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**

3. **Get your key ID**

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

Example output:

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


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

Step 2 – Add GPG Key to GitHub

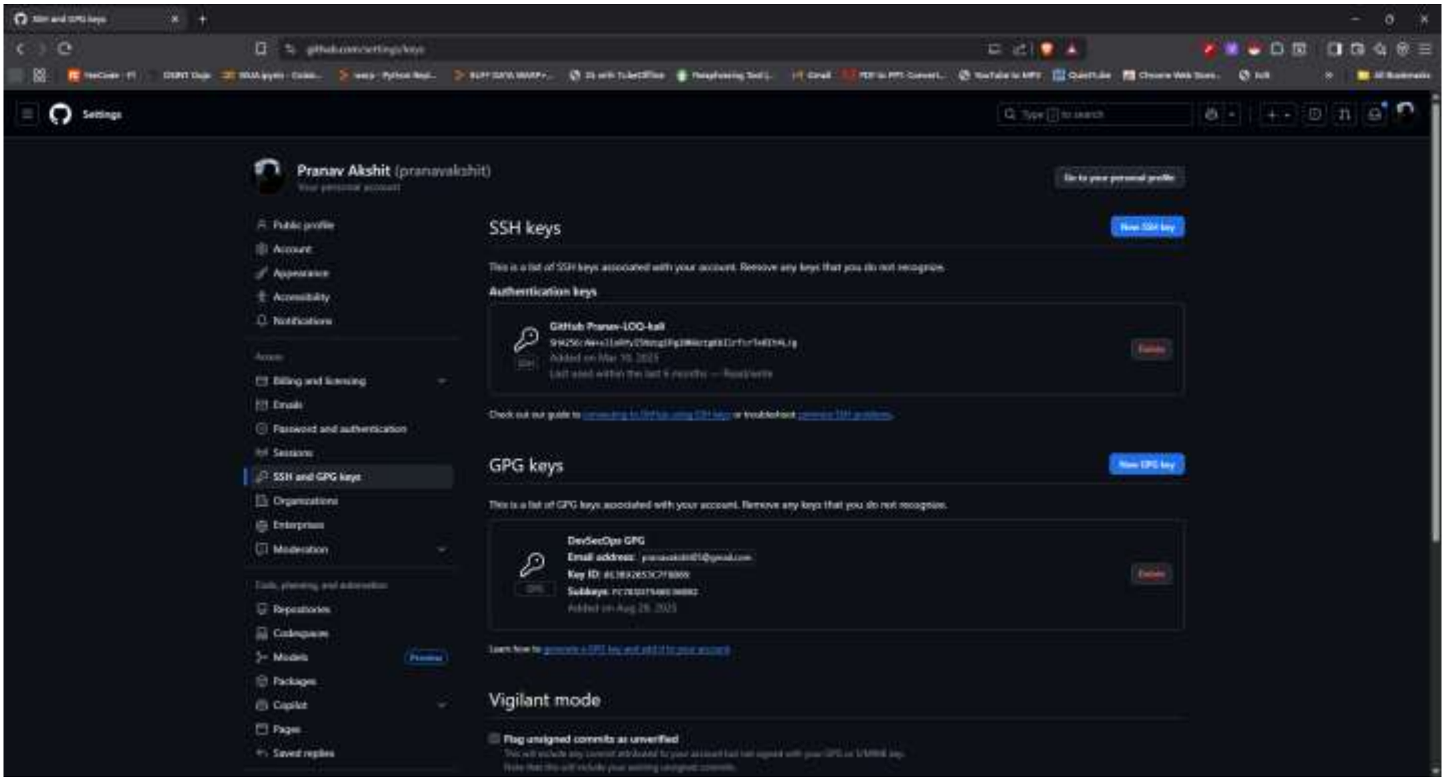
1. Export your public key:

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

2. Copy the output.

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

4. Paste your key and save.



Pranav Akshit (pranavakshit)

Your personal account

SSH keys

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

Authentication keys

Github Pranav-LOQ-kali

SHA256:[illegible]

Added on Mar 10, 2025

Last used within the last 6 months — Read/write

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.

DevSecOps GPG

Email address: [illegible]@gmail.com

Key ID: [illegible]

Subkeys: [illegible]

Added on Aug 28, 2025

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

Vigilant mode

□ Flag unsigned commits as unverified

This will include any commit attributed to your account but not signed with your GPG or S/MIME key.

Note that this will include your existing unsigned commits.

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.

Step 5 – Push and Verify on GitHub

1. Push the commit:

```
git push origin main
```

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

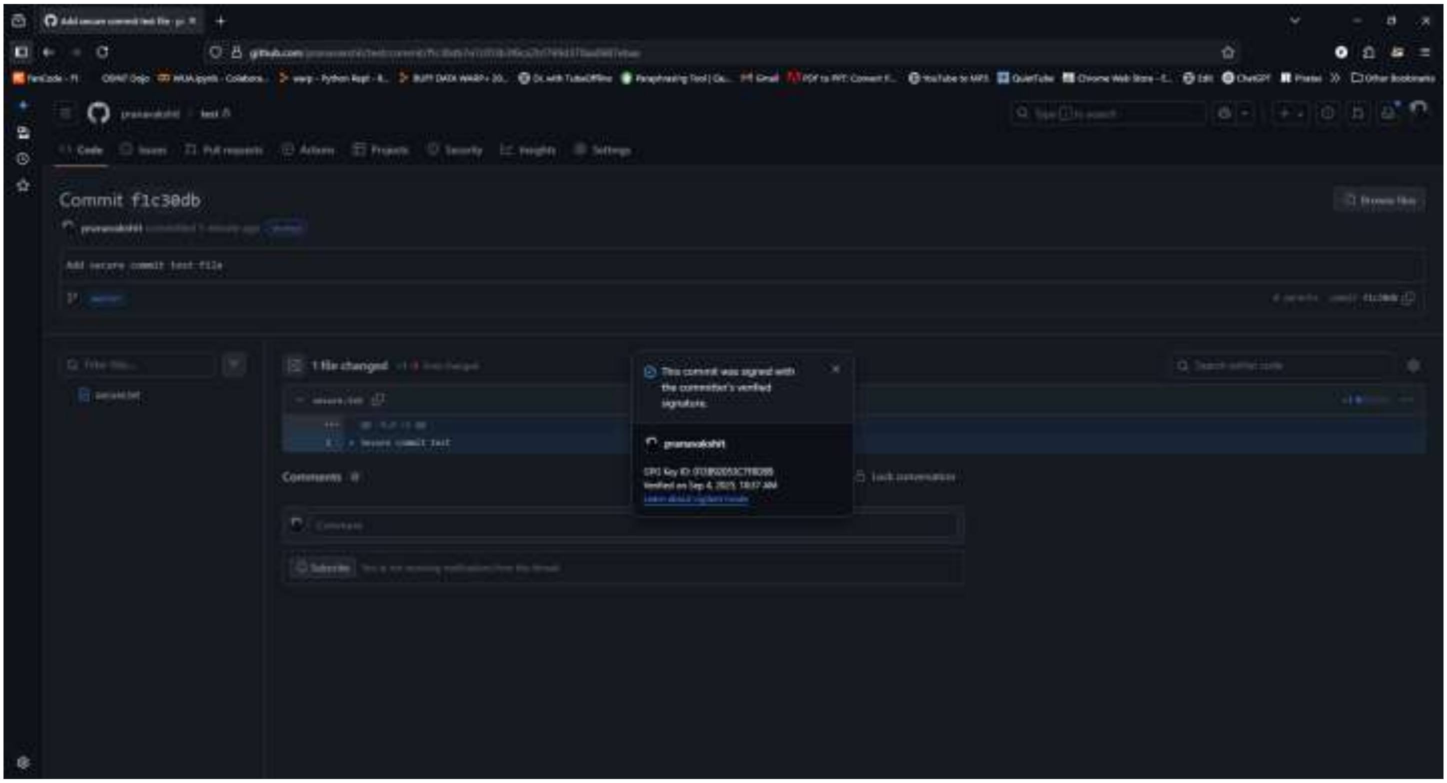
Step 6 – Local Verification of Commit

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

This will display the GPG verification details locally.

Use Case

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



pranavakshit / test

Commit f1c30db

pranavakshit

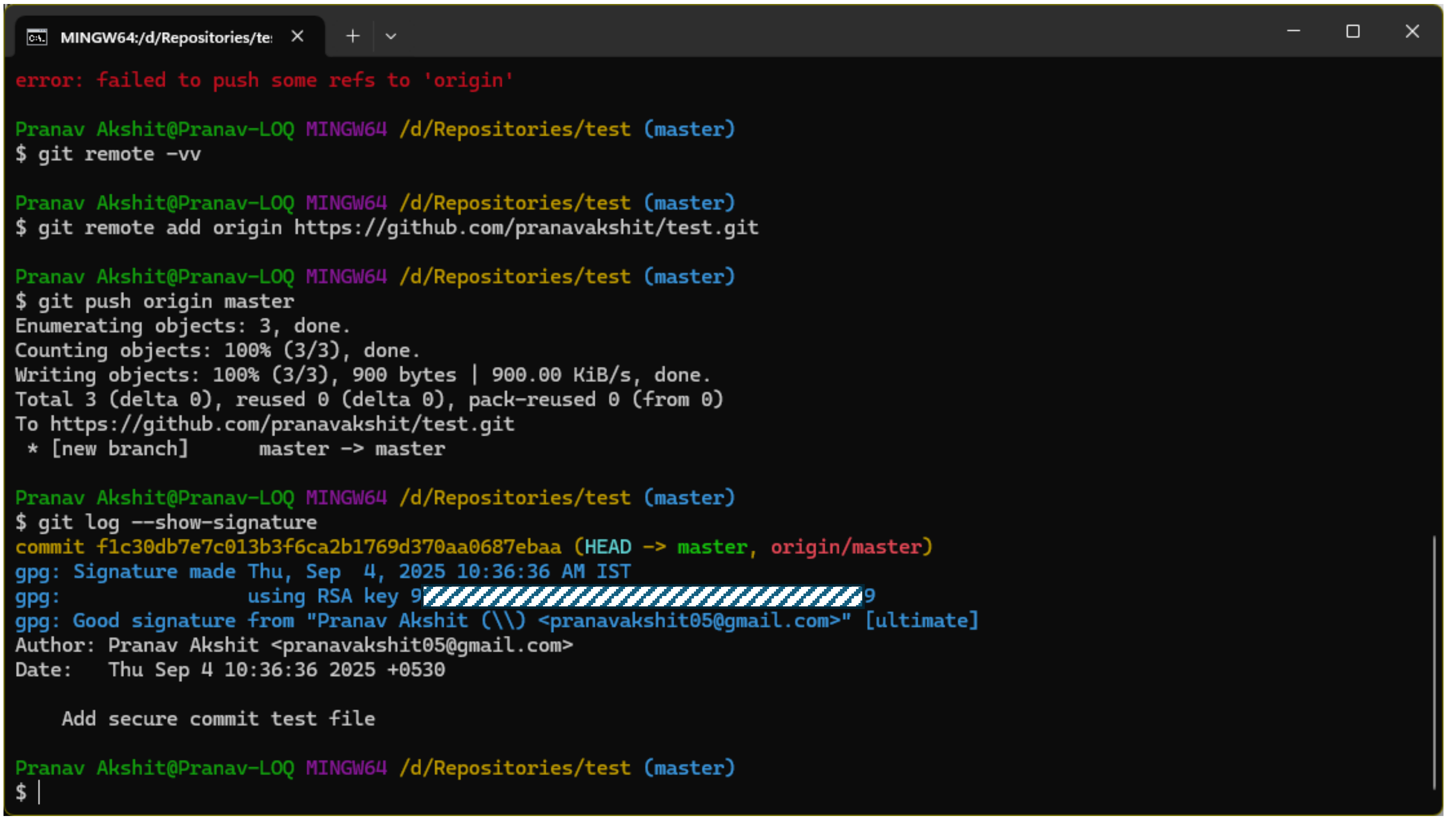
Add secure commit test file

1 file changed

This commit was signed with the committer's verified signature.

pranavakshit

GPG Key ID: [illegible]
Verified on Sep 4, 2025, 10:37 AM



```
error: failed to push some refs to 'origin'

Pranav Akshit@Pranav-LOQ MINGW64 /d/Repositories/test (master)
$ git remote -vv

Pranav Akshit@Pranav-LOQ MINGW64 /d/Repositories/test (master)
$ git remote add origin https://github.com/pranavakshit/test.git

Pranav Akshit@Pranav-LOQ MINGW64 /d/Repositories/test (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 900 bytes | 900.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/pranavakshit/test.git
 * [new branch]      master -> master

Pranav Akshit@Pranav-LOQ MINGW64 /d/Repositories/test (master)
$ git log --show-signature
commit f1c30db7e7c013b3f6ca2b1769d370aa0687ebaa (HEAD -> master, origin/master)
gpg: Signature made Thu, Sep  4, 2025 10:36:36 AM IST
gpg:                using RSA key 9[redacted]9
gpg: Good signature from "Pranav Akshit (\\) <pranavakshit05@gmail.com>" [ultimate]
Author: Pranav Akshit <pranavakshit05@gmail.com>
Date:   Thu Sep 4 10:36:36 2025 +0530

    Add secure commit test file

Pranav Akshit@Pranav-LOQ MINGW64 /d/Repositories/test (master)
$ 
```