GPG generate key

Create key

```
[clerie@krypton ~]$ gpg --full-generate-key --expert
gpg (GnuPG) 2.4.5; Copyright (C) 2024 g10 Code GmbH
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
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
   (7) DSA (set your own capabilities)
   (8) RSA (set your own capabilities)
   (9) ECC (sign and encrypt) *default*
  (10) ECC (sign only)
  (11) ECC (set your own capabilities)
  (13) Existing key
  (14) Existing key from card
Your selection? 8

Possible actions for this RSA key: Sign Certify Encrypt Authenticate
Current allowed actions: Sign Certify Encrypt

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
   (A) Toggle the authenticate capability
   (Q) Finished

Your selection? s

Possible actions for this RSA key: Sign Certify Encrypt Authenticate
Current allowed actions: Certify Encrypt

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
   (A) Toggle the authenticate capability
   (Q) Finished

Your selection? e

Possible actions for this RSA key: Sign Certify Encrypt Authenticate
Current allowed actions: Certify

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
```

```
   (A) Toggle the authenticate capability
   (Q) Finished

Your selection? q
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) 1y
Key expires at Di 15 Apr 2025 12:30:36 CEST
Is this correct? (y/N) y

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

Real name: clerie
Email address: clerie@clerie.de
Comment: test 2024-04-15
You selected this USER-ID:
    "clerie (test 2024-04-15) <clerie@clerie.de>"

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.
gpg: directory '/home/clerie/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/clerie/.gnupg/openpgp-
revocs.d/A1EF35BC29E3A6E55F3CFBDCEB93D1E94EF29562.rev'
public and secret key created and signed.

pub   rsa4096 2024-04-15 [C] [expires: 2025-04-15]
      A1EF35BC29E3A6E55F3CFBDCEB93D1E94EF29562
uid                      clerie (test 2024-04-15) <clerie@clerie.de>
```

Add subkeys

```
[clerie@krypton ~]$ gpg --expert --edit-key
A1EF35BC29E3A6E55F3CFBDCEB93D1E94EF29562
gpg (GnuPG) 2.4.5; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Secret key is available.

gpg: checking the trustdb
```

```
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   8  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 8u
gpg: next trustdb check due at 2025-04-15
sec  rsa4096/EB93D1E94EF29562
     created: 2024-04-15  expires: 2025-04-15  usage: C
     trust: ultimate      validity: ultimate
[ultimate] (1). clerie (test 2024-04-15) <clerie@clerie.de>

gpg> addkey
Please select what kind of key you want:
   (3) DSA (sign only)
   (4) RSA (sign only)
   (5) Elgamal (encrypt only)
   (6) RSA (encrypt only)
   (7) DSA (set your own capabilities)
   (8) RSA (set your own capabilities)
  (10) ECC (sign only)
  (11) ECC (set your own capabilities)
  (12) ECC (encrypt only)
  (13) Existing key
  (14) Existing key from card
Your selection? 8

Possible actions for this RSA key: Sign Encrypt Authenticate
Current allowed actions: Sign Encrypt

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
   (A) Toggle the authenticate capability
   (Q) Finished

Your selection? ea
Invalid selection.

Possible actions for this RSA key: Sign Encrypt Authenticate
Current allowed actions: Sign Encrypt

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
   (A) Toggle the authenticate capability
   (Q) Finished

Your selection? e

Possible actions for this RSA key: Sign Encrypt Authenticate
Current allowed actions: Sign

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
   (A) Toggle the authenticate capability
   (Q) Finished
```

```
Your selection? q
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) 1y
Key expires at Di 15 Apr 2025 12:37:46 CEST
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  rsa4096/EB93D1E94EF29562
     created: 2024-04-15  expires: 2025-04-15  usage: C
     trust: ultimate      validity: ultimate
ssb  rsa4096/0E6F5B37473B0B62
     created: 2024-04-15  expires: 2025-04-15  usage: SR
[ultimate] (1). clerie (test 2024-04-15) <clerie@clerie.de>

gpg> addkey
Please select what kind of key you want:
   (3) DSA (sign only)
   (4) RSA (sign only)
   (5) Elgamal (encrypt only)
   (6) RSA (encrypt only)
   (7) DSA (set your own capabilities)
   (8) RSA (set your own capabilities)
  (10) ECC (sign only)
  (11) ECC (set your own capabilities)
  (12) ECC (encrypt only)
  (13) Existing key
  (14) Existing key from card
Your selection? 8

Possible actions for this RSA key: Sign Encrypt Authenticate
Current allowed actions: Sign Encrypt

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
   (A) Toggle the authenticate capability
   (Q) Finished

Your selection? s
```

```
Possible actions for this RSA key: Sign Encrypt Authenticate
Current allowed actions: Encrypt

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
   (A) Toggle the authenticate capability
   (Q) Finished

Your selection? q
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) 1y
Key expires at Di 15 Apr 2025 12:39:00 CEST
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  rsa4096/EB93D1E94EF29562
     created: 2024-04-15  expires: 2025-04-15  usage: C
     trust: ultimate      validity: ultimate
ssb  rsa4096/0E6F5B37473B0B62
     created: 2024-04-15  expires: 2025-04-15  usage: SR
ssb  rsa4096/D6F5B508A50B13E3
     created: 2024-04-15  expires: 2025-04-15  usage: ER
[ultimate] (1). clerie (test 2024-04-15) <clerie@clerie.de>

gpg> addkey
Please select what kind of key you want:
   (3) DSA (sign only)
   (4) RSA (sign only)
   (5) Elgamal (encrypt only)
   (6) RSA (encrypt only)
   (7) DSA (set your own capabilities)
   (8) RSA (set your own capabilities)
  (10) ECC (sign only)
  (11) ECC (set your own capabilities)
  (12) ECC (encrypt only)
  (13) Existing key
  (14) Existing key from card
Your selection? 8
```

```
Possible actions for this RSA key: Sign Encrypt Authenticate
Current allowed actions: Sign Encrypt

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
   (A) Toggle the authenticate capability
   (Q) Finished

Your selection? s

Possible actions for this RSA key: Sign Encrypt Authenticate
Current allowed actions: Encrypt

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
   (A) Toggle the authenticate capability
   (Q) Finished

Your selection? e

Possible actions for this RSA key: Sign Encrypt Authenticate
Current allowed actions:

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
   (A) Toggle the authenticate capability
   (Q) Finished

Your selection? a

Possible actions for this RSA key: Sign Encrypt Authenticate
Current allowed actions: Authenticate

   (S) Toggle the sign capability
   (E) Toggle the encrypt capability
   (A) Toggle the authenticate capability
   (Q) Finished

Your selection?
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) 1y
Key expires at Di 15 Apr 2025 12:39:33 CEST
```

```
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  rsa4096/EB93D1E94EF29562
     created: 2024-04-15  expires: 2025-04-15  usage: C
     trust: ultimate      validity: ultimate
ssb  rsa4096/0E6F5B37473B0B62
     created: 2024-04-15  expires: 2025-04-15  usage: SR
ssb  rsa4096/D6F5B508A50B13E3
     created: 2024-04-15  expires: 2025-04-15  usage: ER
ssb  rsa4096/E1832F5AAE448C84
     created: 2024-04-15  expires: 2025-04-15  usage: AR
[ultimate] (1). clerie (test 2024-04-15) <clerie@clerie.de>

gpg> save
```

Update expiration date

GPG keys should expire in not later than two years. The expiration dates can be changed any time, receivers of the public key just have to update it as soon it expires.

```
[clerie@krypton ~]$ gpg --edit-key A1EF35BC29E3A6E55F3CFBDCEB93D1E94EF29562
gpg (GnuPG) 2.4.5; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Secret key is available.

gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   8  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 8u
gpg: next trustdb check due at 2025-04-15
sec  rsa4096/EB93D1E94EF29562
     created: 2024-04-15  expires: 2025-04-15  usage: C
     trust: ultimate      validity: ultimate
[ultimate] (1). clerie (test 2024-04-15) <clerie@clerie.de>

gpg> expire
Changing expiration time for the primary key.
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 Mo 22 Apr 2024 12:34:35 CEST
Is this correct? (y/N) y

sec  rsa4096/EB93D1E94EF29562
     created: 2024-04-15  expires: 2024-04-22  usage: C
     trust: ultimate      validity: ultimate
[ultimate] (1). clerie (test 2024-04-15) <clerie@clerie.de>

gpg> save
```

Show keys

```
[clerie@krypton ~]$ gpg --list-keys A1EF35BC29E3A6E55F3CFBDCEB93D1E94EF29562
pub   rsa4096 2024-04-15 [C] [expires: 2025-04-15]
      A1EF35BC29E3A6E55F3CFBDCEB93D1E94EF29562
uid           [ultimate] clerie (test 2024-04-15) <clerie@clerie.de>
sub   rsa4096 2024-04-15 [SR] [expires: 2025-04-15]
sub   rsa4096 2024-04-15 [ER] [expires: 2025-04-15]
sub   rsa4096 2024-04-15 [AR] [expires: 2025-04-15]
```

Export keys

Export secret key with subkeys

```
[clerie@krypton ~]$ gpg --export-secret-keys --armor
A1EF35BC29E3A6E55F3CFBDCEB93D1E94EF29562 > secret-key.asc
```

Export secret subkeys only

```
[clerie@krypton ~]$ gpg --export-secret-subkeys --armor
A1EF35BC29E3A6E55F3CFBDCEB93D1E94EF29562 > secret-subkeys.asc
```

Export public key

```
[clerie@krypton ~]$ gpg --export --armor
A1EF35BC29E3A6E55F3CFBDCEB93D1E94EF29562 > public-key.asc
```

Import keys

Public as well as private keys

```
gpg2 --import file.asc
```

SSH

Export SSH public key

```
[clerie@krypton ~]$ gpg --export-ssh-key
A1EF35BC29E3A6E55F3CFBDCEB93D1E94EF29562
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQC3VDWw6oCPS+ztxJ4ROwnWqDrVkhlmVl8ycRXbIT+/pWeb
DX+z0eqYddSyMIKv3lPMw0qeZCERTp/vDfrWVbSJ+65E2MrGJyy0icBezT9qQBG/WaR/ESZpfckr
ZBle9SkXe1Ftxvl/RsrZz2p9+Xnzz3ZI5gh2tFThEVJkPKypZuofPSzazDP+iDni7MXr9l5Heey4
5t6GZB8RY5+JMxGL7/AE8c+I6+Y7fe7crdDGT0AD0wxyHcDO2GaM3SPtawtOjYjbTkiNI0IiHlAp
S+u8cmOpXTrqIupt6/w5a7Aq9Hua3rPk2w5oWvX9a8Jve69s16ohpHDBYyZ3wOL67XHd7/g00C1K
8bP8jIO5D7DhgPZR71SCJ5tv0GHhctjjUDxZIZnzI+3/2tvyvmes0JjGX2uMA36WkEZO1L8mfFYI
OcFa0NqJoyGrHZRFZ3vto9SrpmABe/gtRi7v9Hh5AnW9uVGMVdywapSJ0LTEGzxD9aPxYyiYcrO8
QR32wVfGdV2d0fkEoo8/1308byvlnqOV1kZT+G0cHC/tSKjMZSk1FyJgD5WcSS90oVHjQc+j9xGC
o/LhFI/ALNE5ZbrijEguoXaDjw8PKgGS6V5UWFeFtHkfMZWSjo2kOFuhpydwIrsiWNPekG70HRjt
q24RHp5Y5bPFWOAUfnv/0T2yj/otlw== openpgp:0xAE448C84
```

Links

- https://riseup.net/en/security/message-security/openpgp/best-practices