

GPG

How to generate and use GPG Keys.

- [Generate Key](#)
- [Add Keys to Github/Gitea](#)
- [Telling Git about your Key](#)

Generate Key

1. `gpg --full-generate-key`
2. Select defaults or the required option if you have them
3. Select an expiration date
4. Enter your personal information (it needs to be consistent with your account info)
5. Use a secure passphrase

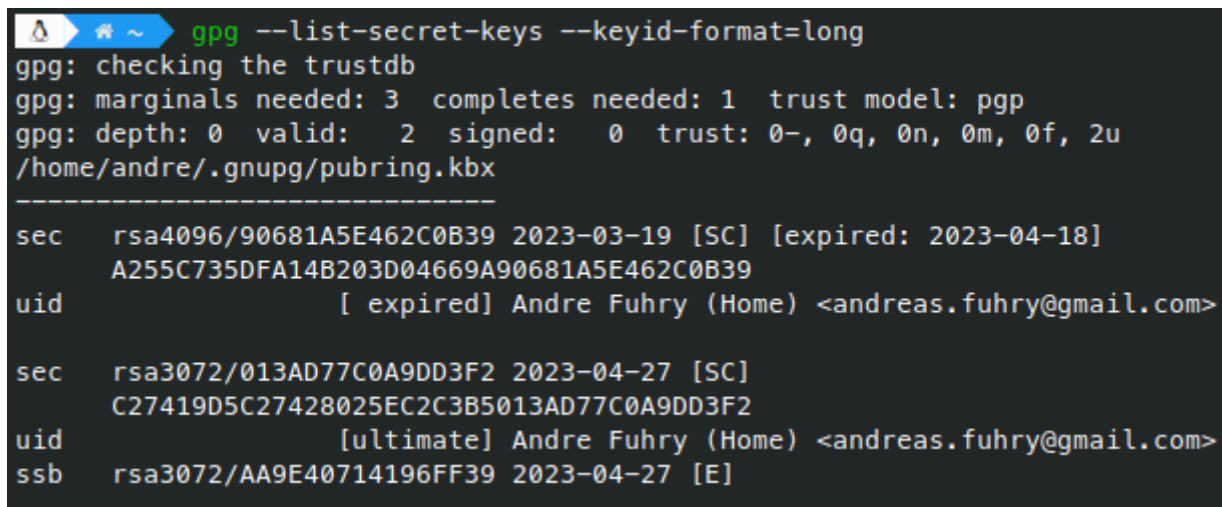
Key can be listed with:

```
gpg --list-secret-keys --keyid-format=long
```

Key can exported with:

```
gpg --armor --export <key>
```

Screenshots:



```
gpg --list-secret-keys --keyid-format=long
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid: 2  signed: 0  trust: 0-, 0q, 0n, 0m, 0f, 2u
/home/andre/.gnupg/pubring.kbx
-----
sec   rsa4096/90681A5E462C0B39 2023-03-19 [SC] [expired: 2023-04-18]
      A255C735DFA14B203D04669A90681A5E462C0B39
uid   [ expired] Andre Fuhry (Home) <andreas.fuhry@gmail.com>

sec   rsa3072/013AD77C0A9DD3F2 2023-04-27 [SC]
      C27419D5C27428025EC2C3B5013AD77C0A9DD3F2
uid   [ultimate] Andre Fuhry (Home) <andreas.fuhry@gmail.com>
ssb   rsa3072/AA9E40714196FF39 2023-04-27 [E]
```

```
gpg --armor --export 013AD77C0A9DD3F2
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
mQGNBGRKtmcBDADGKCT8ul4x0jp4QdNRT0EKHl0H0ZgVL7aZua2LQPctMHFqT7qW
bJax1i5uQUtRTcEnpYIxjYe23vmwMKrdwDxwhm0khpRe43121dYr5QHDQeK9gAcF
EAdyjaKnYNlrloWjGQl2Uggu0ZuRmTB8fVYV0MR57GUosnLoSNYxn0oR1p/IY0+r
zjwrKITuMIpYwrpYg/H0Y8SdbiHdgdDBmA5M/9PGtMN5rjrSGQdr3qvC90D2VpDo
/uKNe6R/UU6k29W0CqPWGnm/sDQzIca7UTF9AALYrzLMjRJ4njaT88e++0D3hNIC
oZS9+o2eVqqPlgXEPXLXkb+bnpMl/UTu07UCi/z7E8Zpg9vLTHEwpWIRgmvXKAzr
PWhZxXbVfuvaUhs6I9ag69KRPhpNu3xz3dRwWB6bxGcMPPBFtvbGagXJdoqPU0PF
Hq0Fc2T1oIUHXLfpSafHXneQ0kwyiCurxn5ggv7vJiqwB4d8LLhKDzSilsj8yIsW
+Do7DymSg8vUeqkAEQEAAbQsQW5kcmUgRnVocnkgKEhvbWUpIDxhbmRyZWZzLmZ1
aHJ5QGdtYWlsLmNvbT6JAc4EEwEIAHgWIQTcdBnVwnQoAl7Cw7UB0td8Cp3T8gUC
ZEq2ZwIbAwULCQgHAgYVCgkICwIEFgIDAQIeAQIXgAAKCRAB0td8Cp3T8jX0C/oC
```

Add Keys to Github/Gitea

Github

1. [Settings](#)
 2. [SSH and GPG keys](#)
 3. New GPG Key
 4. Paste in the public key block
 5. Add GPG Key
 6. Turn on Vigilant mode
-

Gitea

1. [Settings](#)
2. [SSH / GPG Keys](#)
3. Add Key
4. Paste in the public key block
5. Add Key
6. Verify

Screenshot:



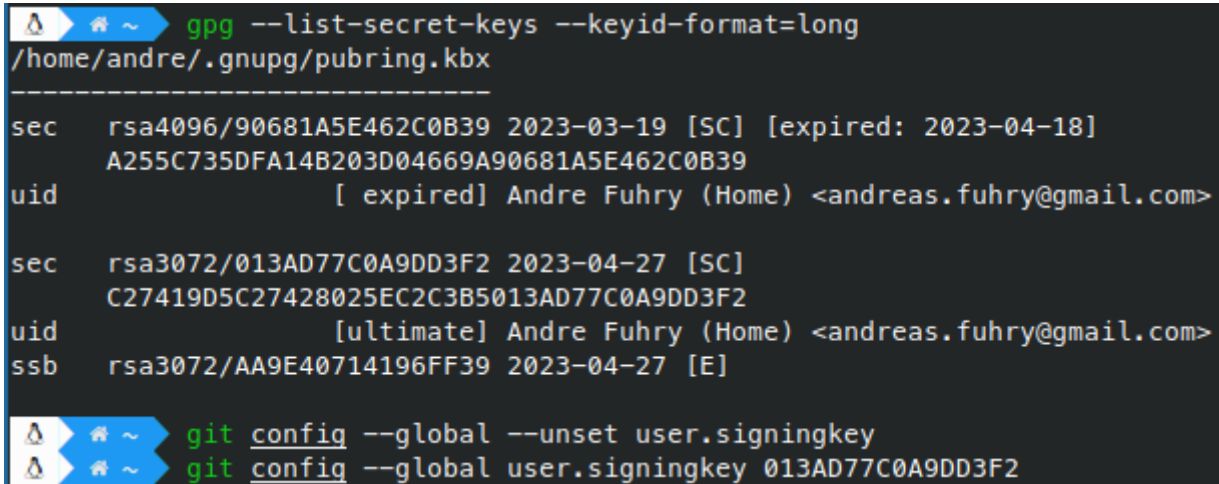
```
echo | gpg -a --default-key [redacted] --detach-sig
gpg: using [redacted] as default secret key for signing
-----BEGIN PGP SIGNATURE-----
[redacted]
-----END PGP SIGNATURE-----
```

Telling Git about your Key

```
gpg --list-secret-keys --keyid-format=long
```

```
git config --global user.signingkey <key>
```

Screenshot:



A terminal window showing the output of the `gpg --list-secret-keys --keyid-format=long` command. The output lists two keys for Andre Fuhry. The first key is expired, and the second key is the current one. Below the output, two `git config` commands are shown: one to unset the signing key and another to set it to the key ID of the second key.

```
gpg --list-secret-keys --keyid-format=long
/home/andre/.gnupg/pubring.kbx
-----
sec   rsa4096/90681A5E462C0B39 2023-03-19 [SC] [expired: 2023-04-18]
      A255C735DFA14B203D04669A90681A5E462C0B39
uid   [ expired] Andre Fuhry (Home) <andreas.fuhry@gmail.com>

sec   rsa3072/013AD77C0A9DD3F2 2023-04-27 [SC]
      C27419D5C27428025EC2C3B5013AD77C0A9DD3F2
uid   [ultimate] Andre Fuhry (Home) <andreas.fuhry@gmail.com>
ssb   rsa3072/AA9E40714196FF39 2023-04-27 [E]

git config --global --unset user.signingkey
git config --global user.signingkey 013AD77C0A9DD3F2
```