Name – Rachit Shrivastava

Sap Id – 500119571

Batch 3 – DevOps

Lab Exercise 4- Signed Commits in Git and GitHub

Objective:

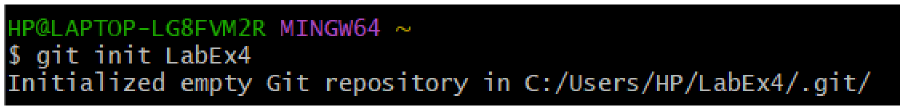
To configure Git to sign commits with GPG, push them to GitHub, and verify commit authenticity for secure code contribution.

Prerequisites:

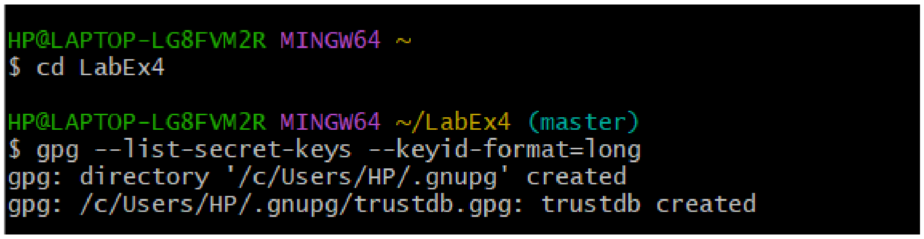
- Git installed on your system
- GPG (GNU Privacy Guard) installed and configured
- GitHub account with a repository (you own or have write access to)
- Basic knowledge of Git commands

➢ Step 1 – Generate or Use an Existing GPG Key

1. **Check for existing keys**



```
HP@LAPTOP-LG8FVM2R MINGW64 ~
$ git init LabEx4
Initialized empty Git repository in C:/Users/HP/LabEx4/.git/
```

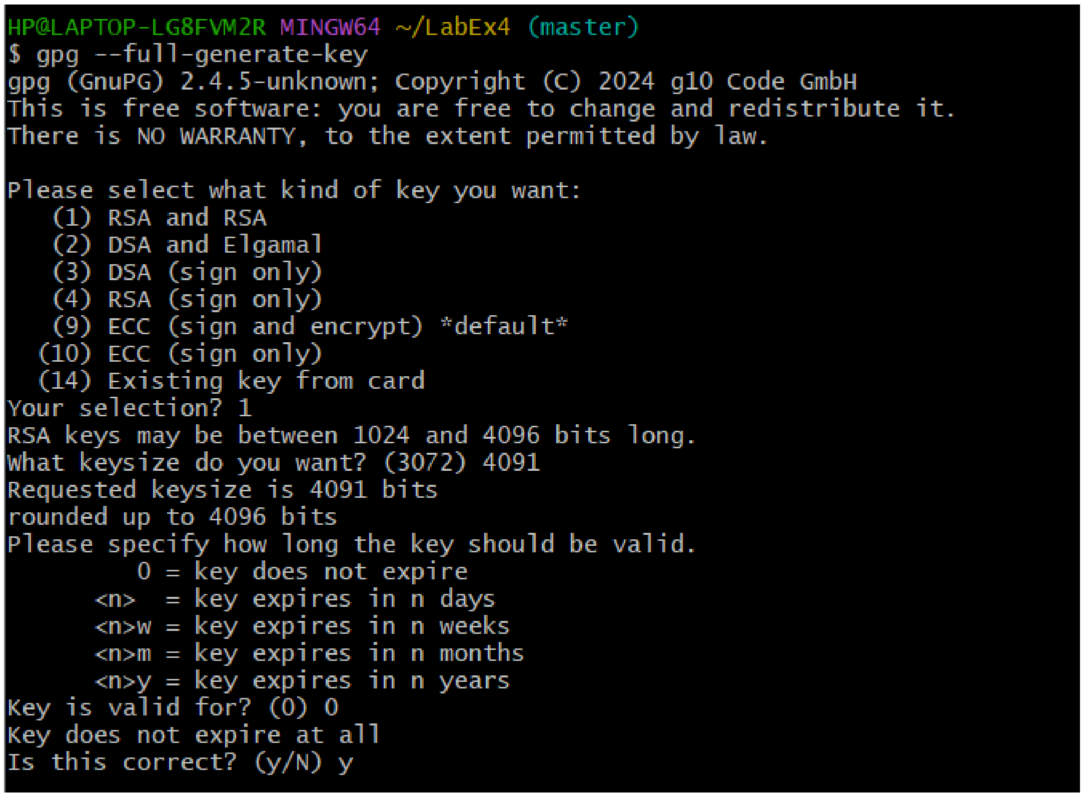


```
HP@LAPTOP-LG8FVM2R MINGW64 ~
$ cd LabEx4

HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ gpg --list-secret-keys --keyid-format=long
gpg: directory '/c/Users/HP/.gnupg' created
gpg: /c/Users/HP/.gnupg/trustdb.gpg: trustdb created
```

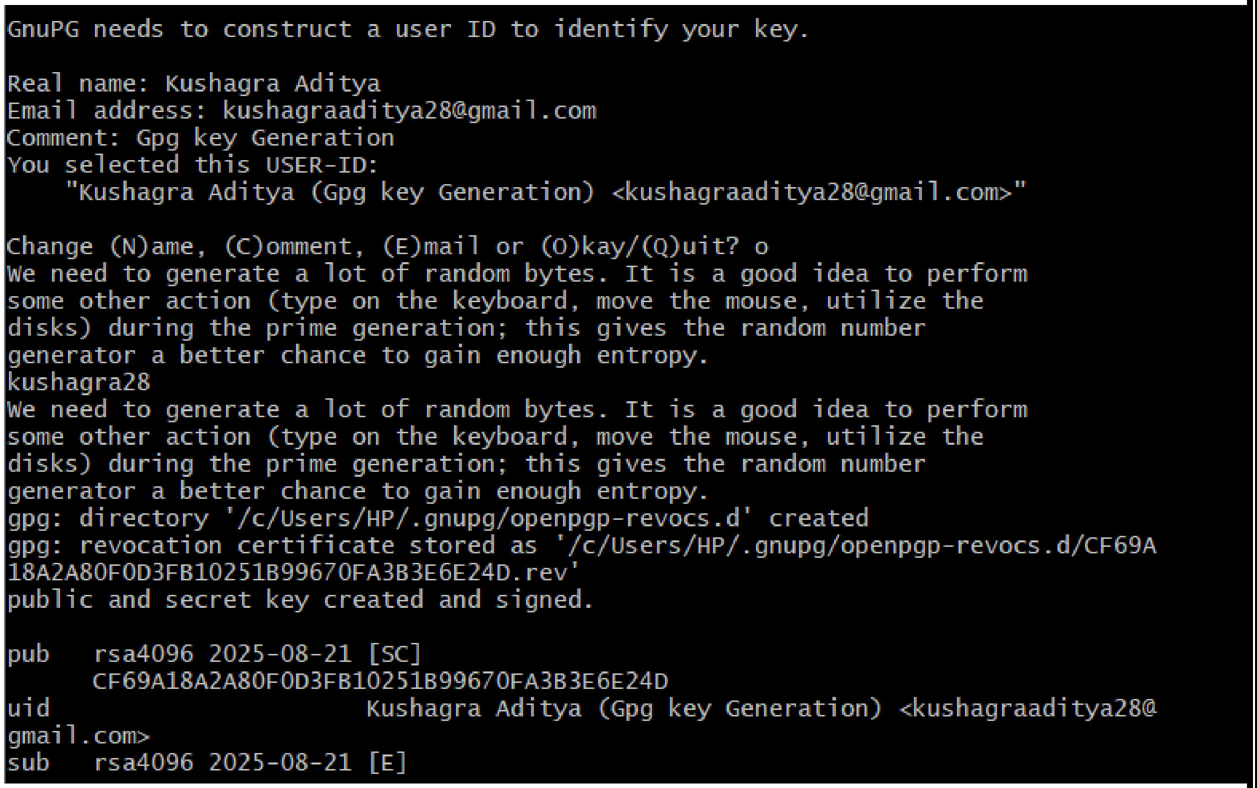
If no key exists, generate a new one

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



```
HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (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.

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) 4091
Requested keysize is 4091 bits
rounded up to 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
```



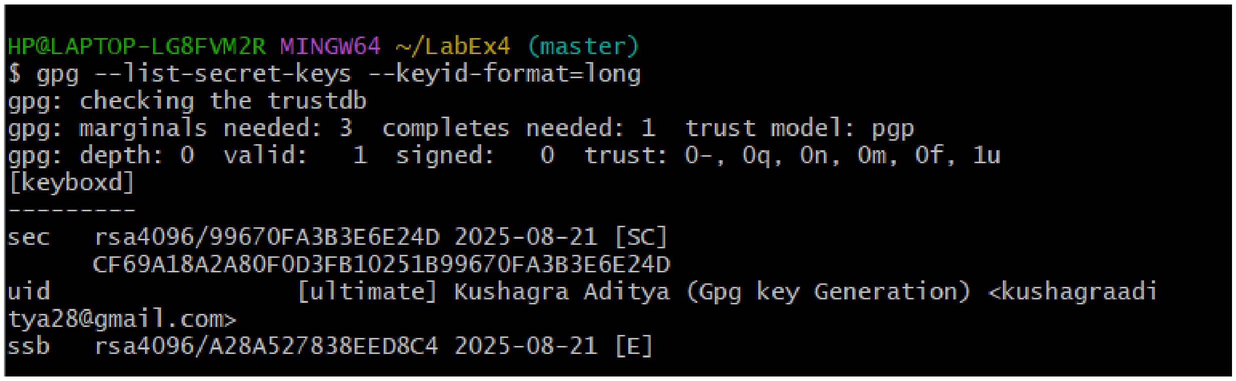
```
GnuPG needs to construct a user ID to identify your key.

Real name: Kushagra Aditya
Email address: kushagraaditya28@gmail.com
Comment: Gpg key Generation
You selected this USER-ID:
    "Kushagra Aditya (Gpg key Generation) <kushagraaditya28@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.
kushagra28
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 '/c/Users/HP/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/c/Users/HP/.gnupg/openpgp-revocs.d/CF69A
18A2A80F0D3FB10251B99670FA3B3E6E24D.rev'
public and secret key created and signed.

pub   rsa4096 2025-08-21 [SC]
      CF69A18A2A80F0D3FB10251B99670FA3B3E6E24D
uid                      Kushagra Aditya (Gpg key Generation) <kushagraaditya28@
gmail.com>
sub   rsa4096 2025-08-21 [E]
```

2. **Get your key ID**



```
HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (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   rsa4096/99670FA3B3E6E24D 2025-08-21 [SC]
      CF69A18A2A80F0D3FB10251B99670FA3B3E6E24D
uid                 [ultimate] Kushagra Aditya (Gpg key Generation) <kushagraadi
tya28@gmail.com>
ssb   rsa4096/A28A527838EED8C4 2025-08-21 [E]
```

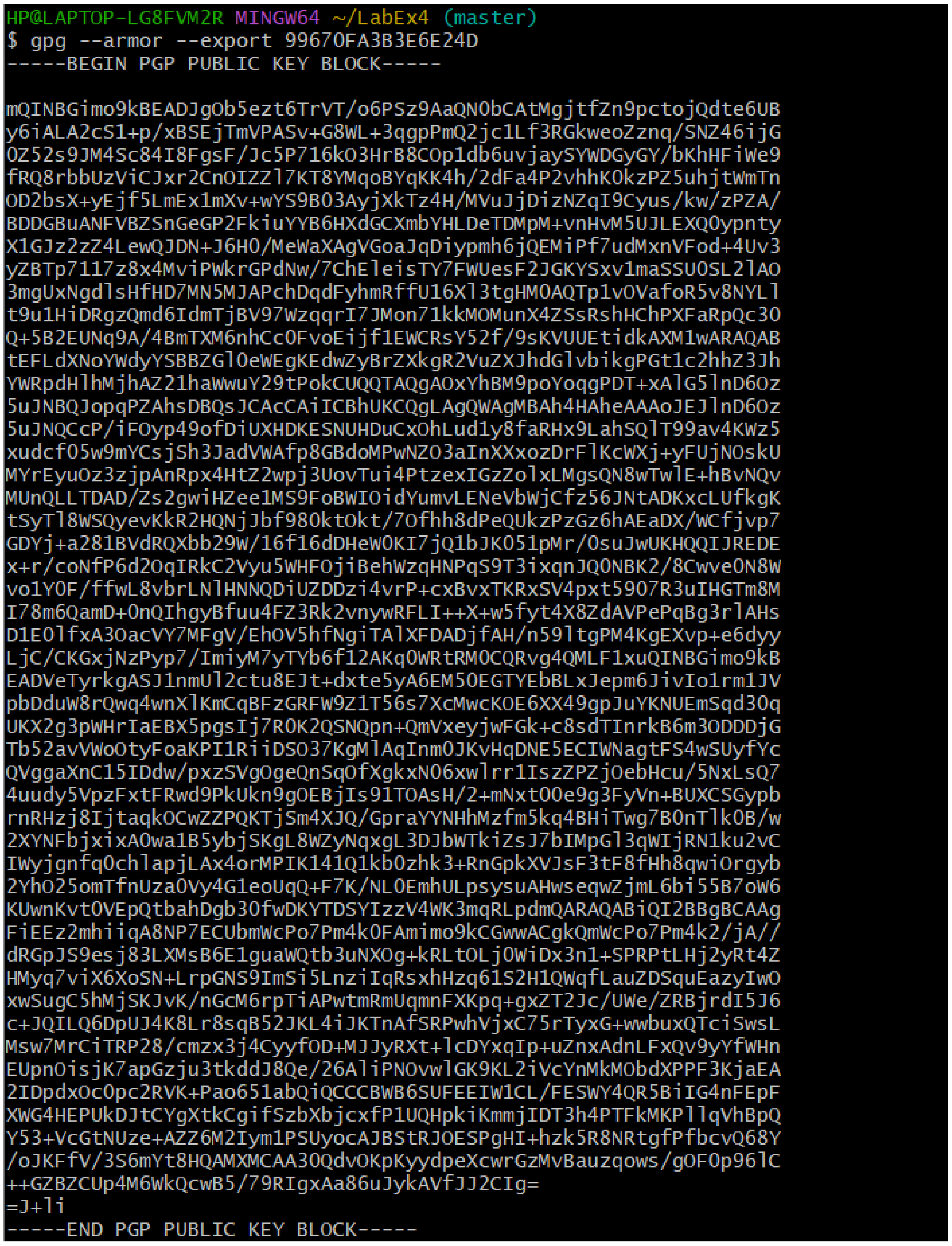
➢ Step 2 – Add GPG Key to GitHub

1. **Export your public key.**

2. **Copy the output.**

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

4. Paste your key and save.



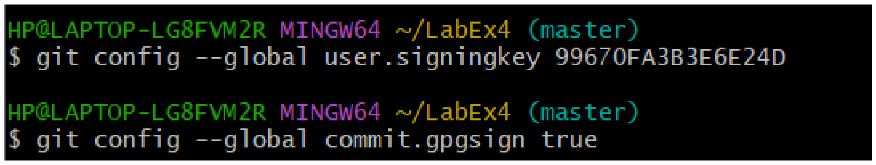
```
HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ gpg --armor --export 99670FA3B3E6E24D
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGimo9kBEADJgOb5ezt6TrVT/o6PSz9AaQNObCAtMgjtfZn9pctojQdte6UB
y6iALA2cS1+p/xBSEjTmVPASv+G8WL+3qgpPmQ2jc1Lf3RGkweoZznq/SNZ46ijG
OZ52s9JM4Sc84I8FgsF/Jc5P716kO3HrB8COp1db6uvjaySYWDGyGY/bKhHFiWe9
fRQ8rbbUzViCJxr2CnOIZZl7KT8YMqoBYqKK4h/2dFa4P2vhhK0kzPZ5uhjtWmTn
OD2bsX+yEjf5LmEx1mXv+wYS9BO3AyjXkTz4H/MVuJjDizNZqI9Cyus/kw/zPZA/
BDDGBuANFVBZSnGeGP2FkiuYYB6HXdGCXmbYHLDeTDMpM+vnHvM5UJLEXQ0ypnty
X1GJz2zZ4LewQJDN+J6H0/MeWaXAgVGoaJqDiypmh6jQEMiPf7udMxnVFod+4Uv3
yZBTp7117z8x4MviPWkrGPdNw/7ChEleisTY7FWUesF2JGKYSxv1maSSU0SL2lAO
3mgUxNgdlsHfHD7MN5MJAPchDqdFyhmRffU16Xl3tgHM0AQTp1vOVafoR5v8NYLl
t9u1HiDRgzQmd6IdmTjBV97WzqqrI7JMon71kkMOMunX4ZSsRshHChPXFaRpQc3O
Q+5B2EUNq9A/4BmTXM6nhCcOFvoEijf1EWCRsY52f/9sKVUUEtidkAXM1wARAQAB
tEFLdXNoYWdyYSBBZGl0eWEgKEdwZyBrZXkgR2VuZXJhdGlvbikgPGt1c2hhZ3Jh
YWRpdHlhMjhAZ21haWwuY29tPokCUQQTAQgAOxYhBM9poYoqgPDT+xAlG5lnD6Oz
5uJNBQJopqPZAhsDBQsJCAcCAiICBhUKCQgLAgQWAgMBAh4HAheAAAoJEJlnD6Oz
5uJNQCcP/iFOyp49ofDiUXHDKESNUHDuCxOhLud1y8faRHx9LahSQlT99av4KWz5
xudcf05w9mYCsjSh3JadVWAfp8GBdoMPwNZO3aInXXxozDrFlKcWXj+yFUjNOskU
MYrEyuOz3zjpAnRpx4HtZ2wpj3UovTui4PtzexIGzZolxLMgsQN8wTwlE+hBvNQv
MUnQLLTDAD/Zs2gwiHZee1MS9FoBWIOidYumvLENeVbWjCfz56JNtADKxcLUfkgK
tSyTl8WSQyevKkR2HQNjJbf980ktOkt/7Ofhh8dPeQUkzPzGz6hAEaDX/WCfjvp7
GDYj+a281BVdRQXbb29W/16f16dDHeW0KI7jQ1bJK051pMr/0suJwUKHQQIJREDE
x+r/coNfP6d2OqIRkC2Vyu5WHFOjiBehWzqHNPqS9T3ixqnJQ0NBK2/8CwveON8W
vo1YOF/ffwL8vbrLNlHNNQDiUZDDzi4vrP+cxBvxTKRxSV4pxt5907R3uIHGTm8M
I78m6QamD+0nQIhgyBfuu4FZ3Rk2vnywRFLI++X+w5fyt4X8ZdAVPePqBg3rlAHs
D1EOlfxA3OacVY7MFgV/EhOV5hfNgiTAlXFDADjfAH/n59ltgPM4KgEXvp+e6dyy
LjC/CKGxjNzPyp7/ImiyM7yTYb6f12AKq0WRtRMOCQRvg4QMLF1xuQINBGimo9kB
EADVeTyrkgASJ1nmUl2ctu8EJt+dxte5yA6EM5OEGTYEbBLxJepm6JivIo1rm1JV
pbDduW8rQwq4wnXlKmCqBFzGRFW9Z1T56s7XcMwcKOE6XX49gpJuYKNUEmSqd3Oq
UKX2g3pWHrIaEBX5pgsIj7ROK2QSNQpn+QmVxeyjwFGk+c8sdTInrkB6m3ODDDjG
Tb52avVWoOtyFoaKPI1RiiDSO37KgMlAqInm0JKvHqDNE5ECIWNagtFS4wSUyfYc
QVggaXnC15IDdw/pxzSVgOgeQnSqOfXgkxNO6xwlrr1IszZPZjOebHcu/5NxLsQ7
4uudy5VpzFxtFRwd9PkUkn9gOEBjIs91TOAsH/2+mNxtO0e9g3FyVn+BUXCSGypb
rnRHzj8IjtaqkOCwZZPQKTjSm4XJQ/GpraYYNHhMzfm5kq4BHiTwg7BOnTlkOB/w
2XYNFbjxixA0wa1B5ybjSKgL8WZyNqxgL3DJbWTkiZsJ7bIMpGl3qWIjRN1ku2vC
IWyjgnfq0chlapjLAx4orMPIK141Q1kb0zhk3+RnGpkXVJsF3tF8fHh8qwiOrgyb
2YhO25omTfnUza0Vy4G1eoUqQ+F7K/NL0EmhULpsysuAHwseqwZjmL6bi55B7oW6
KUwnKvtOVEpQtbahDgb30fwDKYTDSYIzzV4WK3mqRLpdmQARAQABiQI2BBgBCAAg
FiEEz2mhiiqA8NP7ECUbmWcPo7Pm4kOFAmimo9kCGwwACgkQmWcPo7Pm4k2/jA//
dRGpJS9esj83LXMsB6E1guaWQtb3uNXOg+kRLtOLjOWiDx3n1+SPRPtLHj2yRt4Z
HMyq7viX6XoSN+LrpGNS9ImSi5LnziIqRsxhHzq61S2H1QWqfLauZDSquEazyIwO
xwSugC5hMjSKJvK/nGcM6rpTiAPwtmRmUqmnFXKpq+gxZT2Jc/UWe/ZRBjrdI5J6
c+JQILQ6DpUJ4K8Lr8sqB52JKL4iJKTnAfSRPwhVjxC75rTyxG+wwbuxQTciSwsL
Msw7MrCiTRP28/cmzx3j4CyyfOD+MJJyRXt+lcDYxqIp+uZnxAdnLFxQv9yYfWHn
EUpnOisjK7apGzju3tkddJ8Qe/26AliPNOvwlGK9KL2iVcYnMkMObdXPPF3KjaEA
2IDpdxOcOpc2RVK+Pao651abQiQCCCBWB6SUFEEIW1CL/FESWY4QR5BiIG4nFEpF
XWG4HEPUkDJtCYgXtkCgifSzbXbjcxfP1UQHpkiKmmjIDT3h4PTFkMKPllqVhBpQ
Y53+VcGtNUze+AZZ6M2Iym1PSUyocAJBStRJOESPgHI+hzk5R8NRtgfPfbcvQ68Y
/oJKFfV/3S6mYt8HQAMXMCAA30QdvOKpKyydpeXcwrGzMvBauzqows/gOFOp96lC
++GZBZCUp4M6WkQcwB5/79RIgxAa86uJykAVfJJ2CIg=
=J+li
-----END PGP PUBLIC KEY BLOCK-----
```

➢ Step 3 – Configure Git for Signed Commits

1. **Tell Git which key to use**

2. **Enable signing for all commits**

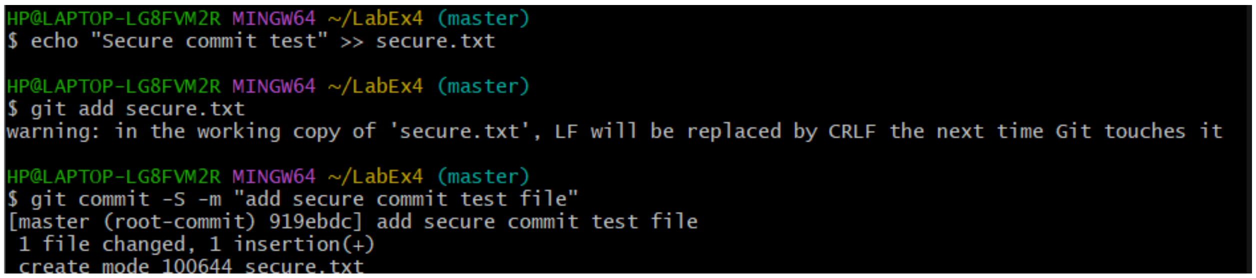


```
HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ git config --global user.signingkey 99670FA3B3E6E24D

HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ git config --global commit.gpgsign true
```

➢ Step 4 – Make a Signed Commit

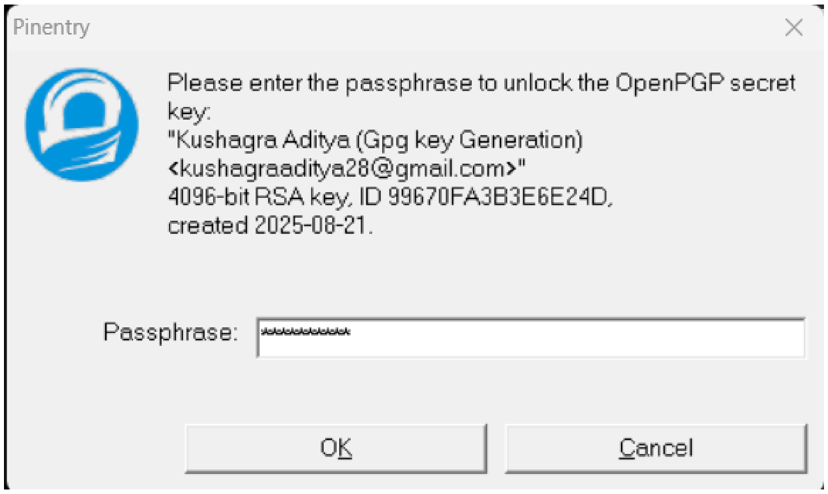
1. **Clone your repo (or use an existing one)**
2. **Edit or create a file**
3. **Commit with signing**
4. **Enter your GPG passphrase when prompted.**



```
HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ echo "Secure commit test" >> secure.txt

HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ git add secure.txt
warning: in the working copy of 'secure.txt', LF will be replaced by CRLF the next time Git touches it

HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ git commit -S -m "add secure commit test file"
[master (root-commit) 919ebdc] add secure commit test file
 1 file changed, 1 insertion(+)
 create mode 100644 secure.txt
```



Pinentry

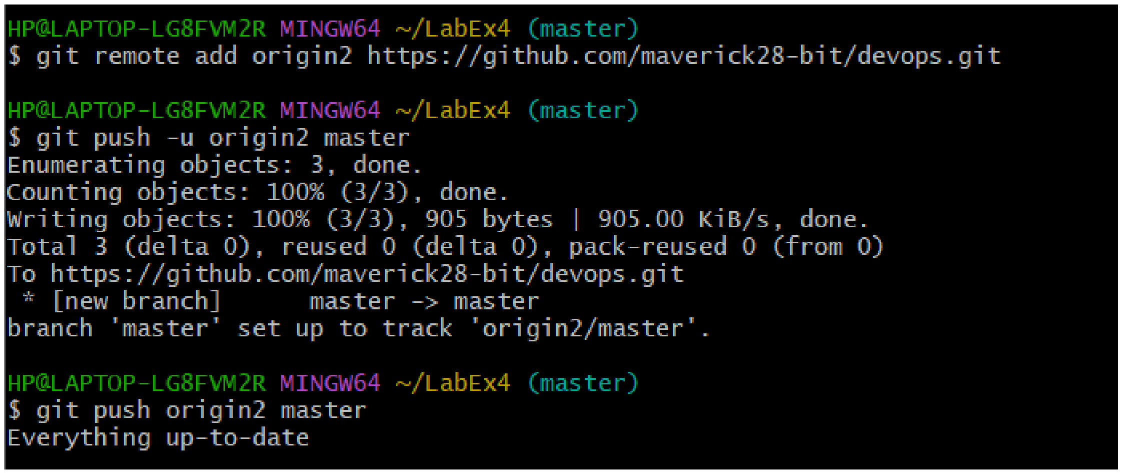
Please enter the passphrase to unlock the OpenPGP secret key:
"Kushagra Aditya (Gpg key Generation) <kushagraaditya28@gmail.com>"
4096-bit RSA key, ID 99670FA3B3E6E24D,
created 2025-08-21.

Passphrase: **********

➢ Step 5 – Push and verify on GitHub

1. **Push the commit:**

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



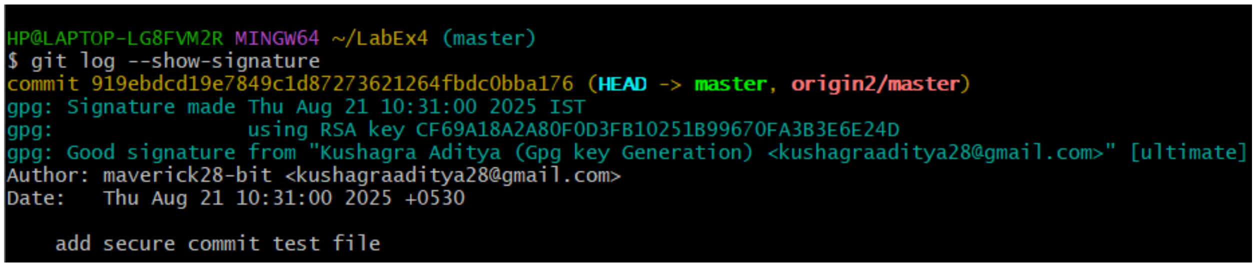
```
HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ git remote add origin2 https://github.com/maverick28-bit/devops.git

HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ git push -u origin2 master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 905 bytes | 905.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/maverick28-bit/devops.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin2/master'.

HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ git push origin2 master
Everything up-to-date
```

➢ Step 6 – Local Verification of Commit

This will display the GPG verification details locally.



```
HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ git log --show-signature
commit 919ebdcd19e7849c1d87273621264fbdc0bba176 (HEAD -> master, origin2/master)
gpg: Signature made Thu Aug 21 10:31:00 2025 IST
gpg:                using RSA key CF69A18A2A80F0D3FB10251B99670FA3B3E6E24D
gpg: Good signature from "Kushagra Aditya (Gpg key Generation) <kushagraaditya28@gmail.com>" [ultimate]
Author: maverick28-bit <kushagraaditya28@gmail.com>
Date:   Thu Aug 21 10:31:00 2025 +0530

    add secure commit test file
```

Use Case: -

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