Name – Namit Rampal

SAP ID – 500123236

Batch – B2

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
```

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

```
MINGW64:/c/Users/namit

namit@ThinkPadE15 MINGW64 ~ (main)
$  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/namit/.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: Namit Rampal
Email address: namitrampal53@gmail.com
Comment:
You selected this USER-ID:
    "Namit Rampal <namitrampal53@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/namit/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/c/Users/namit/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/c/Users/namit/.gnupg/openpgp-revocs.d/7A75D092AEED8E40A5A71C6B56A33C561BFA99D3.rev'
public and secret key created and signed.

pub   rsa4096 2025-08-20 [SC]
      7A75D092AEED8E40A5A71C6B56A33C561BFA99D3
uid                      Namit Rampal <namitrampal53@gmail.com>
sub   rsa4096 2025-08-20 [E]
```

3. **Get your key ID**

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

Output:

```
namit@ThinkPadE15 MINGW64 ~ (main)
$ 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   rsa4096/56A33C561BFA99D3 2025-08-20 [SC]
      7A75D092AEED8E40A5A71C6B56A33C561BFA99D3
uid                 [ultimate] Namit Rampal <namitrampal53@gmail.com>
ssb   rsa4096/B1D202DA68ECC355 2025-08-20 [E]


namit@ThinkPadE15 MINGW64 ~ (main)
$ |
```

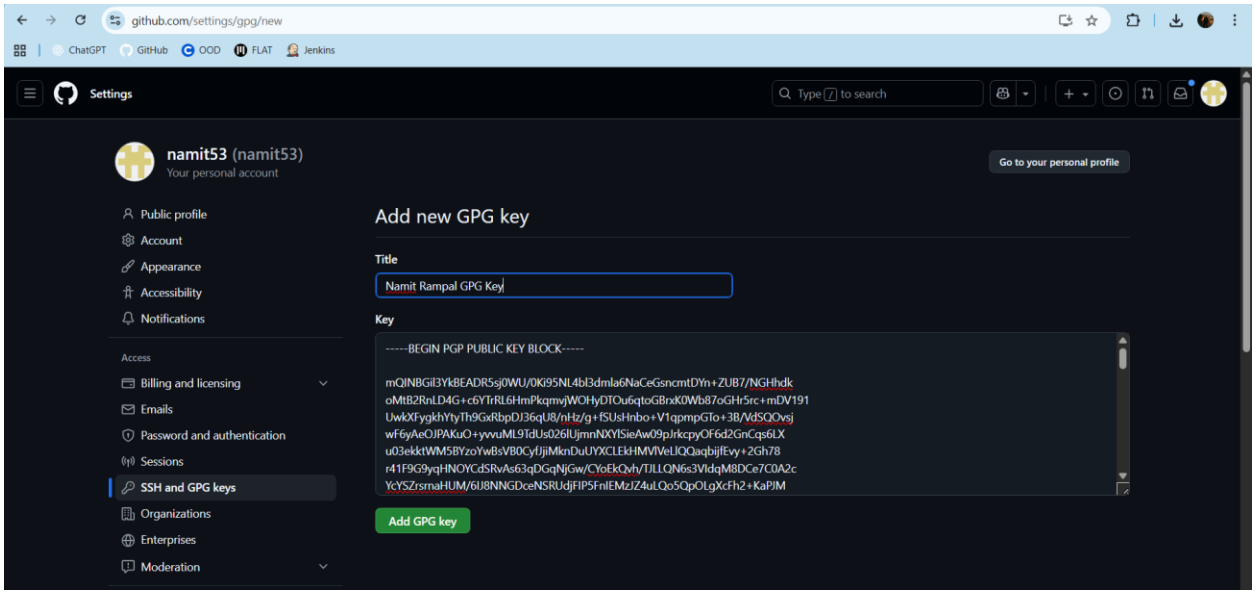
Step 2 – Add GPG Key to GitHub

1. Export your public key:

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

MINGW64:/c/Users/namit

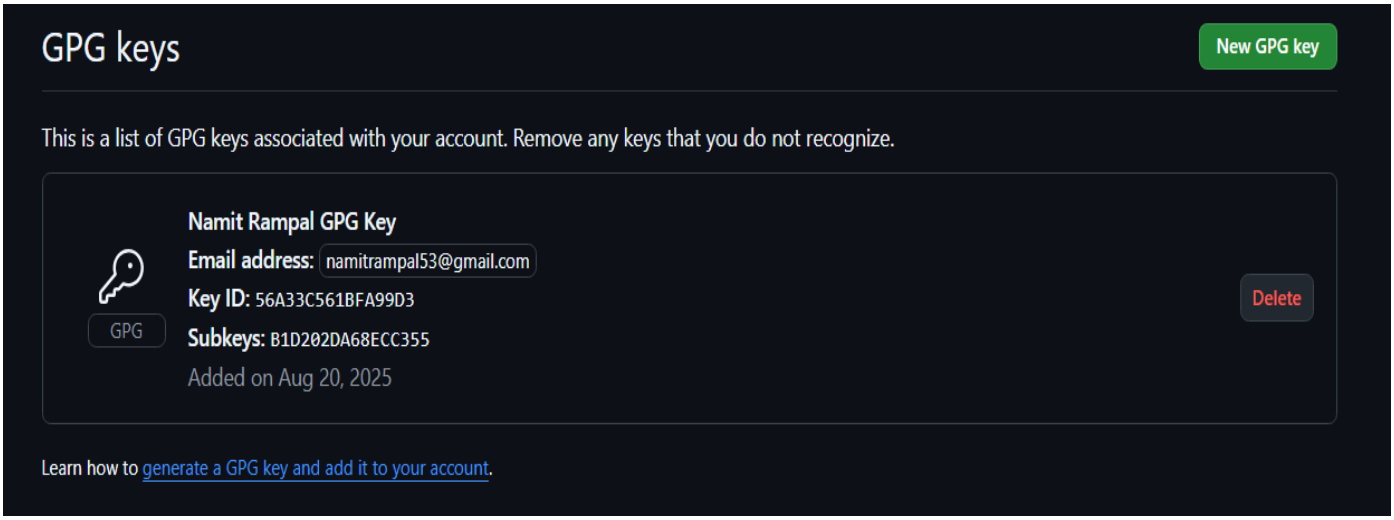
```
namit@ThinkPadE15 MINGW64 ~ (main)
$ gpg --armor --export 56A33C561BFA99D3
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGil3YkBEADR5sj0WU/0Ki95NL4bl3dmla6NaCeGsncmtDYn+ZUB7/NGHhdk
oMtB2RnLD4G+c6YTrRL6HmPkqmvjWOHyDTOu6qtoGBrxK0Wb87oGHr5rc+mDV191
UwkXFygkhYtyTh9GxRbpDJ36qU8/nHz/g+fSUsHnbo+V1qpmpGTo+3B/VdSQOvsj
wF6yAeOJPAKuO+yvvuML9TdUs026lUjmnNXYlSieAw09pJrkcpyOF6d2GnCqs6LX
uO3ekktWM5BYzoYwBsVBOCyfJjiMknDuUYXCLEkHMVlVeLlQQaqbijfEvy+2Gh78
r41F9G9yqHNOYCdSRvAs63qDGqNjGw/CYoEkQvh/TJLLQN6s3VIdqM8DCe7C0A2c
YcYSZrsrnaHUM/6IJ8NNGDceNSRUdjFIP5FnIEMzJZ4uLQo5QpOLgXcFh2+KaPJM
XNQdqZjbZ/SHtQOsiRYF2uOb185ILPNtFGgkicsbqTmJEa1kKvoSdaIsH83DOAWS
O4M64BlUfiWM6Gf8tOneKsYapJmNQHND1APAUYU7GvdGO6MKZkb/o2x3j7ZAisq5
SvMDIMtJOP8+jdp5feK9Fv7Bu38hSVkOO2DiY4/12aWt6Z/5DB6ByPk07dfl3gFr
FLFOlHhdgj470gg196Mtgy7O/JSMsv1Laz8x1OCNMPJ2xl/vNkV5CkUTpQARAQAB
tCZOYW1pdCBSYW1wYWwgPG5hbWlOcmFtcGFsNTNAZ21haWwuY29tPokCUQQTAQgA
OxYhBHp1OJKu7Y5Apacca1ajPFYb+pnTBQJopd2JAhsDBQsJCAcCAiICBhUKCQgL
AgQWAgMBAh4HAheAAAoJEFajPFYb+pnTQtoQAK5/yGyf6su+QNDiIXCK1Wi3WtjZ
kc9+en8hNR3BTZom+JCHAqr+GPz3t/9KvauwodZOjiK0s4gJsuMz1dV9KaRNQnAN
huBiHwxYKs+6k2vQxeoeeoCOVCqVPvb0k3JFipZQjtpmWAAKJAYwAxI+HqZr9rZT
L4ExtvwjHxIfQPfprYJRBOZS1BY96GRxHswl7tH1lmVhIE2SxEsPU4ljMkavWrVG
+ViZcRpLs7qROYHHM64BrRWgORBa3HBLKpo8CR8yIWDS828oMjpSCKbfWX18cbJq
OVagFENn4DZZ6W2035rVOcvk+eDg25oJC3OWJkgdlRUROYmerBH4JzZg1rR2ZsOX
EpQBG1pp7ta7kSgqOG4esdcMuu1e4OZn1RHkIlnC4+xcW+Bic4EquZKuXfC8JEiR
G1dV++CPh+ee0coH13kl4HQoY2V/c5emw+2bDwg8f3VZOdE0FY16K2GFeSCKShBD
sXBROhjZpeK+272RYAruyAGTjFGeg33N7t5NvfQs9dGBCrcfCwabrcdl9LHDoroI
fIXig3v67fLyUBxRJcucwRbp01OPlFbUt5WceEfBkRg6Tx8a5piNHFUDr9IPbtuA
baoz3OuEfUT75RN/Lu/IrlfgxP16H7wB1MFmOlOstgWdQxlQZFFXZMfBFPcth8aW
/Q1BXSO2HwgKVZUxuQINBGil3YkBEADUexIGW5Of7BrI4sK/+fdzOIPmdrPyTZiB
EyO3xGDs5QEN0ErzCHGz0+TMskj299O4BbANUOZ47+zqIkZS8OXuj91cSoLPkztI
qy8liDOIeNDffxwrtVrLoIrGXTXYvYo7MQ6nHAWIQ1KJaXz0X47PyYEP4xRjpaV6
JSzrq7hXLFnXi0eMs7yirtv4CvOySQ/0fOA0LLsk4cs3nEG1NWxbvMWYFZMYRQZS
YL4vwkivLBxjdwSc89Oird8m1TrY8AAfHPLKa1j1pfKtHtRmbPG32WXyZu/5eR2l
1ryXuxNSKcvQgA+s2U1I446elAtkcQlKPODniMuNLUQueF3fhnN/SrkYb6NLcPnB
kT3/OTdzN9V5ieFIh/vm0gi7ag997Z+OM1UupKI6X4+BJByVbIKDO+51EHrzDE3T
o3uAH/MfnijmPbKtQmHkaFuqdmqpm90S8yxRJ2Oirwbmcpm4/J6uLC/75cnBzHdb
WYJgyILXKE9tpLebZjEIFYthuaSaAwisImm1hFkwS+zCTTOzZ9Aqa8xgeqFFOQZK
rHZ2U8C7O3k2NWDkgupVXTPNSBSOQOVNz4PRYBftL4RInql5F+Bz44Pp1FHAqKpU
mM4hJeeQOeqe48MpkKICQPNfFQl4Rpz45zep3ZYvUQ9IfE6WfSQ2UbjldrtEVz98
bdwOJaPEmQARAQABiQI2BBgBCAAgFiEEenXQkq7tjkClpxxrVqM8Vhv6mdMFAmil
3YkCGwwACgkQVqM8Vhv6mdPdHQ/+MZRaT3quRSJYnliTfN+UMGCLDOOTeR4mWwlY
IZuzgOiaRtiH/KKUbKRSHdjvKW8PmJ8KqnSidDNWAhS3434I5RUBl5IM/WMQ6lfj
h/FXRWIUVcie7RtsYxjhMQirCDWsYLPkkzpWWef4V2urrJhfPBByJ+Iy4VEOYAtq
LHBvAoQ2+4/HtXA4/W5qZOEyX4IQsMnsAPU7/RJAK6mvKtNIsstHoFSDgpTd24J4
1jOuxSf3Nq/vjRKOOkUnZg46fW3Q+3XU4U9I+KNOjKDBGPwTR6eBdE37k81lrDrF
wgXm/3nRiGk1cZQhvZtFbXETUGeIxutEKUNOFW0q3+jixWwsHunQygQnYpzLEe+M
+aWY9SOgvUV8yWammxWJsma5qC1qtH86Ermm9ugtKy/W/gJlsKfiUfu7CnGdwPuh
nMGGhL/9Tf6MFNr/Dzy2pEJbplb9Eg1nO3lw2io47se1efwaF9o66/DA6WJ6oQZq
+1h4VdzHQu6GFMyQWXnaRiVxPIXC8lPQUnDep4anMZ6Xu9FvJ+BLrDzIlOCs7EPB
PU6WJWxpgawgE3qxd/Kgu1/S4uvismo3Ak6mOeONylyNYPJKMbqn5u/giyyUpmnN
IVCp69Gn/58Q2yCEoZX87JhUbmze04LiCs5vJLUsAnET9iqnoDKqvgcrHT7Ee3RO
zvX5RBo=
=whHc
-----END PGP PUBLIC KEY BLOCK-----

namit@ThinkPadE15 MINGW64 ~ (main)
$ |
```



2. Copy the output.

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

4. Paste your key and save.



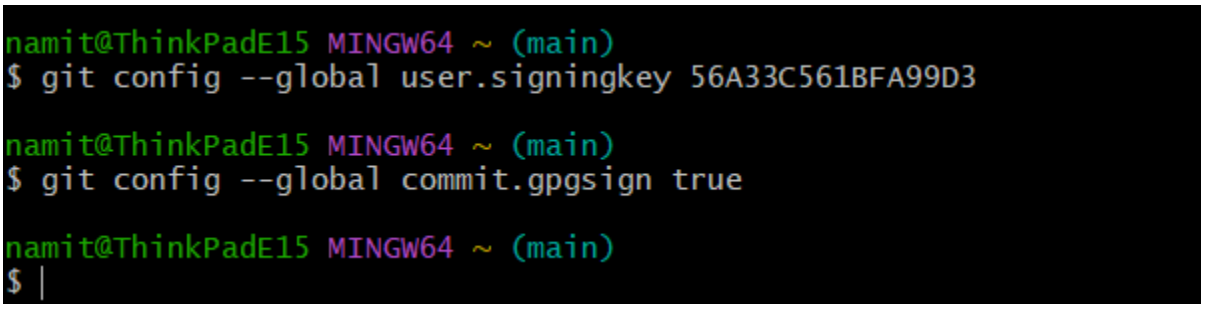
Step 3 – Configure Git for Signed Commits

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
```



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>
```

2. Edit or create a file:

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

git add secure.txt
```

3. Commit with signing:

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

4. Enter your GPG passphrase when prompted.

```
namit@ThinkPadE15 MINGW64 ~ (main)
$ git clone https://github.com/namit53/exp4_DevSecOps
Cloning into 'exp4_DevSecOps'...
warning: You appear to have cloned an empty repository.

namit@ThinkPadE15 MINGW64 ~ (main)
$ ls
 AppData/                      Links/
'Application Data'@            'Local Settings'@
 BrawlhallaReplays/            Music/
'Cisco Packet Tracer 8.2.2'/   'My Documents'@
 Contacts/                     NTUSER.DAT
 Cookies@                      NTUSER.DAT{685ad9db-2747-11f0-afc7-dcba70cb53ef}.TM.blf
 Documents/                    NTUSER.DAT{685ad9db-2747-11f0-afc7-dcba70cb53ef}.TMContainer00000000000000000001.regtrans-ms
 Downloads/                    NTUSER.DAT{685ad9db-2747-11f0-afc7-dcba70cb53ef}.TMContainer00000000000000000002.regtrans-ms
 Exp_6/                        'Namit Rampal_500123236_AIML Kmenas Assignment.ipynb'
 Favorites/                    NetHood@

namit@ThinkPadE15 MINGW64 ~ (main)
$ cd  exp4_DevSecOps

namit@ThinkPadE15 MINGW64 ~/exp4_DevSecOps (main)
$ 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

namit@ThinkPadE15 MINGW64 ~/exp4_DevSecOps (main)
$ git commit -S -m "Add secure commit test file"
[main (root-commit) 90ec18e] Add secure commit test file
 1 file changed, 1 insertion(+)
 create mode 100644 secure.txt

namit@ThinkPadE15 MINGW64 ~/exp4_DevSecOps (main)
$ |
```

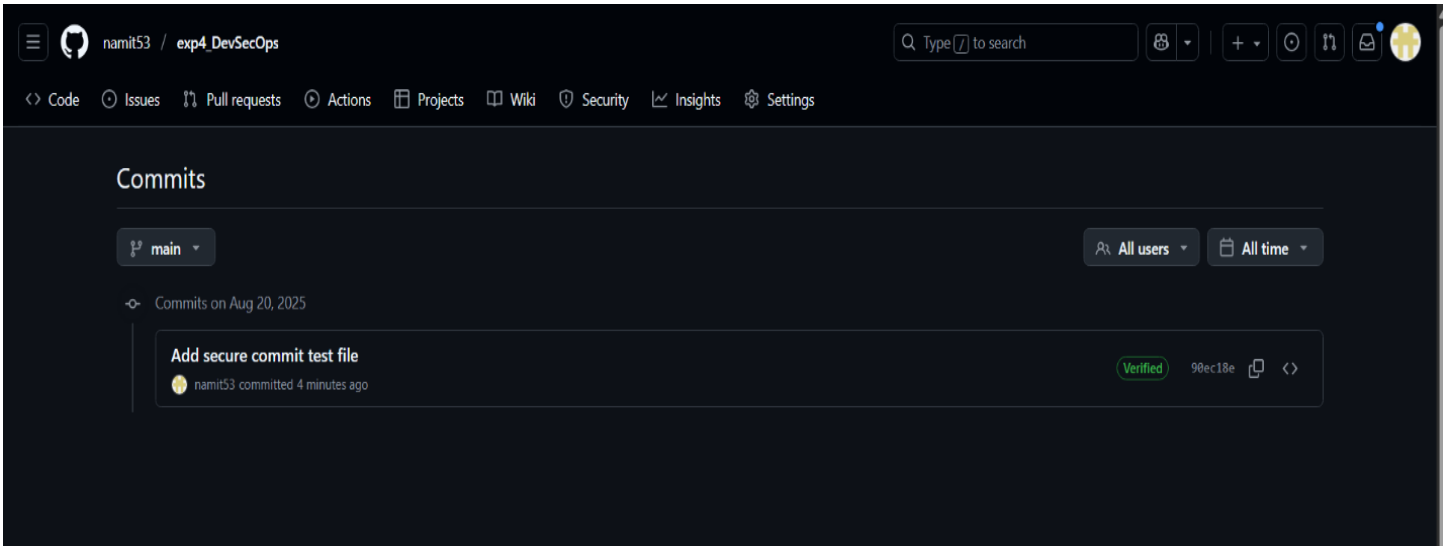
Step 5 – Push and Verify on GitHub

1. Push the commit:

```
git push origin main
```

```
namit@ThinkPadE15 MINGW64 ~/exp4_DevSecOps (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 898 bytes | 449.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/namit53/exp4_DevSecOps
 * [new branch]      main -> main

namit@ThinkPadE15 MINGW64 ~/exp4_DevSecOps (main)
$ |
```

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



Step 6 – Local Verification of Commit

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

This will display the GPG verification details locally.

```
namit@ThinkPadE15 MINGW64 ~/exp4_DevSecOps (main)
$ git log --show-signature
commit 90ec18e088b8abbdcf5545ab2c43c4ddb236fab8 (HEAD -> main, origin/main)
gpg: Signature made Wed Aug 20 21:02:11 2025 IST
gpg:                using RSA key 7A75D092AEED8E40A5A71C6B56A33C561BFA99D3
gpg: Good signature from "Namit Rampal <namitrampal53@gmail.com>" [ultimate]
Author: namit53 <namitrampal53@gmail.com>
Date:   Wed Aug 20 21:02:11 2025 +0530

    Add secure commit test file

namit@ThinkPadE15 MINGW64 ~/exp4_DevSecOps (main)
$
```

Use Case

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