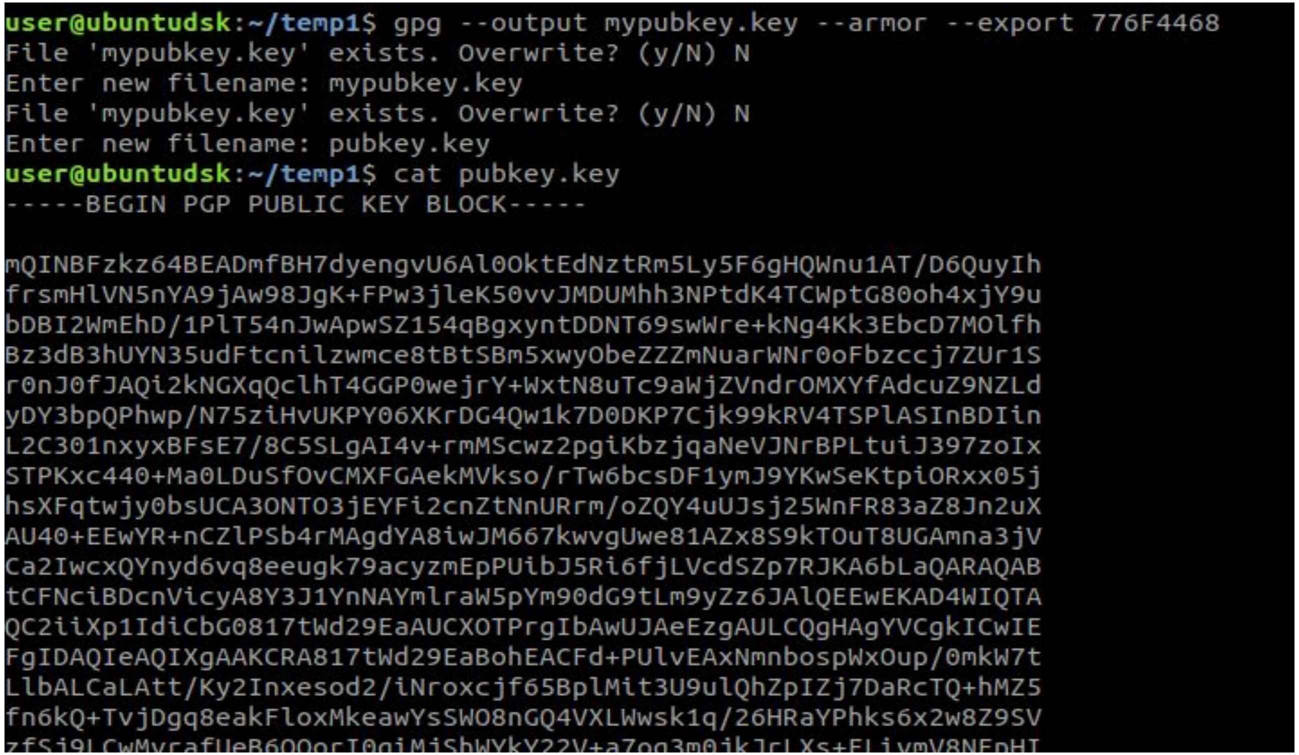
gpg --list-keys --keyid-format SHORT



```
user@ubuntudsk:~/temp1$ gpg --list-keys --keyid-format SHORT
/home/user/.gnupg/pubring.kbx
------------------------------
pub   rsa2048/359524B9 2019-05-22 [SC] [expires: 2019-05-29]
      85C020FB1D4401F57814AB5820DD78CB359524B9
uid         [ultimate] Spongebob Squarepants <spongebob@bikinibottom.org>
sub   rsa2048/93D30D5A 2019-05-22 [E] [expires: 2019-05-29]

pub   rsa4096/776F4468 2019-05-22 [SC] [expires: 2020-05-21]
      C0402DA2897A7521D8826C6D3CD7BB56776F4468
uid         [ultimate] Mr Crubs <crubs@bikinibottom.org>
sub   rsa4096/06B739B7 2019-05-22 [E] [expires: 2020-05-21]
```

PGP

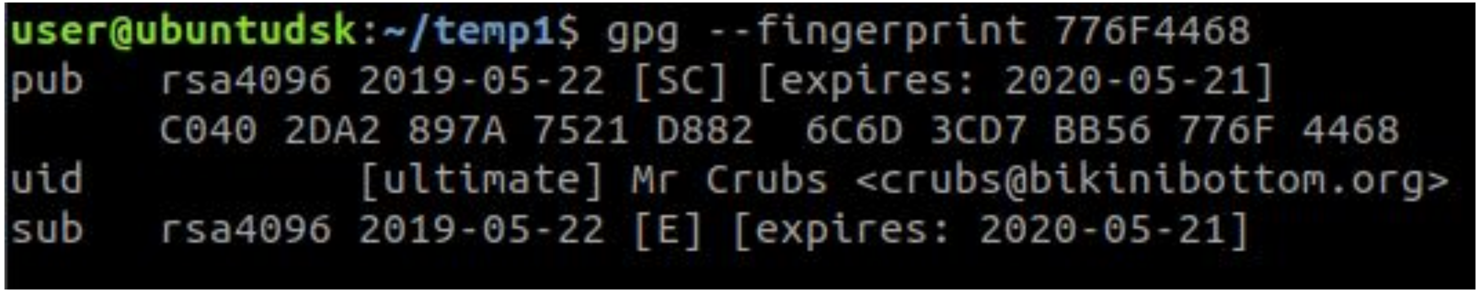
gpg --output mygpg.key --armor --export 776F4468



```
user@ubuntudsk:~/temp1$ gpg --output mypubkey.key --armor --export 776F4468
File 'mypubkey.key' exists. Overwrite? (y/N) N
Enter new filename: mypubkey.key
File 'mypubkey.key' exists. Overwrite? (y/N) N
Enter new filename: pubkey.key
user@ubuntudsk:~/temp1$ cat pubkey.key
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFzkz64BEADmfBH7dyengvU6Al0OktEdNztRm5Ly5F6gHQWnu1AT/D6QuyIh
frsmHlVN5nYA9jAw98JgK+FPw3jleK50vvJMDUMhh3NPtdK4TCWptG80oh4xjY9u
bDBI2WmEhD/1PlT54nJwApwSZ154qBgxyntDDNT69swWre+kNg4Kk3EbcD7MOlfh
Bz3dB3hUYN35udFtcnilzwmce8tBtSBm5xwyObeZZZmNuarWNr0oFbzccj7ZUr1S
r0nJ0fJAQi2kNGXqQclhT4GGP0wejrY+WxtN8uTc9aWjZVndrOMXYfAdcuZ9NZLd
yDY3bpQPhwp/N75ziHvUKPY06XKrDG4Qw1k7D0DKP7Cjk99kRV4TSPlASInBDIin
L2C301nxyxBFsE7/8C5SLgAI4v+rmMScwz2pgiKbzjqaNeVJNrBPLtuiJ397zoIx
STPKxc440+Ma0LDuSfOvCMXFGAekMVkso/rTw6bcsDF1ymJ9YKwSeKtpiORxx05j
hsXFqtwjy0bsUCA3ONTO3jEYFi2cnZtNnURrm/oZQY4uUJsj25WnFR83aZ8Jn2uX
AU40+EEwYR+nCZlPSb4rMAgdYA8iwJM667kwvgUwe81AZx8S9kTOuT8UGAmna3jV
Ca2IwcxQYnyd6vq8eeugk79acyzmEpPUibJ5Ri6fjLVcdSZp7RJKA6bLaQARAQAB
tCFNciBDcnVicyA8Y3J1YnNAYmlraW5pYm90dG9tLm9yZz6JAlQEEwEKAD4WIQTA
QC2iiXp1IdiCbG0817tWd29EaAUCXOTPrgIbAwUJAeEzgAULCQgHAgYVCgkICwIE
FgIDAQIeAQIXgAAKCRA817tWd29EaBohEACFd+PUlvEAxNmnbospWxOup/0mkW7t
LlbALCaLAtt/Ky2Inxesod2/iNroxcjf65BplMit3U9ulQhZpIZj7DaRcTQ+hMZ5
fn6kQ+TvjDgq8eakFloxMkeawYsSWO8nGQ4VXLWwsk1q/26HRaYPhks6x2w8Z9SV
zfSj9LCwMyrafUeB6OOorI0qiMjShWYkY22V+a7oq3m0jkJrLXs+ELiymV8NEpHI
```

PGP

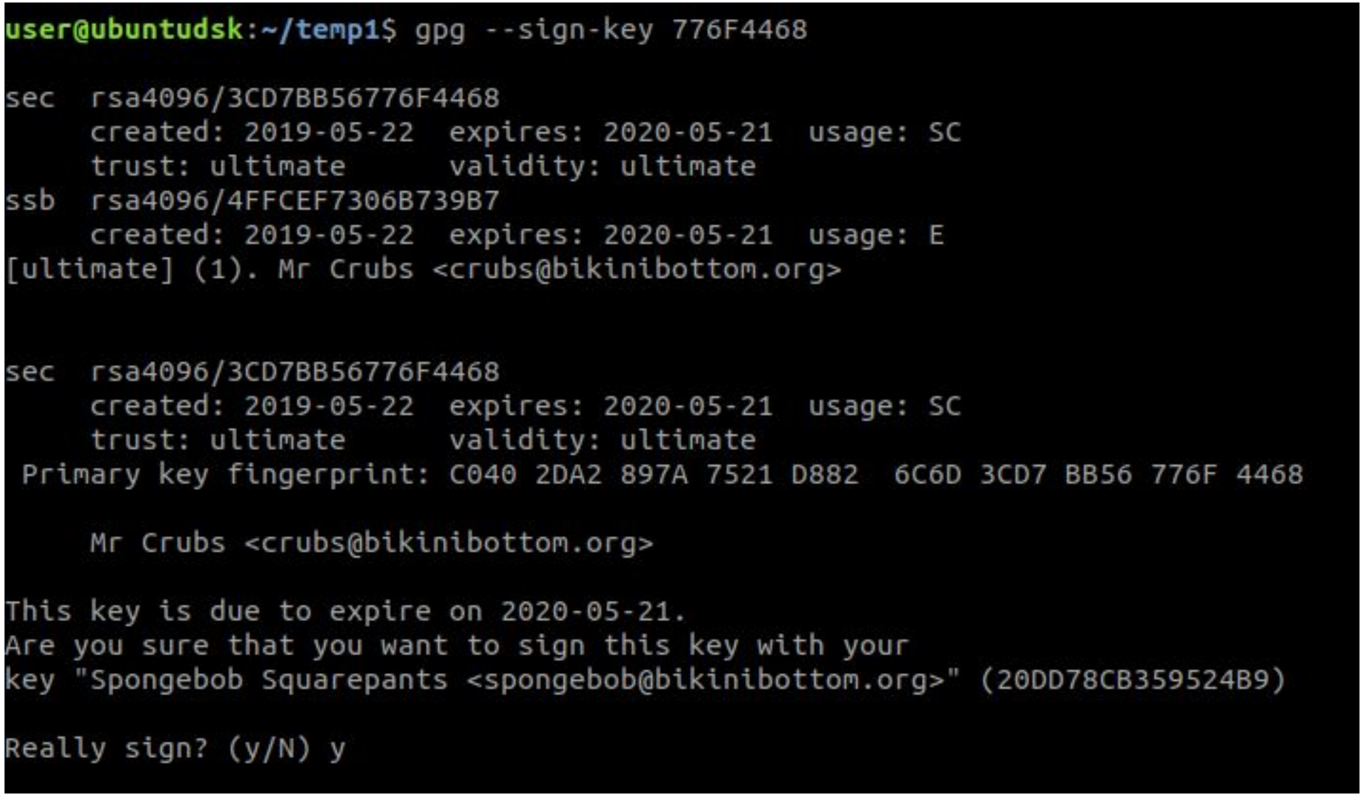
gpg --fingerprint 776F4468



```
user@ubuntudsk:~/temp1$ gpg --fingerprint 776F4468
pub   rsa4096 2019-05-22 [SC] [expires: 2020-05-21]
      C040 2DA2 897A 7521 D882  6C6D 3CD7 BB56 776F 4468
uid           [ultimate] Mr Crubs <crubs@bikinibottom.org>
sub   rsa4096 2019-05-22 [E] [expires: 2020-05-21]
```

PGP

gpg --sign-key key_id or email@example.com



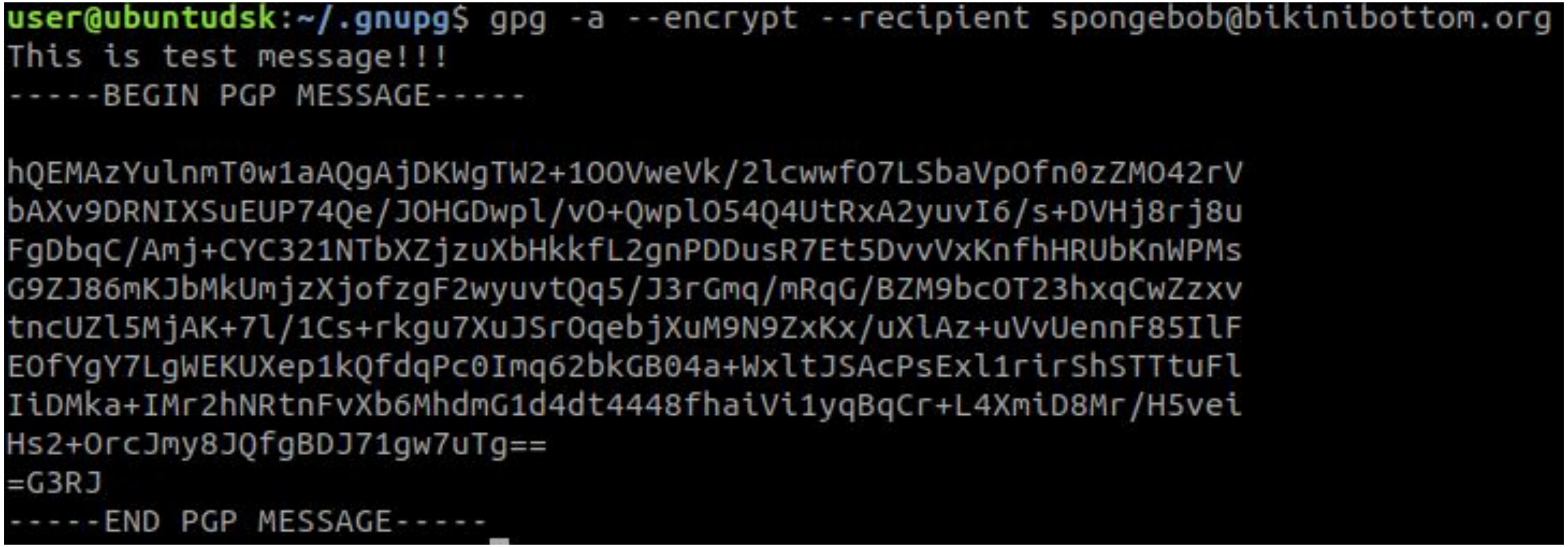
```
user@ubuntudsk:~/temp1$ gpg --sign-key 776F4468

sec  rsa4096/3CD7BB56776F4468
     created: 2019-05-22  expires: 2020-05-21  usage: SC
     trust: ultimate      validity: ultimate
ssb  rsa4096/4FFCEF7306B739B7
     created: 2019-05-22  expires: 2020-05-21  usage: E
[ultimate] (1). Mr Crubs <crubs@bikinibottom.org>


sec  rsa4096/3CD7BB56776F4468
     created: 2019-05-22  expires: 2020-05-21  usage: SC
     trust: ultimate      validity: ultimate
 Primary key fingerprint: C040 2DA2 897A 7521 D882  6C6D 3CD7 BB56 776F 4468

     Mr Crubs <crubs@bikinibottom.org>

This key is due to expire on 2020-05-21.
Are you sure that you want to sign this key with your
key "Spongebob Squarepants <spongebob@bikinibottom.org>" (20DD78CB359524B9)

Really sign? (y/N) y
```

PGP

```
gpg --keyserver pgp.mit.edu --send-keys 776F4468

gpg --keyserver pgp.mit.edu --recv-keys 776F4468
```

PGP

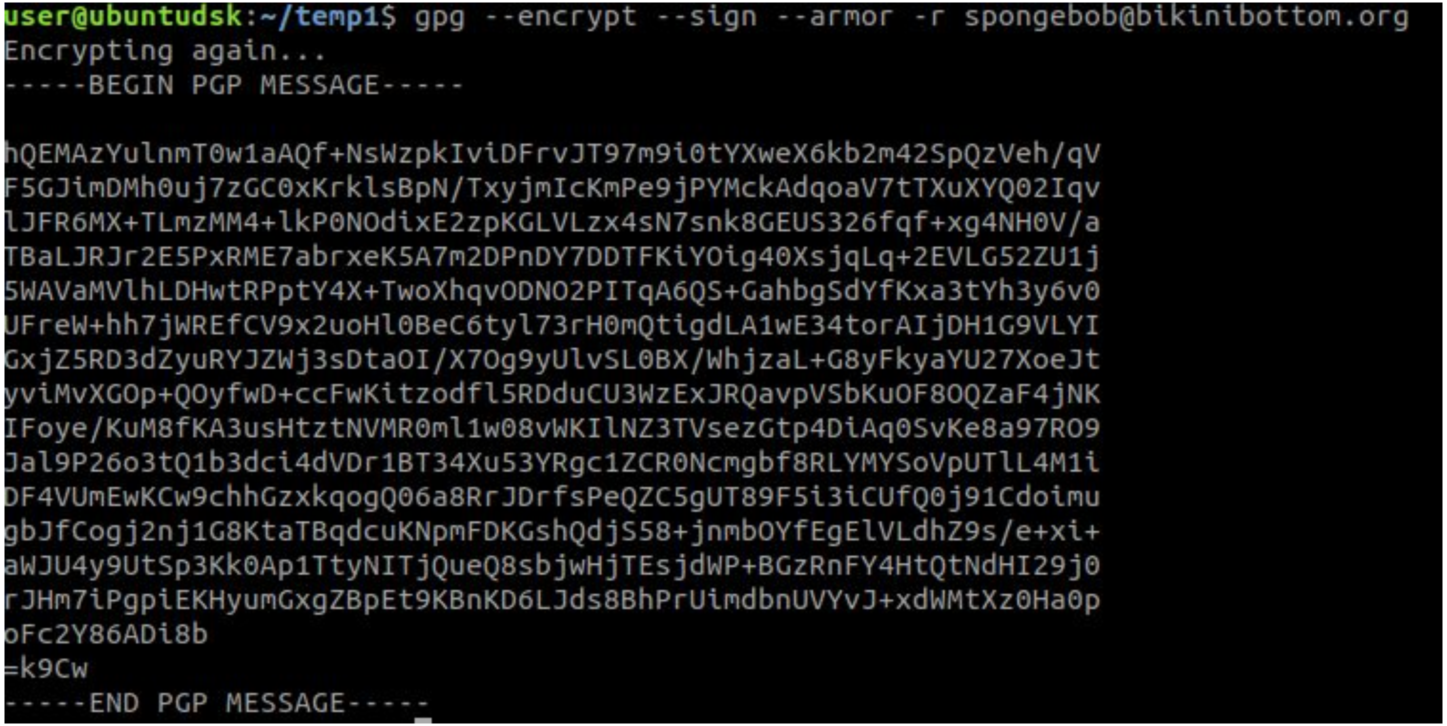
gpg -a --encrypt --recipient key_id or spongebob@bikinibottom.org



```
user@ubuntudsk:~/.gnupg$ gpg -a --encrypt --recipient spongebob@bikinibottom.org
This is test message!!!
-----BEGIN PGP MESSAGE-----

hQEMAzYulnmT0w1aAQgAjDKWgTW2+1OOVweVk/2lcwwfO7LSbaVpOfn0zZMO42rV
bAXv9DRNIXSuEUP74Qe/JOHGDwpl/vO+QwplO54Q4UtRxA2yuvI6/s+DVHj8rj8u
FgDbqC/Amj+CYC321NTbXZjzuXbHkkfL2gnPDDusR7Et5DvvVxKnfhHRUbKnWPMs
G9ZJ86mKJbMkUmjzXjofzgF2wyuvtQq5/J3rGmq/mRqG/BZM9bcOT23hxqCwZzxv
tncUZl5MjAK+7l/1Cs+rkgu7XuJSrOqebjXuM9N9ZxKx/uXlAz+uVvUennF85IlF
EOfYgY7LgWEKUXep1kQfdqPc0Imq62bkGB04a+WxltJSAcPsExl1rirShSTTtuFl
IiDMka+IMr2hNRtnFvXb6MhdmG1d4dt4448fhaiVi1yqBqCr+L4XmiD8Mr/H5vei
Hs2+OrcJmy8JQfgBDJ71gw7uTg==
=G3RJ
-----END PGP MESSAGE-----
```

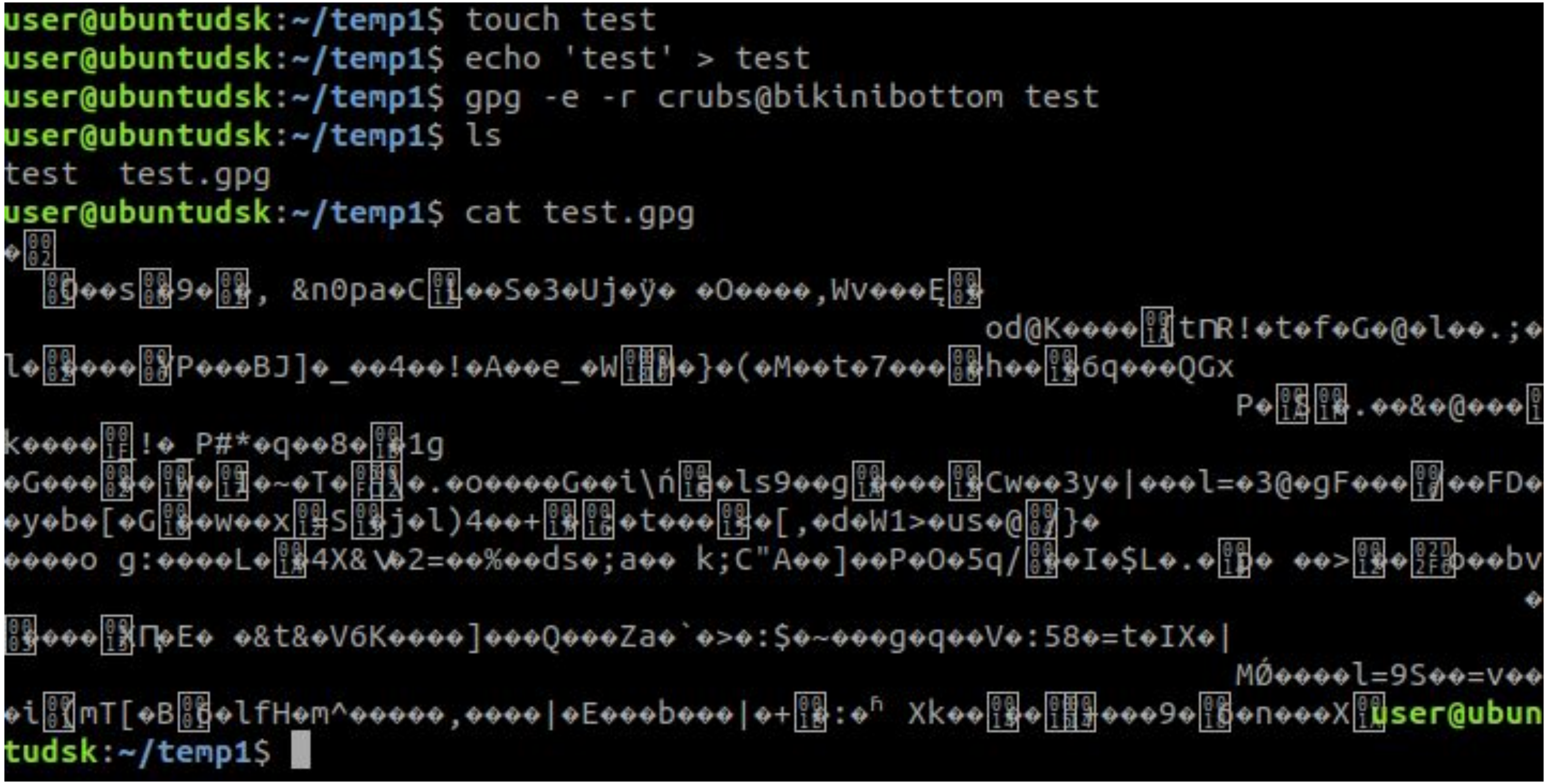
PGP

gpg --encrypt --sign --armor -r key_id or spongebob@bikinibottom.org



```
user@ubuntudsk:~/temp1$ gpg --encrypt --sign --armor -r spongebob@bikinibottom.org
Encrypting again...
-----BEGIN PGP MESSAGE-----

hQEMAzYulnmT0w1aAQf+NsWzpkIviDFrvJT97m9i0tYXweX6kb2m42SpQzVeh/qV
F5GJimDMh0uj7zGC0xKrklsBpN/TxyjmIcKmPe9jPYMckAdqoaV7tTXuXYQ02Iqv
lJFR6MX+TLmzMM4+lkP0NOdixE2zpKGLVLzx4sN7snk8GEUS326fqf+xg4NH0V/a
TBaLJRJr2E5PxRME7abrxeK5A7m2DPnDY7DDTFKiYOig40XsjqLq+2EVLG52ZU1j
5WAVaMVlhLDHwtRPptY4X+TwoXhqvODNO2PITqA6QS+GahbgSdYfKxa3tYh3y6v0
UFreW+hh7jWREfCV9x2uoHl0BeC6tyl73rH0mQtigdLA1wE34torAIjDH1G9VLYI
GxjZ5RD3dZyuRYJZWj3sDtaOI/X7Og9yUlvSL0BX/WhjzaL+G8yFkyaYU27XoeJt
yviMvXGOp+QOyfwD+ccFwKitzodfl5RDduCU3WzExJRQavpVSbKuOF8OQZaF4jNK
IFoye/KuM8fKA3usHtztNVMR0ml1w08vWKIlNZ3TVsezGtp4DiAq0SvKe8a97RO9
Jal9P26o3tQ1b3dci4dVDr1BT34Xu53YRgc1ZCR0Ncmgbf8RLYMYSoVpUTlL4M1i
DF4VUmEwKCw9chhGzxkqogQ06a8RrJDrfsPeQZC5gUT89F5i3iCUfQ0j91Cdoimu
gbJfCogj2nj1G8KtaTBqdcuKNpmFDKGshQdjS58+jnmbOYfEgElVLdhZ9s/e+xi+
aWJU4y9UtSp3Kk0Ap1TtyNITjQueQ8sbjwHjTEsjdWP+BGzRnFY4HtQtNdHI29j0
rJHm7iPgpiEKHyumGxgZBpEt9KBnKD6LJds8BhPrUimdbnUVYvJ+xdWMtXz0Ha0p
oFc2Y86ADi8b
=k9Cw
-----END PGP MESSAGE-----
```

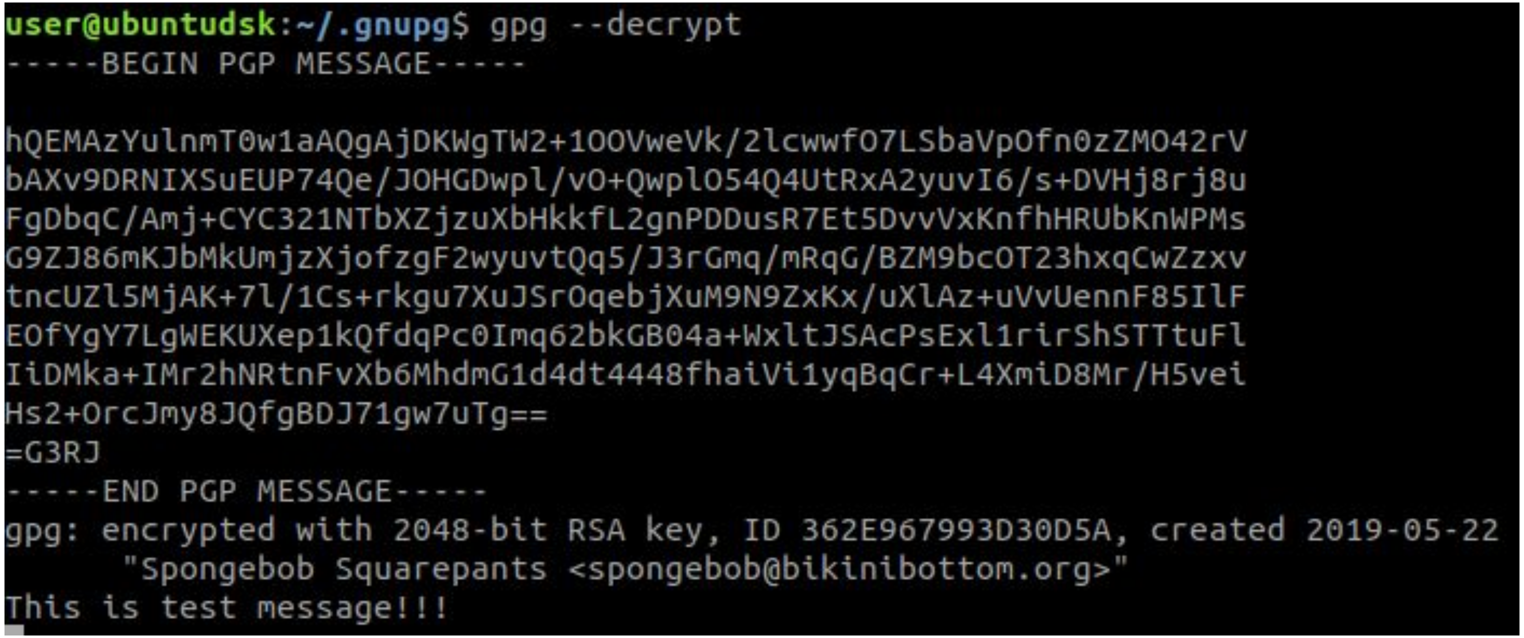
PGP



```
user@ubuntudsk:~/temp1$ touch test
user@ubuntudsk:~/temp1$ echo 'test' > test
user@ubuntudsk:~/temp1$ gpg -e -r crubs@bikinibottom test
user@ubuntudsk:~/temp1$ ls
test  test.gpg
user@ubuntudsk:~/temp1$ cat test.gpg
�▯
  ▯Q��s▯�9�▯�, &n0pa�C▯l��S�3�Uj�ÿ� �O�����,Wv���Ę▯�
                                                     od@K�����▯[tПR!�t�f�G�@�l��.;�
l�▯����▯YP���BJ]�_��4��!�A���e_�W▯▯��}�(�M��t�7���▯�h��▯�6q���QGx
                                                                P�▯S▯�.��&�@���▯
k�����▯!�_P#*�q��8�▯�1g
�G���▯��▯W�▯]�~�T�▯▯�.�o�����G��i\ń▯�ls9��g▯����▯�Cw��3y�|���l=�3@�gF���▯��FD�
�y�b�[�G▯��w��x▯=S▯�j�l)4��+▯�▯�t���▯�[,�d�W1>�us�@▯}�
����o g:����L�▯�4X&V�2=��%��ds�;a�� k;C"A��]��P�O�5q/▯��I�$L�.�▯p� ��>▯�▯p��bv
                                                                              �
▯����▯XП̧�E� �&t&�V6K����]���Q���Za�`�>�:$�~���g�q��V�:58�=t�IX�|
                                                                MØ����l=9S��=v��
�i▯{mT[�B▯б�lfH�m^�����,����|�E���b���|�+▯�:�ʰ Xk��▯�▯▯���9�▯б�n���X▯user@ubun
tudsk:~/temp1$ 
```

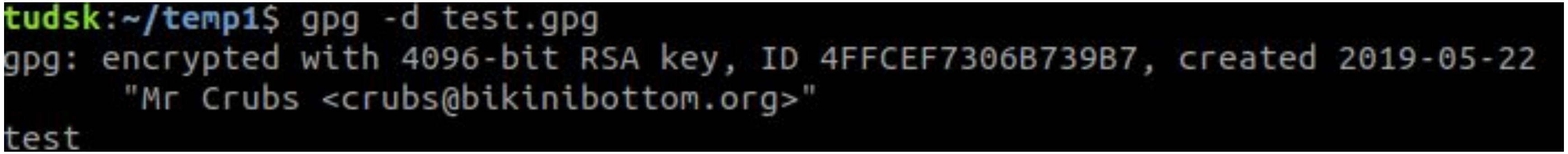
PGP

gpg --decrypt



```
user@ubuntudsk:~/.gnupg$ gpg --decrypt
-----BEGIN PGP MESSAGE-----

hQEMAzYulnmT0w1aAQgAjDKWgTW2+1OOVweVk/2lcwwfO7LSbaVpOfn0zZMO42rV
bAXv9DRNIXSuEUP74Qe/JOHGDwpl/vO+QwplO54Q4UtRxA2yuvI6/s+DVHj8rj8u
FgDbqC/Amj+CYC321NTbXZjzuXbHkkfL2gnPDDusR7Et5DvvVxKnfhHRUbKnWPMs
G9ZJ86mKJbMkUmjzXjofzgF2wyuvtQq5/J3rGmq/mRqG/BZM9bcOT23hxqCwZzxv
tncUZl5MjAK+7l/1Cs+rkgu7XuJSrOqebjXuM9N9ZxKx/uXlAz+uVvUennF85IlF
EOfYgY7LgWEKUXep1kQfdqPc0Imq62bkGB04a+WxltJSAcPsExl1rirShSTTtuFl
IiDMka+IMr2hNRtnFvXb6MhdmG1d4dt4448fhaiVi1yqBqCr+L4XmiD8Mr/H5vei
Hs2+OrcJmy8JQfgBDJ71gw7uTg==
=G3RJ
-----END PGP MESSAGE-----
gpg: encrypted with 2048-bit RSA key, ID 362E967993D30D5A, created 2019-05-22
      "Spongebob Squarepants <spongebob@bikinibottom.org>"
This is test message!!!
```

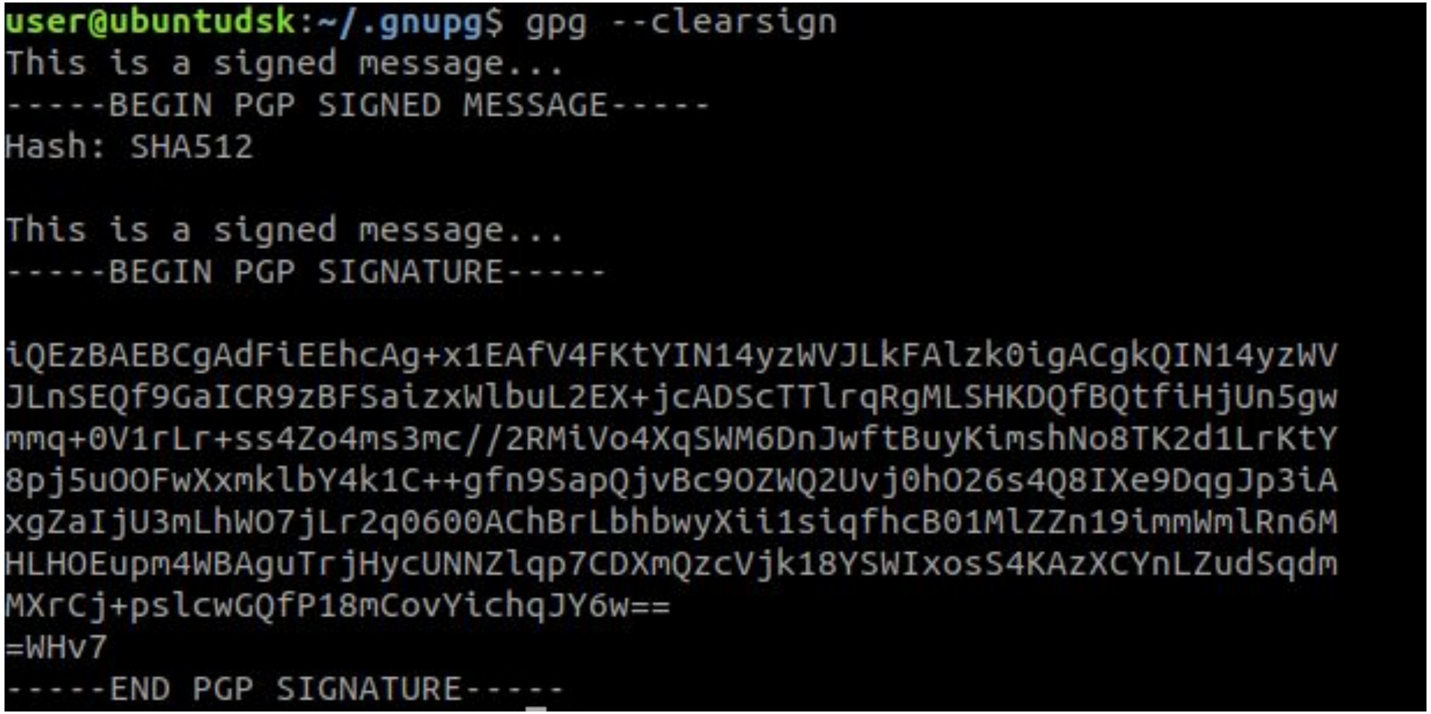
PGP



```
tudsk:~/temp1$ gpg -d test.gpg
gpg: encrypted with 4096-bit RSA key, ID 4FFCEF7306B739B7, created 2019-05-22
      "Mr Crubs <crubs@bikinibottom.org>"
test
```

PGP

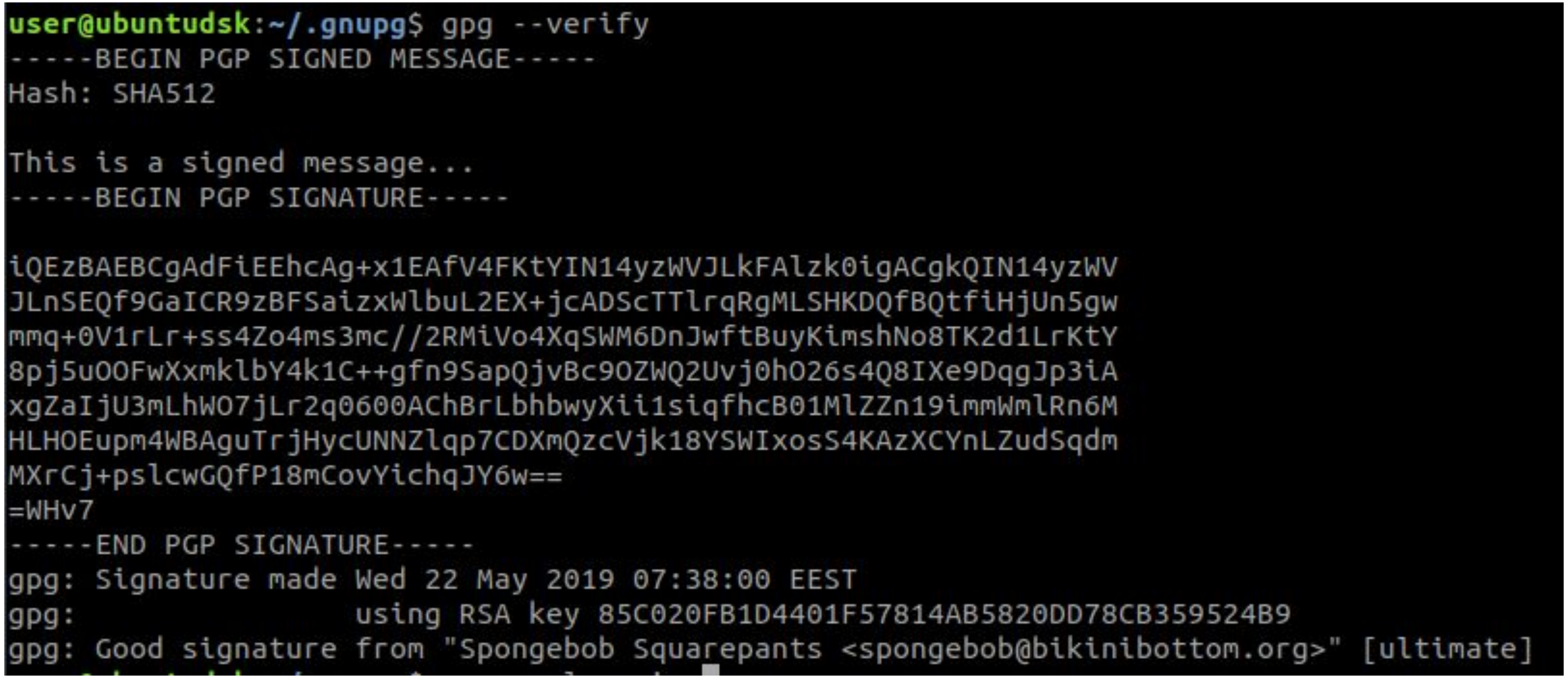
gpg --clearsign



```
user@ubuntudsk:~/.gnupg$ gpg --clearsign
This is a signed message...
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

This is a signed message...
-----BEGIN PGP SIGNATURE-----

iQEzBAEBCgAdFiEEhcAg+x1EAfV4FKtYIN14yzWVJLkFAlzk0igACgkQIN14yzWV
JLnSEQf9GaICR9zBFSaizxWlbuL2EX+jcADScTTlrqRgMLSHKDQfBQtfiHjUn5gw
mmq+0V1rLr+ss4Zo4ms3mc//2RMiVo4XqSWM6DnJwftBuyKimshNo8TK2d1LrKtY
8pj5uOOFwXxmklbY4k1C++gfn9SapQjvBc9OZWQ2Uvj0hO26s4Q8IXe9DqgJp3iA
xgZaIjU3mLhWO7jLr2q0600AChBrLbhbwyXii1siqfhcB01MlZZn19immWmlRn6M
HLHOEupm4WBAguTrjHycUNNZlqp7CDXmQzcVjk18YSWIxosS4KAzXCYnLZudSqdm
MXrCj+pslcwGQfP18mCovYichqJY6w==
=WHv7
-----END PGP SIGNATURE-----
```

PGP

gpg --verify



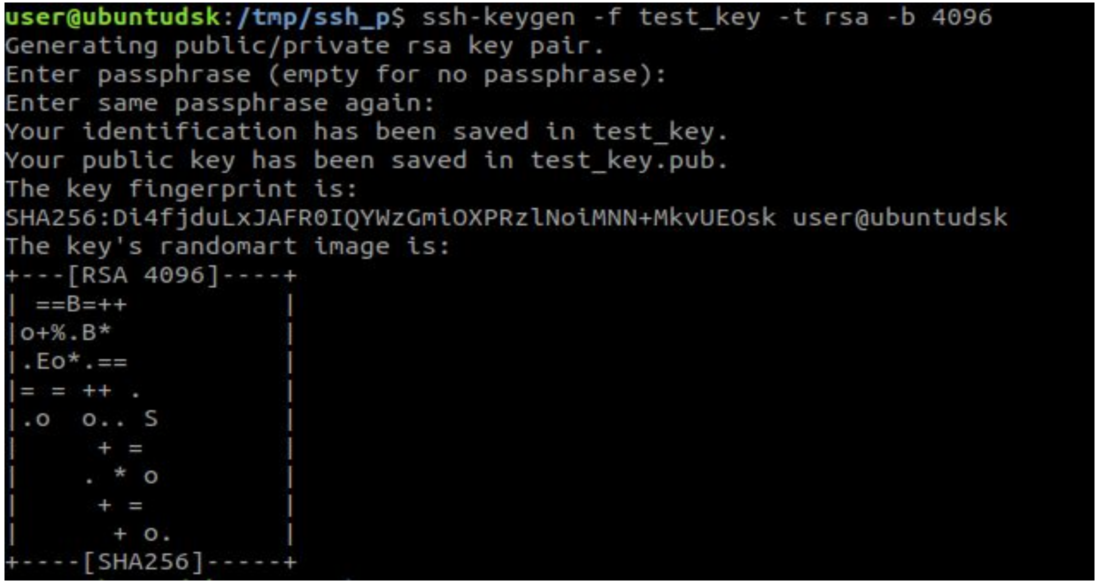
```
user@ubuntudsk:~/.gnupg$ gpg --verify
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

This is a signed message...
-----BEGIN PGP SIGNATURE-----

iQEzBAEBCgAdFiEEhcAg+x1EAfV4FKtYIN14yzWVJLkFAlzk0igACgkQIN14yzWV
JLnSEQf9GaICR9zBFSaizxWlbuL2EX+jcADScTTlrqRgMLSHKDQfBQtfiHjUn5gw
mmq+0V1rLr+ss4Zo4ms3mc//2RMiVo4XqSWM6DnJwftBuyKimshNo8TK2d1LrKtY
8pj5uOOFwXxmklbY4k1C++gfn9SapQjvBc9OZWQ2Uvj0hO26s4Q8IXe9DqgJp3iA
xgZaIjU3mLhWO7jLr2q0600AChBrLbhbwyXii1siqfhcB01MlZZn19immWmlRn6M
HLHOEupm4WBAguTrjHycUNNZlqp7CDXmQzcVjk18YSWIxosS4KAzXCYnLZudSqdm
MXrCj+pslcwGQfP18mCovYichqJY6w==
=WHv7
-----END PGP SIGNATURE-----
gpg: Signature made Wed 22 May 2019 07:38:00 EEST
gpg:                using RSA key 85C020FB1D4401F57814AB5820DD78CB359524B9
gpg: Good signature from "Spongebob Squarepants <spongebob@bikinibottom.org>" [ultimate]
```

Next slides are from some topics that we talked about + ssh key creation.

SSH

ssh-keygen -f test_key -t rsa -b 4096

-f <filename_to_save_private_key>	
-t <algorithm_to_use>	rsa, dsa, ecdsa, ed25519
-b <key_size>	differs to each algorithm

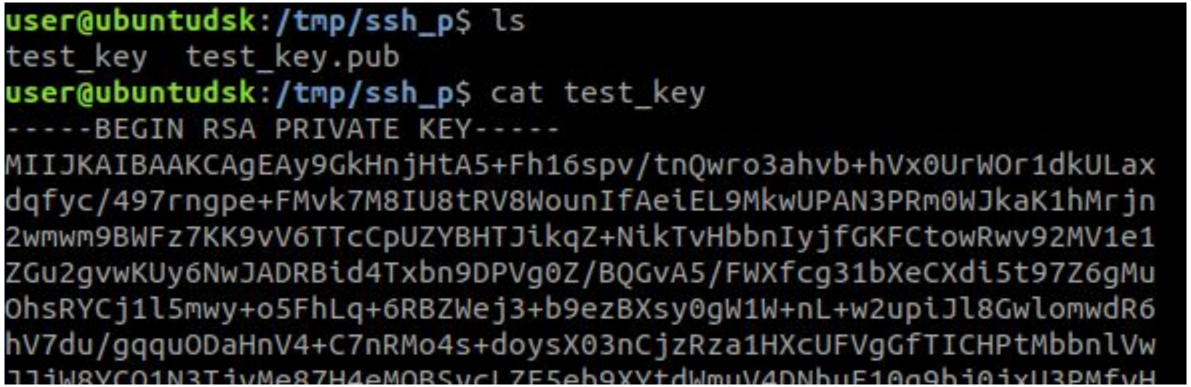


```
user@ubuntudsk:/tmp/ssh_p$ ssh-keygen -f test_key -t rsa -b 4096
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in test_key.
Your public key has been saved in test_key.pub.
The key fingerprint is:
SHA256:Di4fjduLxJAFR0IQYWzGmiOXPRzlNoiMNN+MkvUEOsk user@ubuntudsk
The key's randomart image is:
+---[RSA 4096]----+
| ==B=++          |
|o+%.B*           |
|.Eo*.==          |
|= = ++ .         |
|.o  o.. S        |
|     + =         |
|    . * o        |
|     + =         |
|      + o.       |
+----[SHA256]-----+
```

More info:
https://www.digitalocean.com/community/tutorials/ssh-essentials-working-with-ssh-servers-clients-and-keys

SSH

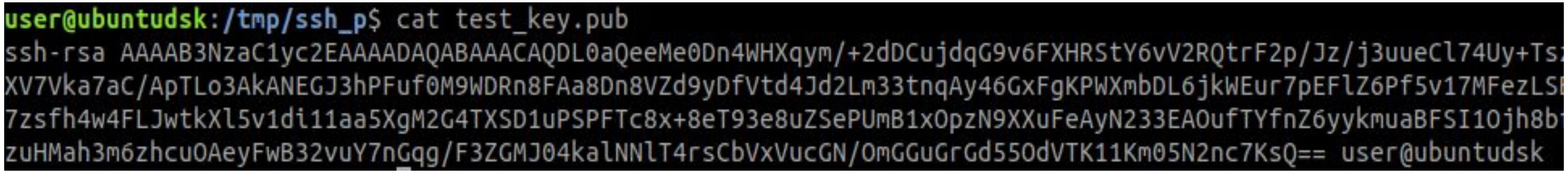
Private key



```
user@ubuntudsk:/tmp/ssh_p$ ls
test_key  test_key.pub
user@ubuntudsk:/tmp/ssh_p$ cat test_key
-----BEGIN RSA PRIVATE KEY-----
MIIJKAIBAAKCAgEAy9GkHnjHtA5+Fh16spv/tnQwro3ahvb+hVx0UrWOr1dkULax
dqfyc/497rngpe+FMvk7M8IU8tRV8WounIfAeiEL9MkwUPAN3PRm0WJkaK1hMrjn
2wmwm9BWFz7KK9vV6TTcCpUZYBHTJikqZ+NikTvHbbnIyjfGKFCtowRwv92MV1e1
ZGu2gvwKUy6NwJADRBid4Txbn9DPVg0Z/BQGvA5/FWXfcg31bXeCXdi5t97Z6gMu
OhsRYCj1l5mwy+o5FhLq+6RBZWej3+b9ezBXsy0gW1W+nL+w2upiJl8GwlomwdR6
hV7du/gqquODaHnV4+C7nRMo4s+doysX03nCjzRza1HXcUFVgGfTICHPtMbbnlVw
JJiW8YCQ1N3TivMe87H4eMQBSvcLZF5eb9XYtdWmuV4DNbuF10q9bj0ixU3PMfvH
```

SSH

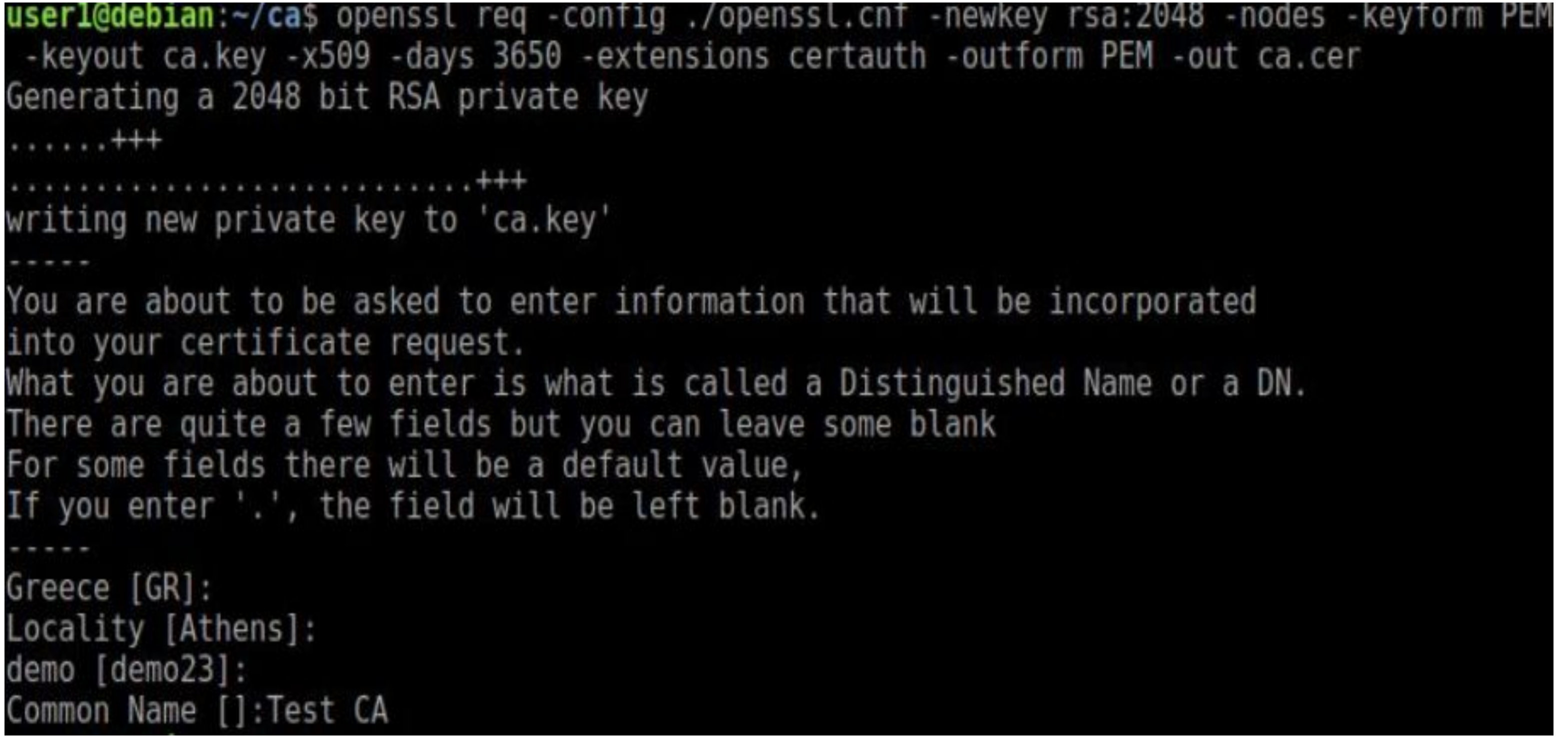
Public key (store this to the remote computer that you want to connect to)



```
user@ubuntudsk:/tmp/ssh_p$ cat test_key.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDL0aQeeMe0Dn4WHXqym/+2dDCujdqG9v6FXHRStY6vV2RQtrF2p/Jz/j3uueCl74Uy+Ts
XV7Vka7aC/ApTLo3AkANEGJ3hPFuf0M9WDRn8FAa8Dn8VZd9yDfVtd4Jd2Lm33tnqAy46GxFgKPWXmbDL6jkWEur7pEFlZ6Pf5v17MFezLS
7zsfh4w4FLJwtkXl5v1di11aa5XgM2G4TXSD1uPSPFTc8x+8eT93e8uZSePUmB1xOpzN9XXuFeAyN233EAOufTYfnZ6yykmuaBFSI1Ojh8b
zuHMah3m6zhcuOAeyFwB32vuY7nGqg/F3ZGMJ04kalNNlT4rsCbVxVucGN/OmGGuGrGd55OdVTK11Km05N2nc7KsQ== user@ubuntudsk
```

Openssl, x509, CA

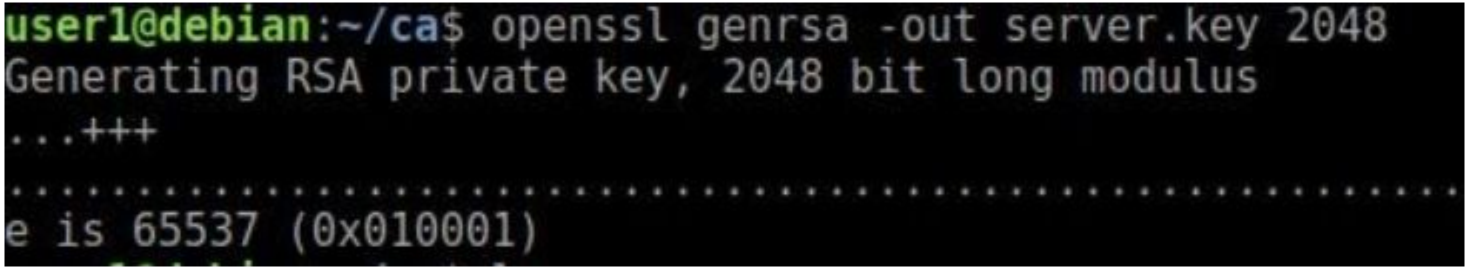
Creating certificate authority ca.key and ca.cer



```
user1@debian:~/ca$ openssl req -config ./openssl.cnf -newkey rsa:2048 -nodes -keyform PEM
 -keyout ca.key -x509 -days 3650 -extensions certauth -outform PEM -out ca.cer
Generating a 2048 bit RSA private key
......+++
...........................+++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Greece [GR]:
Locality [Athens]:
demo [demo23]:
Common Name []:Test CA
```

Openssl, x509, CA

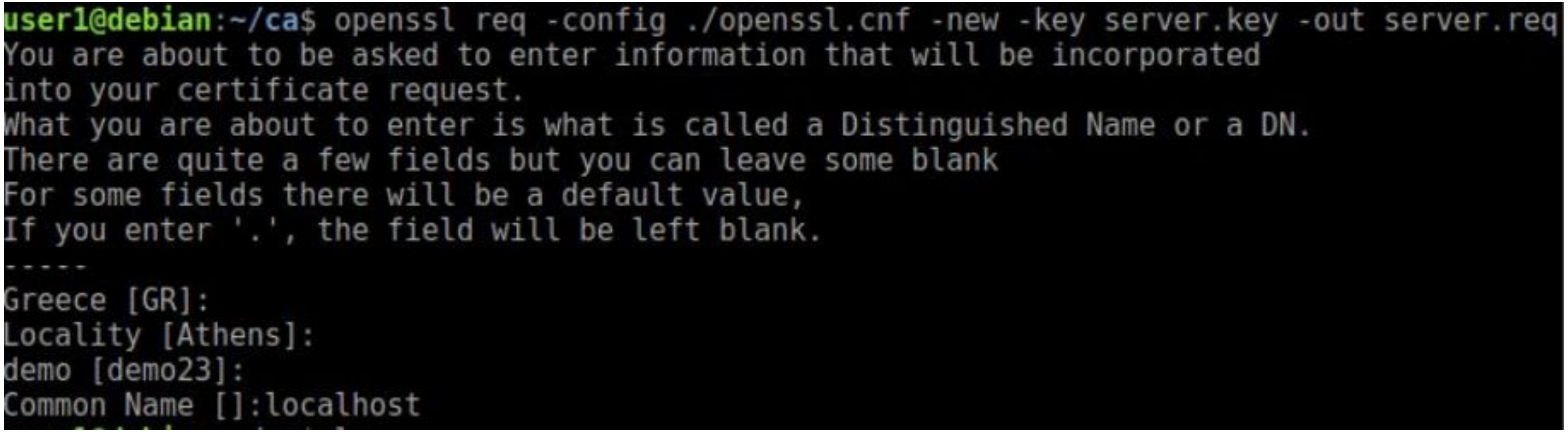
Generating our server’s private key



```
user1@debian:~/ca$ openssl genrsa -out server.key 2048
Generating RSA private key, 2048 bit long modulus
...+++
..................................................................
e is 65537 (0x010001)
```

Openssl, x509, CA

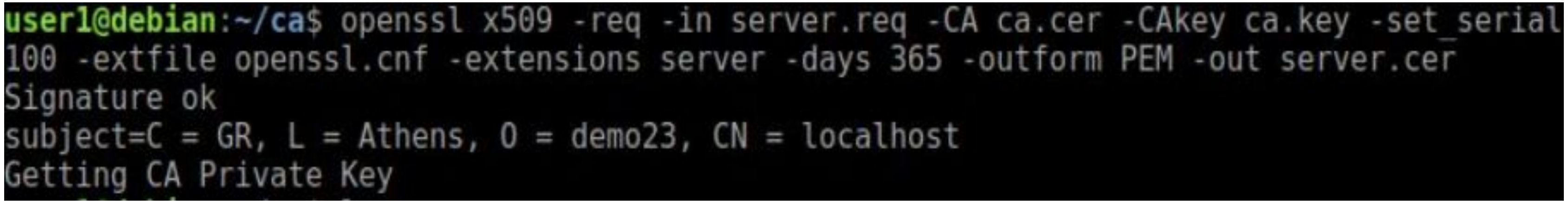
Generating a certificate signing request that we send to our ca .



```
user1@debian:~/ca$ openssl req -config ./openssl.cnf -new -key server.key -out server.req
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Greece [GR]:
Locality [Athens]:
demo [demo23]:
Common Name []:localhost
```

Openssl, x509, CA

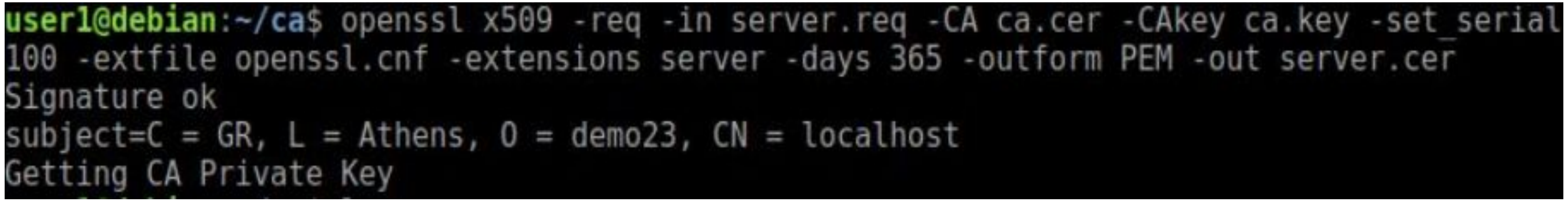
CA receives our request and generates our x509 certificate.



```
user1@debian:~/ca$ openssl x509 -req -in server.req -CA ca.cer -CAkey ca.key -set_serial
100 -extfile openssl.cnf -extensions server -days 365 -outform PEM -out server.cer
Signature ok
subject=C = GR, L = Athens, O = demo23, CN = localhost
Getting CA Private Key
```

Openssl, x509, CA

We need to enable apache2 to use our ssl certificate.
After that we have One way SSL authentication to our web server.

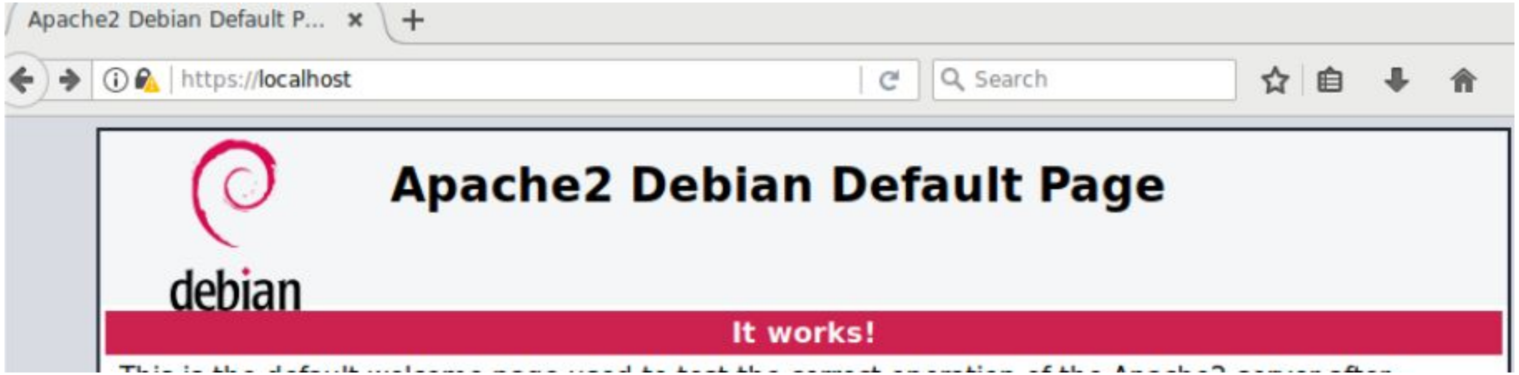


```
user1@debian:~/ca$ openssl x509 -req -in server.req -CA ca.cer -CAkey ca.key -set_serial
100 -extfile openssl.cnf -extensions server -days 365 -outform PEM -out server.cer
Signature ok
subject=C = GR, L = Athens, O = demo23, CN = localhost
Getting CA Private Key
```

More info on configuring apache to use a certificate:
https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-16-04
https://www.digitalocean.com/community/tutorials/openssl-essentials-working-with-ssl-certificates-private-keys-and-csrs
https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-apache-for-debian-8

Openssl, x509, CA

Finally, our web server with our certificate.



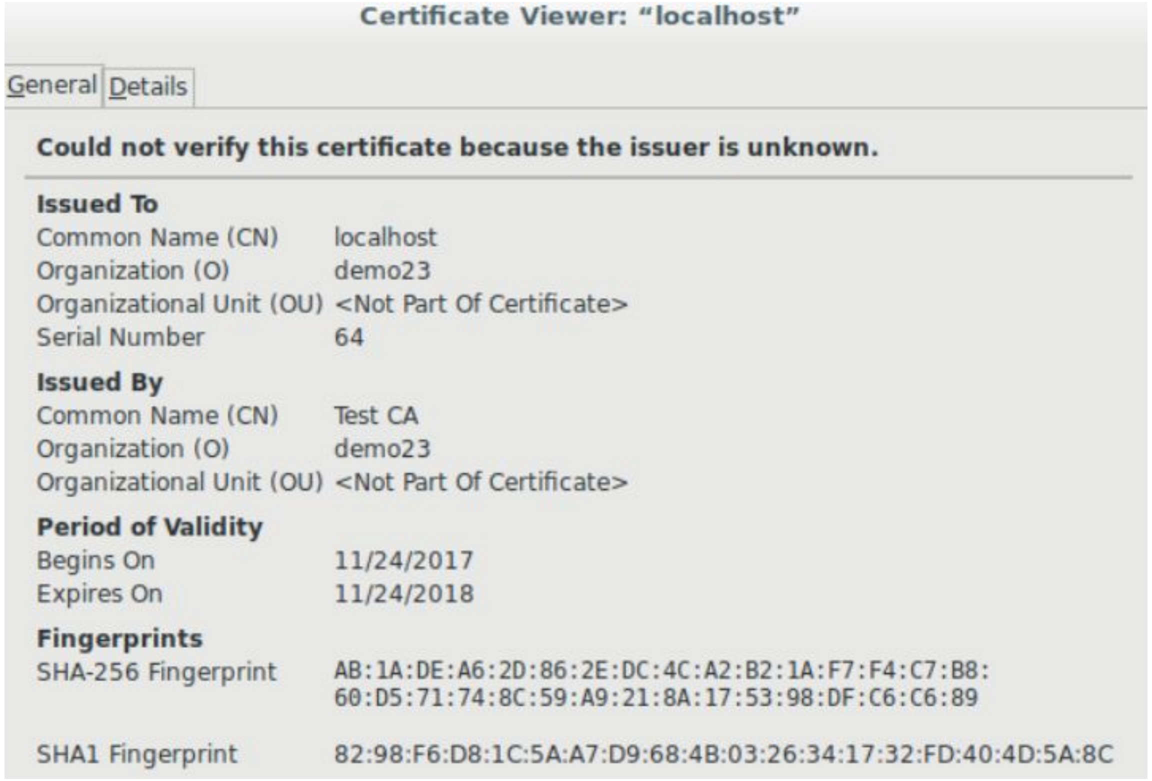
Apache2 Debian Default Page

debian

It works!

Openssl, x509, CA

Finally, our web server with our certificate.
Our certificate information that we can get from any browser.



Certificate Viewer: “localhost”

Could not verify this certificate because the issuer is unknown.

Issued To

Common Name (CN)	localhost
Organization (O)	demo23
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	64

Issued By

Common Name (CN)	Test CA
Organization (O)	demo23
Organizational Unit (OU)	<Not Part Of Certificate>

Period of Validity

Begins On	11/24/2017
Expires On	11/24/2018

Fingerprints

SHA-256 Fingerprint	AB:1A:DE:A6:2D:86:2E:DC:4C:A2:B2:1A:F7:F4:C7:B8: 60:D5:71:74:8C:59:A9:21:8A:17:53:98:DF:C6:C6:89
SHA1 Fingerprint	82:98:F6:D8:1C:5A:A7:D9:68:4B:03:26:34:17:32:FD:40:4D:5A:8C