Lab Exercise 4- Signed Commits in Git and GitHub

Objective:

To configure Git to sign commits with GPG, push them to GitHub, and verify commit authenticity for secure code contribution.

Prerequisites:

- Git installed on your system
- GPG (GNU Privacy Guard) installed and configured
- GitHub account with a repository (you own or have write access to)
- Basic knowledge of Git commands

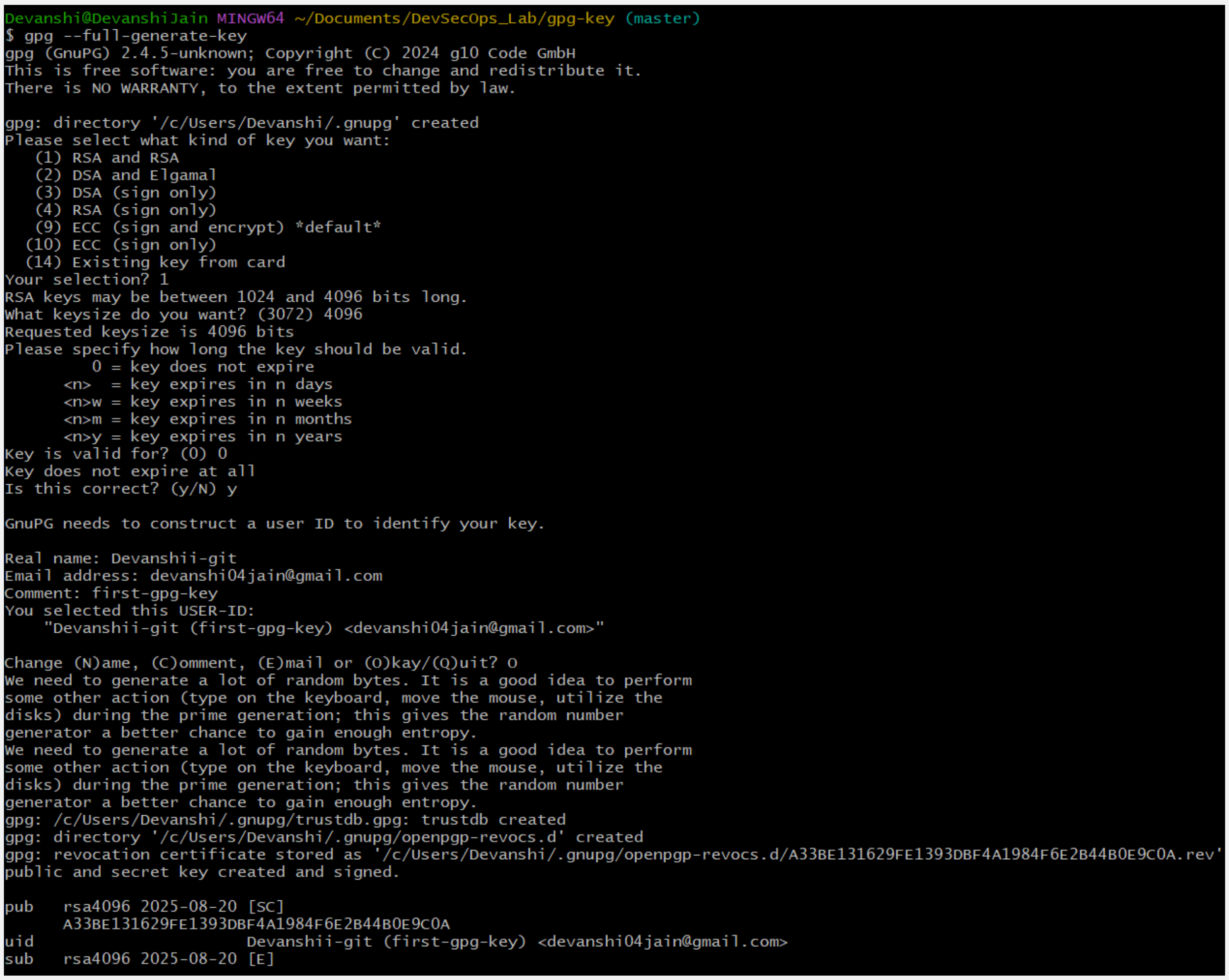
Step 1 – Generate or Use an Existing GPG Key

1. **Check for existing keys**

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

2. **If no key exists, generate a new one**

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

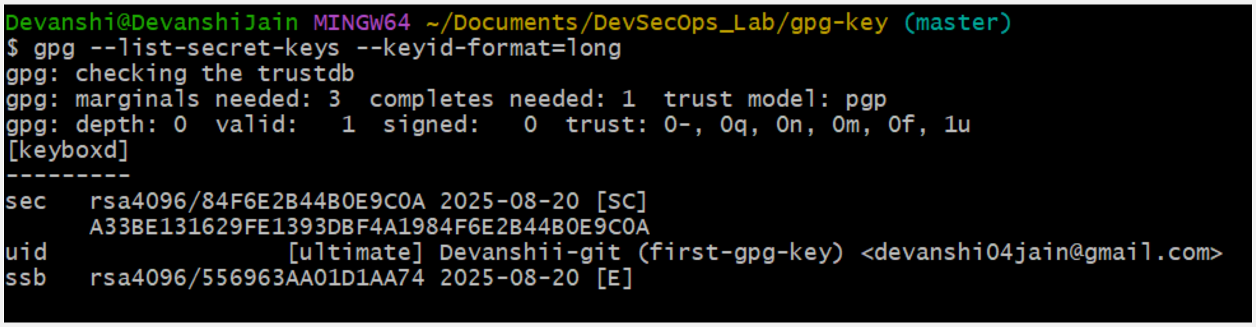


```
Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ gpg --full-generate-key
gpg (GnuPG) 2.4.5-unknown; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/c/Users/Devanshi/.gnupg' created
Please select what kind of key you want:
   (1) RSA and RSA
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
   (9) ECC (sign and encrypt) *default*
  (10) ECC (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) 0
Key does not expire at all
Is this correct? (y/N) y

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

Real name: Devanshii-git
Email address: devanshi04jain@gmail.com
Comment: first-gpg-key
You selected this USER-ID:
    "Devanshii-git (first-gpg-key) <devanshi04jain@gmail.com>"

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: /c/Users/Devanshi/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/c/Users/Devanshi/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/c/Users/Devanshi/.gnupg/openpgp-revocs.d/A33BE131629FE1393DBF4A1984F6E2B44B0E9C0A.rev'
public and secret key created and signed.

pub   rsa4096 2025-08-20 [SC]
      A33BE131629FE1393DBF4A1984F6E2B44B0E9C0A
uid                      Devanshii-git (first-gpg-key) <devanshi04jain@gmail.com>
sub   rsa4096 2025-08-20 [E]
```

- Select **RSA and RSA**
- Key size: **4096**
- Expiration: **0** (never) or a fixed date
- Enter your **GitHub-registered name and email**

3. **Get your key ID**

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



```
Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ 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:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
[keyboxd]
---------
sec   rsa4096/84F6E2B44B0E9C0A 2025-08-20 [SC]
      A33BE131629FE1393DBF4A1984F6E2B44B0E9C0A
uid                 [ultimate] Devanshii-git (first-gpg-key) <devanshi04jain@gmail.com>
ssb   rsa4096/556963AA01D1AA74 2025-08-20 [E]
```

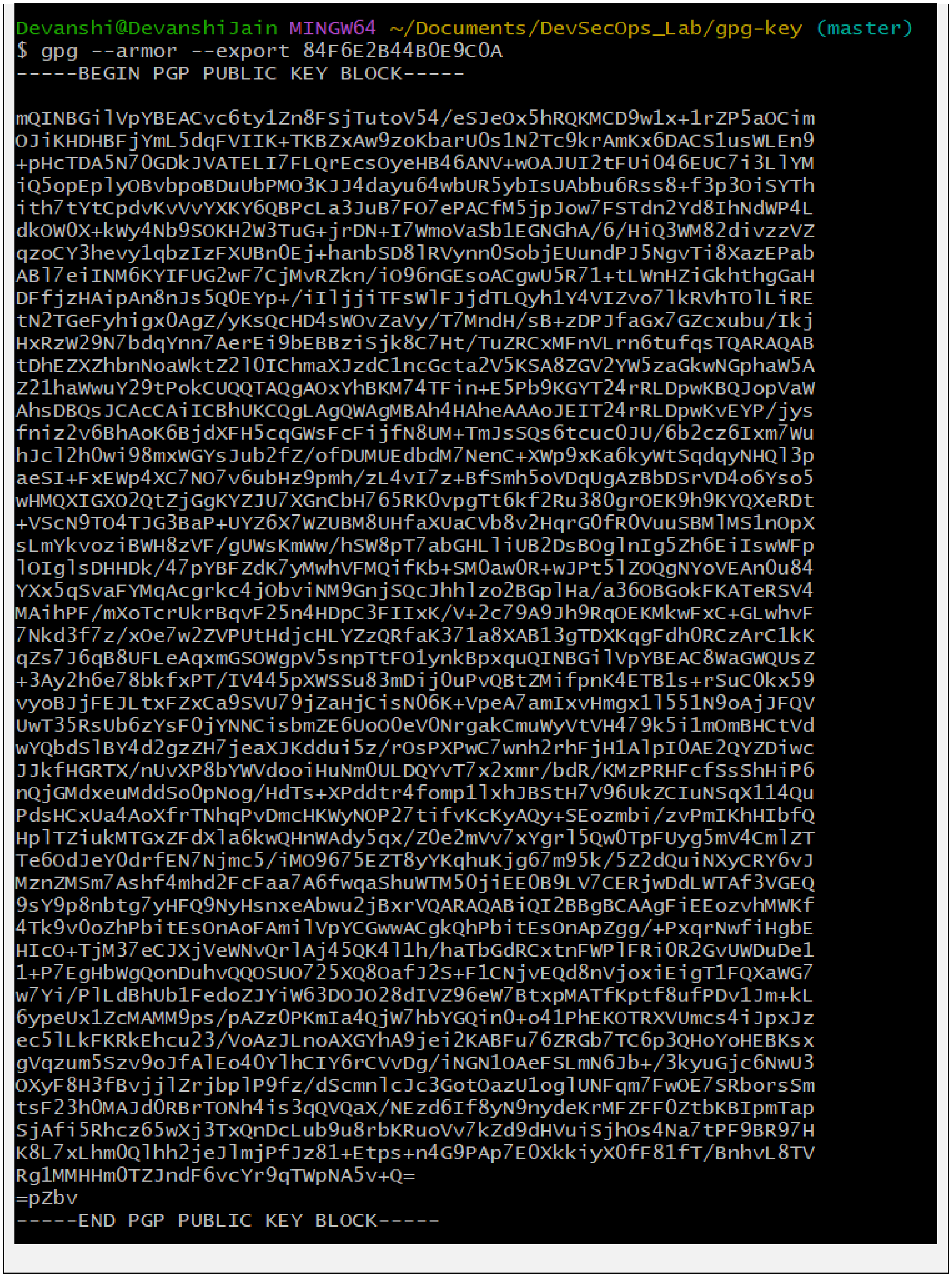
Example output:

```
sec   rsa4096/3AA5C34371567BD2 2025-08-13 [SC]

Here, 3AA5C34371567BD2 is your key ID.
```

Step 2 – Add GPG Key to GitHub

1. Export your public key:

```
gpg --armor --export YOUR_KEY_ID
```



```
Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ gpg --armor --export 84F6E2B44B0E9C0A
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

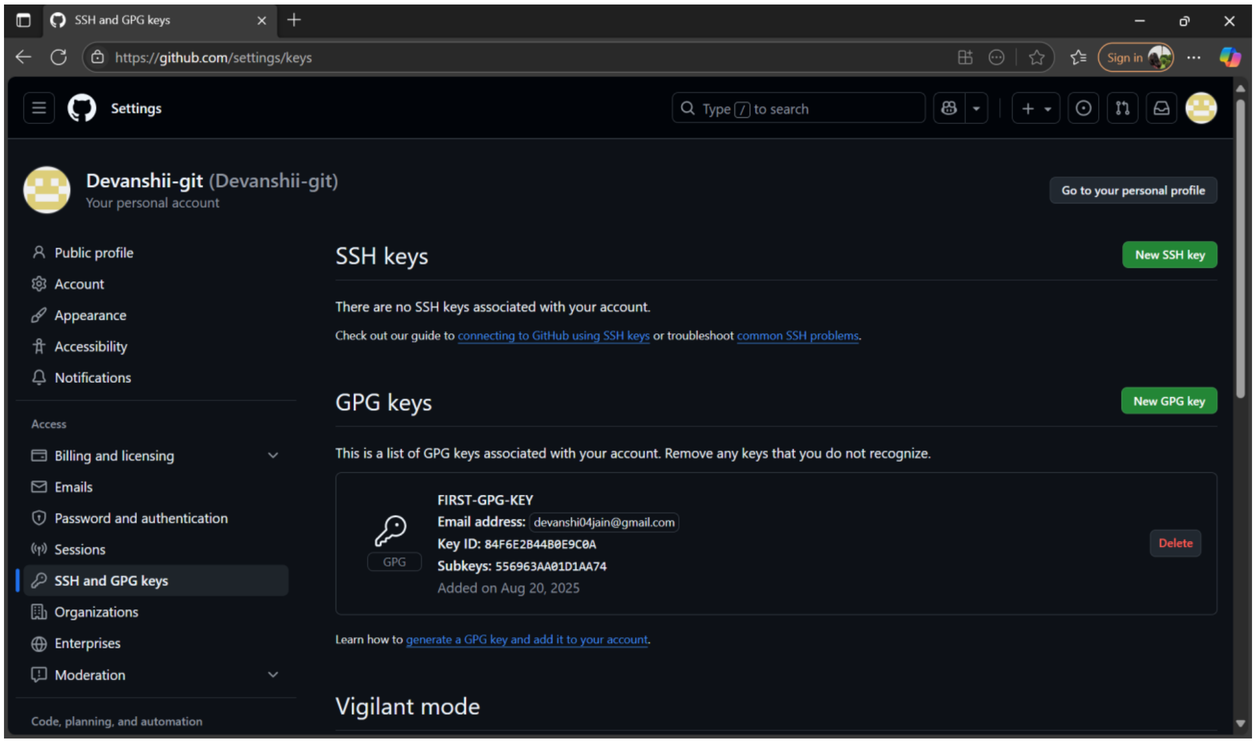
```
mQINBGilVpYBEACvc6ty1Zn8FSjTutoV54/eSJeOx5hRQKMCD9w1x+1rZP5aOCim
OJiKHDHBFjYmL5dqFVIIK+TKBZxAw9zoKbarU0s1N2Tc9krAmKx6DACS1usWLEn9
+pHcTDA5N70GDkJVATELI7FLQrEcsOyeHB46ANV+wOAJUI2tFUi046EUC7i3LlYM
iQ5opEplyOBvbpoBDuUbPMO3KJJ4dayu64wbUR5ybIsUAbbu6Rss8+f3p3OiSYTh
ith7tYtCpdvKvVvYXKY6QBPcLa3JuB7FO7ePACfM5jpJow7FSTdn2Yd8IhNdWP4L
dkOW0X+kWy4Nb9SOKH2W3TuG+jrDN+I7WmoVaSb1EGNGhA/6/HiQ3WM82divzzVZ
qzoCY3hevy1qbzIzFXUBn0Ej+hanbSD8lRVynn0SobjEUundPJ5NgvTi8XazEPab
ABl7eiINM6KYIFUG2wF7CjMvRZkn/iO96nGEsoACgwU5R71+tLWnHZiGkhthgGaH
DFfjzHAipAn8nJs5Q0EYp+/iIljjiTFsWlFJjdTLQyh1Y4VIZvo7lkRVhTOlLiRE
tN2TGeFyhigx0AgZ/yKsQcHD4sWOvZaVy/T7MndH/sB+zDPJfaGx7GZcxubu/Ikj
HxRzW29N7bdqYnn7AerEi9bEBBziSjk8C7Ht/TuZRCxMFnVLrn6tufqsTQARAQAB
tDhEZXZhbnNoaWktZ2l0IChmaXJzdC1ncGcta2V5KSA8ZGV2YW5zaGkwNGphaW5A
Z21haWwuY29tPokCUQQTAQgAOxYhBKM74TFin+E5Pb9KGYT24rRLDpwKBQJopVaW
AhsDBQsJCAcCAiICBhUKCQgLAgQWAgMBAh4HAheAAAoJEIT24rRLDpwKvEYP/jys
fniz2v6BhAoK6BjdXFH5cqGWsFcFijfN8UM+TmJsSQs6tcuc0JU/6b2cz6Ixm7Wu
hJcl2h0wi98mxWGYsJub2fZ/ofDUMUEdbdM7NenC+XWp9xKa6kyWtSqdqyNHQl3p
aeSI+FxEWp4XC7NO7v6ubHz9pmh/zL4vI7z+BfSmh5oVDqUgAzBbDSrVD4o6Yso5
wHMQXIGXO2QtZjGgKYZJU7XGnCbH765RK0vpgTt6kf2Ru380grOEK9h9KYQXeRDt
+VScN9TO4TJG3BaP+UYZ6X7WZUBM8UHfaXUaCVb8v2HqrG0fR0VuuSBMlMS1nOpX
sLmYkvoziBWH8zVF/gUWsKmWw/hSW8pT7abGHLliUB2DsBOglnIg5Zh6EiIswWFp
lOIglsDHHDk/47pYBFZdK7yMwhVFMQifKb+SM0aw0R+wJPt5lZOQgNYoVEAn0u84
YXx5qSvaFYMqAcgrkc4jObviNM9GnjSQcJhhlzo2BGplHa/a36OBGokFKATeRSV4
MAihPF/mXoTcrUkrBqvF25n4HDpC3FIIxK/V+2c79A9Jh9RqOEKMkwFxC+GLwhvF
7Nkd3f7z/xOe7w2ZVPUtHdjcHLYZzQRfaK371a8XAB13gTDXKqgFdh0RCzArC1kK
qZs7J6qB8UFLeAqxmGSOWgpV5snpTtFO1ynkBpxquQINBGilVpYBEAC8WaGWQUsZ
+3Ay2h6e78bkfxPT/IV445pXWSSu83mDij0uPvQBtZMifpnK4ETB1s+rSuC0kx59
vyoBJjFEJLtxFZxCa9SVU79jZaHjCisN06K+VpeA7amIxvHmgx1l551N9oAjJFQV
UwT35RsUb6zYsF0jYNNCisbmZE6UoO0eV0NrgakCmuWyVtVH479k5i1mOmBHCtVd
wYQbdSlBY4d2gzZH7jeaXJKddui5z/rOsPXPwC7wnh2rhFjH1AlpI0AE2QYZDiwc
JJkfHGRTX/nUvXP8bYWVdooiHuNm0ULDQYvT7x2xmr/bdR/KMzPRHFcfSsShHiP6
nQjGMdxeuMddSo0pNog/HdTs+XPddtr4fomp1lxhJBStH7V96UkZCIuNSqX1l4Qu
PdsHCxUa4AoXfrTNhqPvDmcHKWyNOP27tifvKcKyAQy+SEozmbi/zvPmIKhHIbfQ
HplTZiukMTGxZFdXla6kwQHnWAdy5qx/Z0e2mVv7xYgrl5Qw0TpFUyg5mV4CmlZT
Te6OdJeY0drfEN7Njmc5/iMO9675EZT8yYKqhuKjg67m95k/5Z2dQuiNXyCRY6vJ
MznZMSm7Ashf4mhd2FcFaa7A6fwqaShuWTM50jiEE0B9LV7CERjwDdLWTAf3VGEQ
9sY9p8nbtg7yHFQ9NyHsnxeAbwu2jBxrVQARAQABiQI2BBgBCAAgFiEEozvhMWKf
4Tk9v0oZhPbitEsOnAoFAmilVpYCGwwACgkQhPbitEsOnApZgg/+PxqrNwfiHgbE
HIcO+TjM37eCJXjVeWNvQrlAj45QK4l1h/haTbGdRCxtnFWPlFRi0R2GvUWDuDe1
1+P7EgHbWgQonDuhvQQOSUO725XQ8OafJ2S+F1CNjvEQd8nVjoxiEigT1FQXaWG7
w7Yi/PlLdBhUb1FedoZJYiW63DOJO28dIVZ96eW7BtxpMATfKptf8ufPDv1Jm+kL
6ypeUx1ZcMAMM9ps/pAZz0PKmIa4QjW7hbYGQin0+o41PhEKOTRXVUmcs4iJpxJz
ec5lLkFKRkEhcu23/VoAzJLnoAXGYhA9jei2KABFu76ZRGb7TC6p3QHoYoHEBKsx
gVqzum5Szv9oJfAlEo40YlhCIY6rCVvDg/iNGN1OAeFSLmN6Jb+/3kyuGjc6NwU3
OXyF8H3fBvjjlZrjbplP9fz/dScmnlcJc3GotOazU1oglUNFqm7FwOE7SRborsSm
tsF23h0MAJd0RBrTONh4is3qQVQaX/NEzd6If8yN9nydeKrMFZFF0ZtbKBIpmTap
SjAfi5Rhcz65wXj3TxQnDcLub9u8rbKRuoVv7kZd9dHVuiSjhOs4Na7tPF9BR97H
K8L7xLhm0Qlhh2jeJlmjPfJz81+Etps+n4G9PAp7E0XkkiyX0fF81fT/BnhvL8TV
Rg1MMHHm0TZJndF6vcYr9qTWpNA5v+Q=
=pZbv
```

```
-----END PGP PUBLIC KEY BLOCK-----
```

2. Copy the output.

3. Go to **GitHub → Settings → SSH and GPG Keys → New GPG Key**.

4. Paste your key and save.



Devanshii-git (Devanshii-git)
Your personal account

SSH keys

There are no SSH keys associated with your account.

Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.

GPG keys

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.

FIRST-GPG-KEY
Email address: devanshi04jain@gmail.com
Key ID: 84F6E2B44B0E9C0A
Subkeys: 556963AA01D1AA74
Added on Aug 20, 2025

Learn how to generate a GPG key and add it to your account.

Vigilant mode

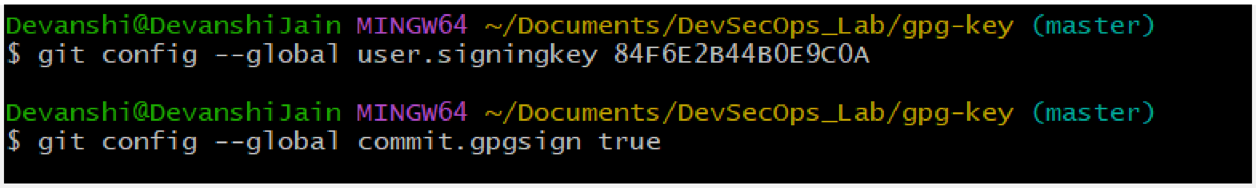
Step 3 – Configure Git for Signed Commits

1. Tell Git which key to use:

```
git config --global user.signingkey YOUR_KEY_ID
```

2. Enable signing for all commits:

```
git config --global commit.gpgsign true
```

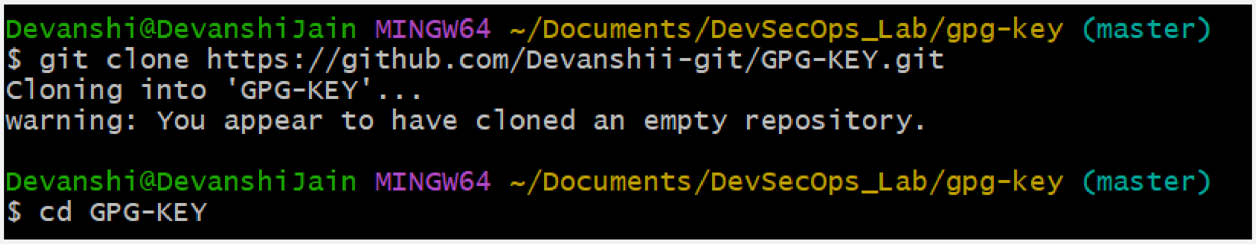


```
Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ git config --global user.signingkey 84F6E2B44B0E9C0A

Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ git config --global commit.gpgsign true
```

Step 4 – Make a Signed Commit

1. Clone your repo (or use an existing one):

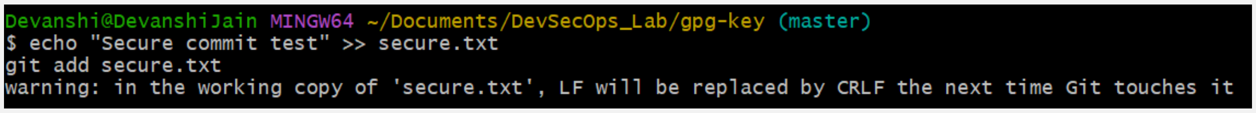
```
git clone https://github.com/<username>/<repository>.git

cd <repository>
```



```
Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ git clone https://github.com/Devanshii-git/GPG-KEY.git
Cloning into 'GPG-KEY'...
warning: You appear to have cloned an empty repository.

Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ cd GPG-KEY
```

2. Edit or create a file:

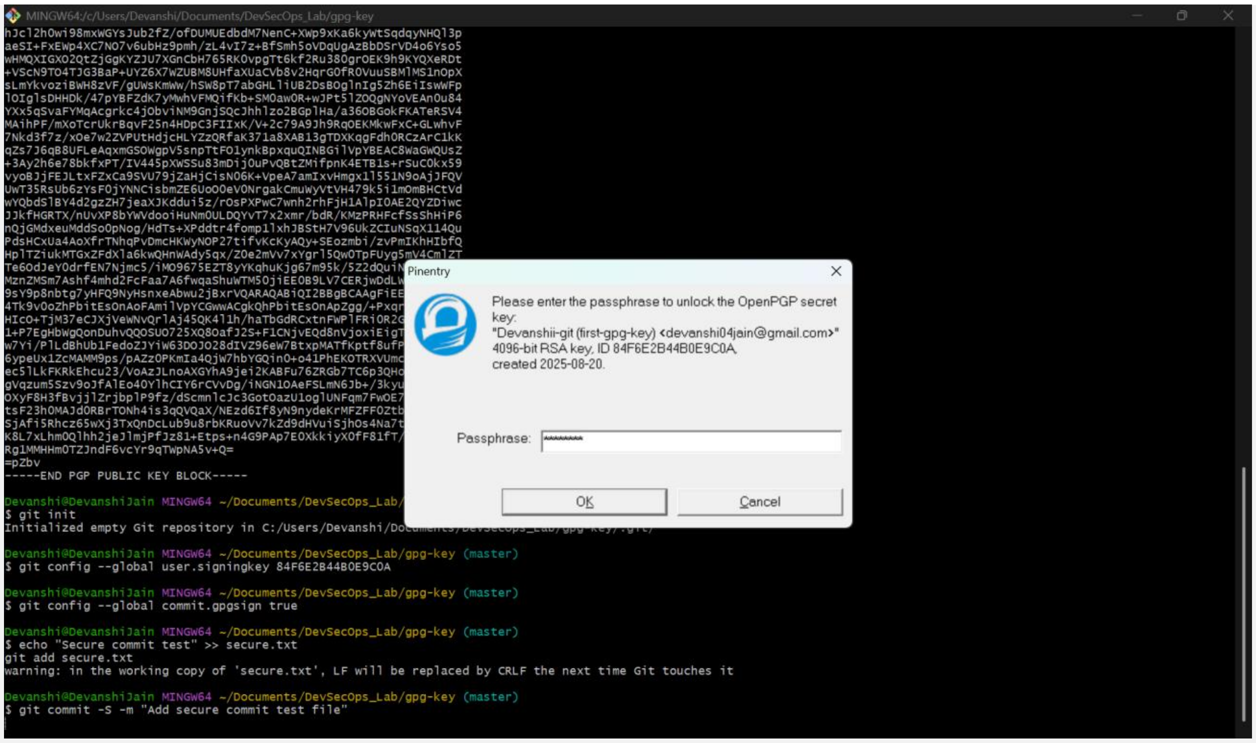
```
echo "Secure commit test" >> secure.txt

git add secure.txt
```



```
Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ echo "Secure commit test" >> secure.txt
git add secure.txt
warning: in the working copy of 'secure.txt', LF will be replaced by CRLF the next time Git touches it
```

3. Commit with signing:

```
git commit -S -m "Add secure commit test file"
```



MINGW64:/c/Users/Devanshi/Documents/DevSecOps_Lab/gpg-key

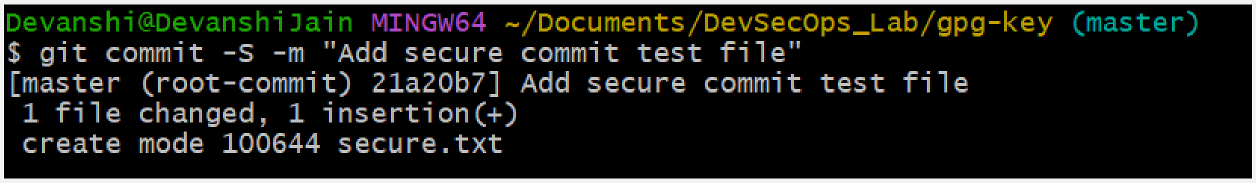
```
hJcl2h0wi98mxWGYsJub2fZ/ofDUMUEdbdM7NenC+XWp9xKa6kyWtSqdqyNHQl3p
aeSI+FxEWp4XC7NO7v6ubHz9pmh/zL4vI7z+BfSmh5oVDqUgAzBbDSrVD4o6Yso5
wHMQXIGXO2QtZjGgKYZJU7XGnCbH765RK0vpgTt6kf2Ru38OgrOEK9h9KYQXeRDt
+VScN9TO4TJG3BaP+UYZ6X7WZUBM8UHfaXUaCVb8v2HqrG0fR0VuuSBMlMS1nOpX
sLmYkvoziBWH8zVF/gUWsKmWw/hSW8pT7abGHLliUB2DsBOglnIg5Zh6EiIswWFp
lOIglsDHHDk/47pYBFZdK7yMwhVFMQifKb+SM0aw0R+wJPt5lZOQgNYoVEAnOu84
YXx5qSvaFYMqAcgrkc4jObviNM9GnjSQcJhhlzo2BGplHa/a36OBGokFKATeRSV4
MAihPF/mXoTcrUkrBqvF25n4HDpC3FIIxK/V+2c79A9Jh9RqOEKMkwFxC+GLwhvF
7Nkd3f7z/xOe7w2ZVPUtHdjcHLYZzQRfaK371a8XAB13gTDXKqgFdh0RCzArC1kK
qZs7J6qB8UFLeAqxmGSOWgpV5snpTtFO1ynkBpxquQINBGilVpYBEAC8WaGWQUsZ
+3Ay2h6e78bkfxPT/IV445pXWSSu83mDijOuPvQBtZMifpnK4ETB1s+rSuC0kx59
vyoBJjFEJLtxFZxCa9SVU79jZaHjCisNO6K+VpeA7amIxvHmgx1l551N9oAjJFQV
UwT35RsUb6zYsFOjYNNCisbmZE6UoO0eVONrgakCmuWyVtVH479k5i1mOmBHCtVd
wYQbdSlBY4d2gzZH7jeaXJKddui5z/rOsPXPwC7wnh2rhFjH1AlpIOAE2QYZDiwc
JJkfHGRTX/nUvXP8bYWVdooiHuNm0ULDQYvT7x2xmr/bdR/KMzPRHFcfSsShHiP6
nQjGMdxeuMddSoOpNog/HdTs+XPddtr4fomp1lxhJBStH7V96UkZCIuNSqX114Qu
PdsHCxUa4AoXfrTNhqPvDmcHKWyNOP27tifvKcKyAQy+SEozmbi/zvPmIKhHIbfQ
HplTZiukMTGxZFdXla6kwQHnWAdy5qx/ZOe2mVv7xYgrl5QwOTpFUyg5mV4CmlZT
Te6OdJeYOdrfEN7Njmc5/iMO9675EZT8yYKqhuKjg67m95k/5Z2dQuiN
MznZMSm7Ashf4mhd2FcFaa7A6fwqaShuWTM5OjiEEOB9LV7CERjwDdLW
9sY9p8nbtg7yHFQ9NyHsnxeAbwu2jBxrVQARAQABiQI2BBgBCAAgFiEE
4Tk9vOoZhPbitEsOnAoFAmilVpYCGwwACgkQhPbitEsOnApZgg/+Pxqr
HIcO+TjM37eCJXjVeWNvQrlAj45QK4l1h/haTbGdRCxtnFWPlFRiOR2G
1+P7EgHbWgQonDuhvQQOSUO725XQ8OafJ2S+F1CNjvEQd8nVjoxiEigT
w7Yi/PlLdBhUb1FedoZJYiW63DOJO28dIVZ96eW7BtxpMATfKptf8ufP
6ypeUx1ZcMAMM9ps/pAZzOPKmIa4QjW7hbYGQin0+o41PhEKOTRXVUmc
ec5lLkFKRkEhcu23/VoAzJLnoAXGYhA9jei2KABFu76ZRGb7TC6p3QHo
gVqzum5Szv9oJfAlEo4OYlhCIY6rCVvDg/iNGN1OAeFSLmN6Jb+/3kyu
OXyF8H3fBvjjlZrjbplP9fz/dScmnlcJc3GotOazU1oglUNFqm7FwOE7
tsF23hOMAJdORBrTONh4is3qQVQaX/NEzd6If8yN9nydeKrMFZFFOZtb
SjAfi5Rhcz65wXj3TxQnDcLub9u8rbKRuoVv7kZd9dHVuiSjhOs4Na7t
K8L7xLhmOQlhh2jeJlmjPfJz81+Etps+n4G9PAp7EOXkkiyXOfF81fT/
Rg1MMHHmOTZJndF6vcYr9qTWpNA5v+Q=
=pZbv
-----END PGP PUBLIC KEY BLOCK-----
```

```
Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/
$ git init
Initialized empty Git repository in C:/Users/Devanshi/Documents/DevSecOps_Lab/gpg-key/.git/

Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ git config --global user.signingkey 84F6E2B44B0E9C0A

Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ git config --global commit.gpgsign true

Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ echo "Secure commit test" >> secure.txt
git add secure.txt
warning: in the working copy of 'secure.txt', LF will be replaced by CRLF the next time Git touches it

Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ git commit -S -m "Add secure commit test file"
```

Pinentry

Please enter the passphrase to unlock the OpenPGP secret key:
"Devanshii-git (first-gpg-key) <devanshi04jain@gmail.com>"
4096-bit RSA key, ID 84F6E2B44B0E9C0A,
created 2025-08-20.

Passphrase: *********



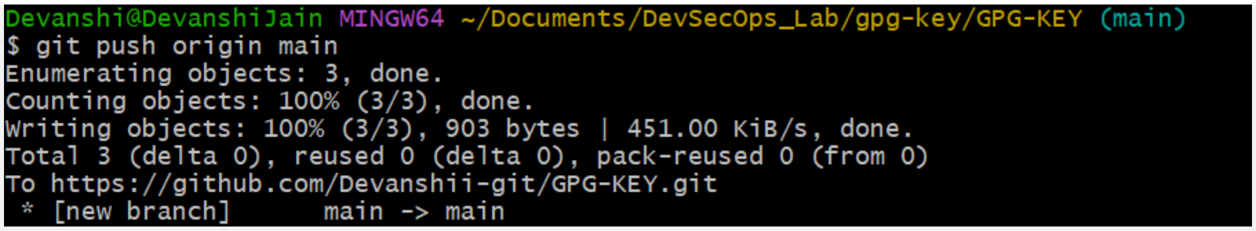
```
Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key (master)
$ git commit -S -m "Add secure commit test file"
[master (root-commit) 21a20b7] Add secure commit test file
 1 file changed, 1 insertion(+)
 create mode 100644 secure.txt
```

4. Enter your GPG passphrase when prompted.

Step 5 – Push and Verify on GitHub

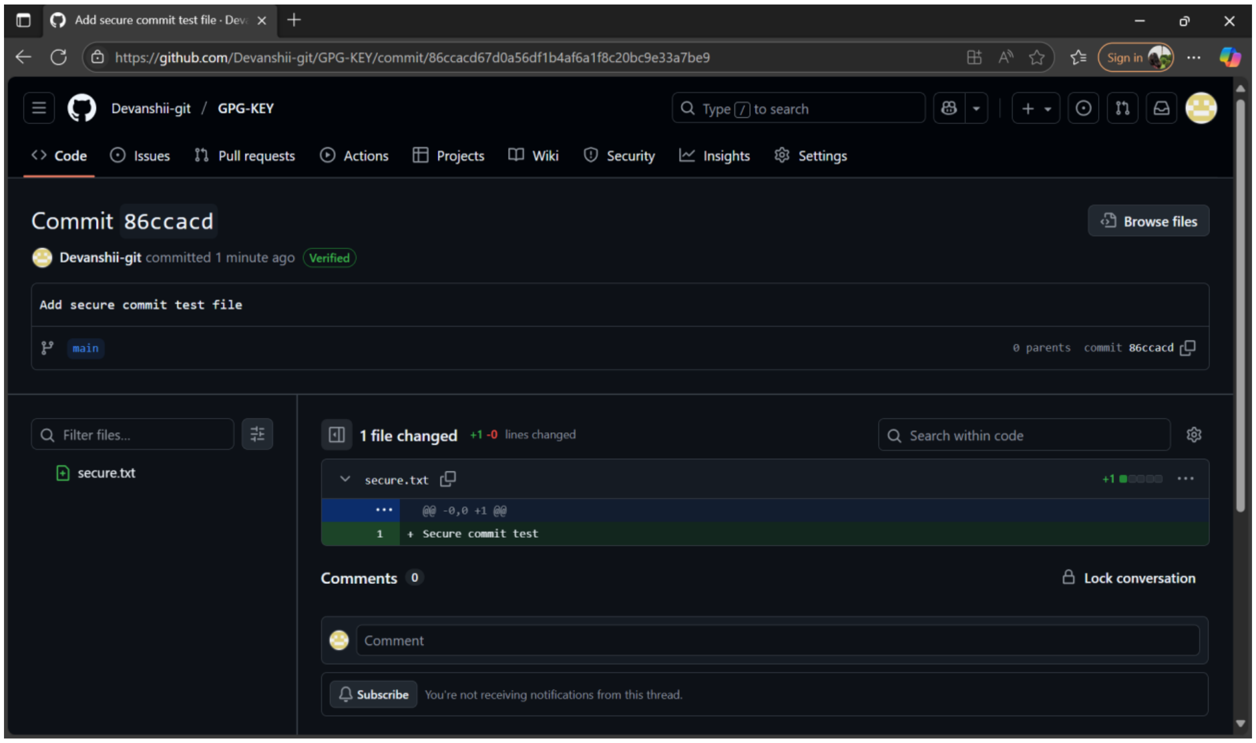
1. Push the commit:

```
git push origin main
```



```
Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key/GPG-KEY (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 903 bytes | 451.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Devanshii-git/GPG-KEY.git
 * [new branch]      main -> main
```

2. Go to your repository on GitHub → Click the commit → You should see a **green “Verified” badge**.



Commit 86ccacd

Devanshii-git committed 1 minute ago Verified

```
Add secure commit test file
main                                              0 parents  commit 86ccacd
```

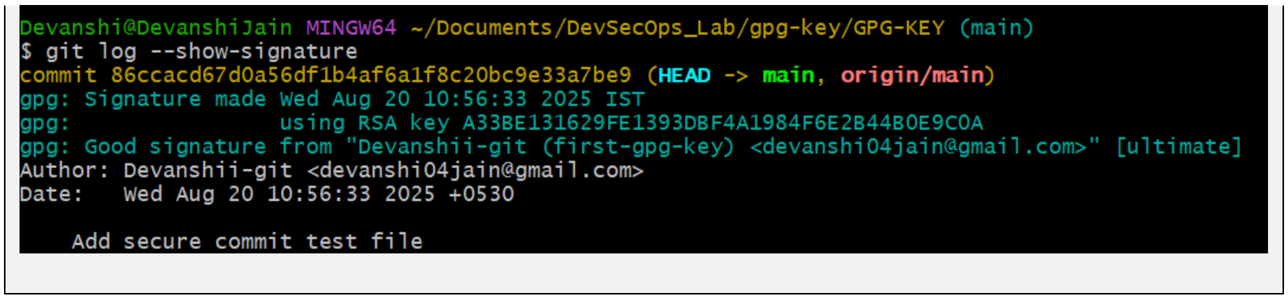
1 file changed +1 -0 lines changed

```
secure.txt                                                              +1
   ...   @@ -0,0 +1 @@
    1  + Secure commit test
```

You're not receiving notifications from this thread.

Step 6 – Local Verification of Commit

```
git log --show-signature
```



```
Devanshi@DevanshiJain MINGW64 ~/Documents/DevSecOps_Lab/gpg-key/GPG-KEY (main)
$ git log --show-signature
commit 86ccacd67d0a56df1b4af6a1f8c20bc9e33a7be9 (HEAD -> main, origin/main)
gpg: Signature made Wed Aug 20 10:56:33 2025 IST
gpg:                using RSA key A33BE131629FE1393DBF4A1984F6E2B44B0E9C0A
gpg: Good signature from "Devanshii-git (first-gpg-key) <devanshi04jain@gmail.com>" [ultimate]
Author: Devanshii-git <devanshi04jain@gmail.com>
Date:   Wed Aug 20 10:56:33 2025 +0530

    Add secure commit test file
```

This will display the GPG verification details locally.

Use Case

Signed commits prevent identity spoofing in collaborative projects, ensuring only verified authors can make trusted changes in critical codebases.