Name : Gourav das

Sap id – 500122586

DevOps

Batch no - 2

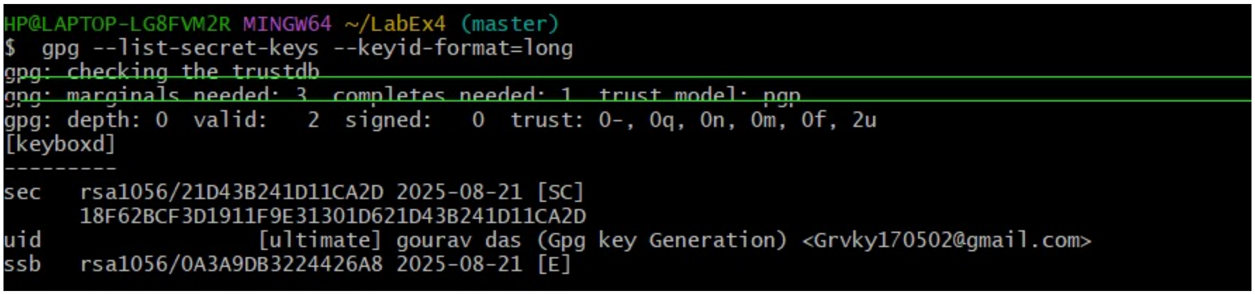
Lab Exercise 4- Signed Commits in Git and GitHub

Objective:

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

Prerequisites:

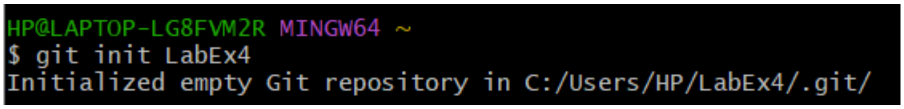
- Git installed on your system
- GPG (GNU Privacy Guard) installed and configured
- GitHub account with a repository (you own or have write access to)
- Basic knowledge of Git commands



```
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:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
[keyboxd]
---------
sec   rsa1056/21D43B241D11CA2D 2025-08-21 [SC]
      18F62BCF3D1911F9E31301D621D43B241D11CA2D
uid                 [ultimate] gourav das (Gpg key Generation) <Grvky170502@gmail.com>
ssb   rsa1056/0A3A9DB3224426A8 2025-08-21 [E]
```

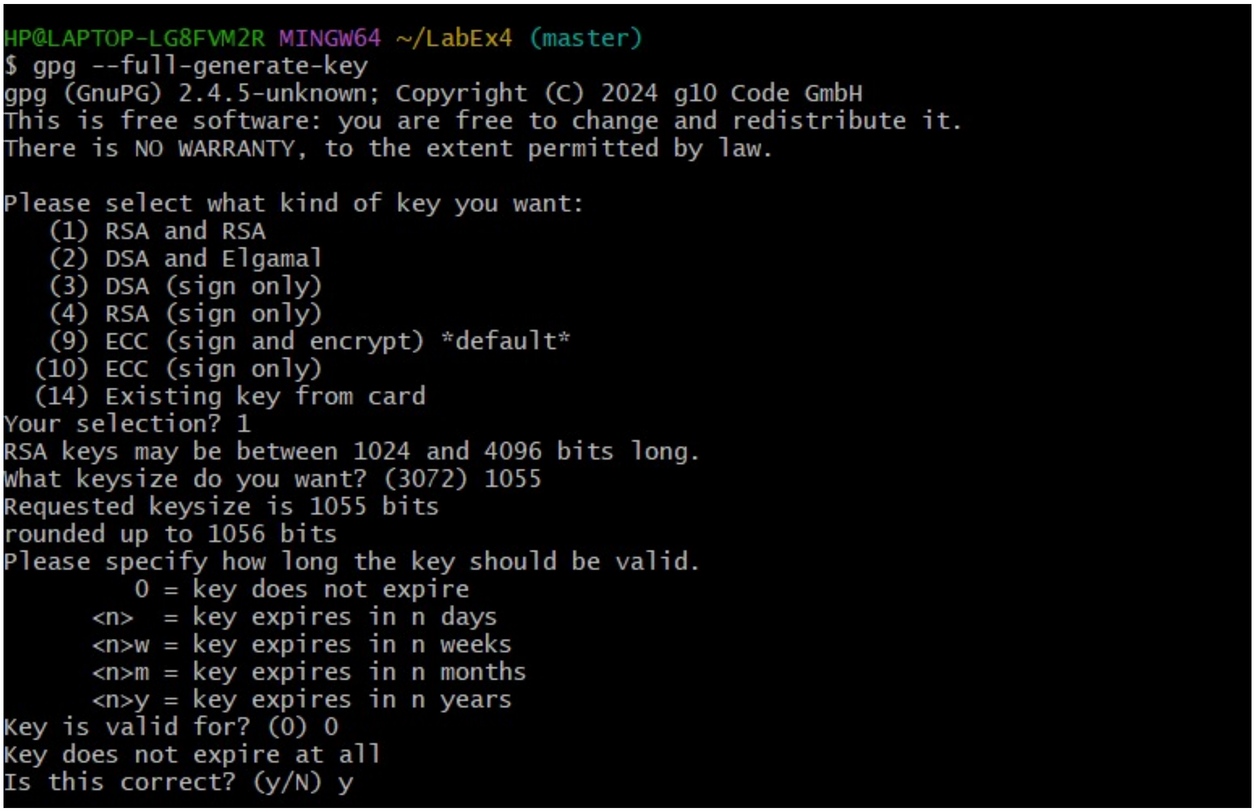
➢ Step 1 – Generate or Use an Existing GPG Key

1. Check for existing keys



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

2. If no key exists, generate a new one

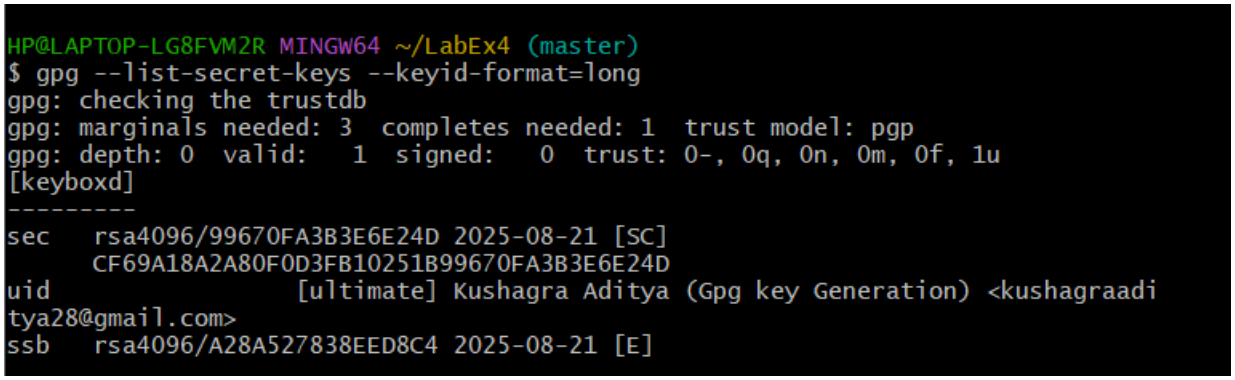


```
HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ gpg --full-generate-key
gpg (GnuPG) 2.4.5-unknown; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

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) 1055
Requested keysize is 1055 bits
rounded up to 1056 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
```

- Select **RSA and RSA**
- Key size: **4096**

- Expiration: 0 (never) or a fixed date
- Enter your GitHub-registered name and email

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]
```

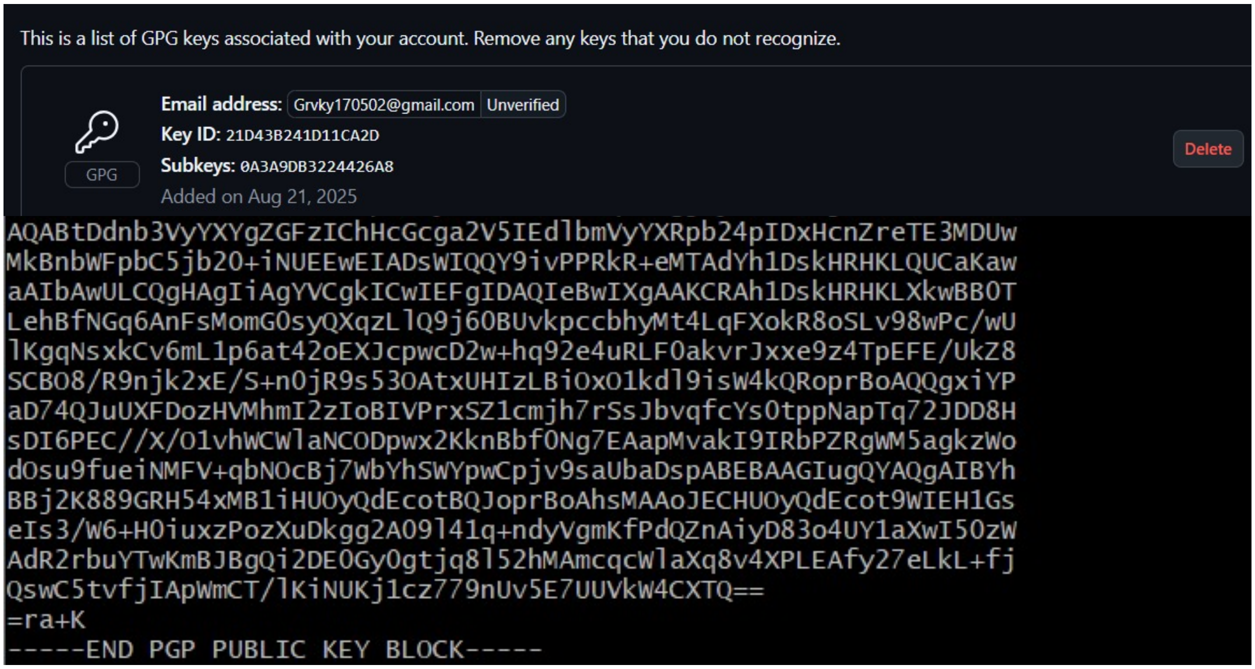
➢ Step 2 – Add GPG Key to GitHub

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.



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

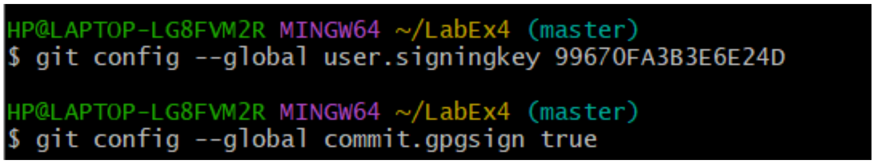
Email address: Grvky170502@gmail.com Unverified
Key ID: 21D43B241D11CA2D
Subkeys: 0A3A9DB3224426A8
Added on Aug 21, 2025

```
AQABtDdnb3VyYXYgZGFzIChHcGcga2V5IEdlbmVyYXRpb24pIDxHcnZreTE3MDUw
MkBnbWFpbC5jb20+iNUEEwEIADsWIQQY9ivPPRkR+eMTAdYh1DskHRHKLQUCaKaw
aAIbAwULCQgHAgIiAgYVCgkICwIEFgIDAQIeBwIXgAAKCRAh1DskHRHKLXkwBB0T
LehBfNGq6AnFsMomG0syQXqzLlQ9j60BUvkpccbhyMt4LqFXokR8oSLv98wPc/wU
lKgqNsxkCv6mL1p6at42oEXJcpwcD2w+hq92e4uRLF0akvrJxxe9z4TpEFE/UkZ8
SCBO8/R9njk2xE/S+n0jR9s53OAtxUHIzLBiOxO1kdl9isW4kQRoprBoAQQgxiYP
aD74QJuUXFDozHVMhmI2zIoBIVPrxSZ1cmjh7rSsJbvqfcYs0tppNapTq72JDD8H
sDI6PEC//X/O1vhWCWlaNCODpwx2KknBbf0Ng7EAapMvakI9IRbPZRgWM5agkzWo
dOsu9fueiNMFV+qbNOcBj7WbYhSWYpwCpjv9saUbaDspABEBAAGIugQYAQgAIBYh
BBj2K889GRH54xMB1iHUOyQdEcotBQJoprBoAhsMAAoJECHUOyQdEcot9WIEH1Gs
eIs3/W6+H0iuxzPozXuDkgg2A09l41q+ndyVgmKfPdQZnAiyD83o4UY1aXwI50zW
AdR2rbuYTwKmBJBgQi2DE0Gy0gtjq8l52hMAmcqcWlaXq8v4XPLEAfy27eLkL+fj
QswC5tvfjIApWmCT/lKiNUKj1cz779nUv5E7UUVkW4CXTQ==
=ra+K
-----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
```

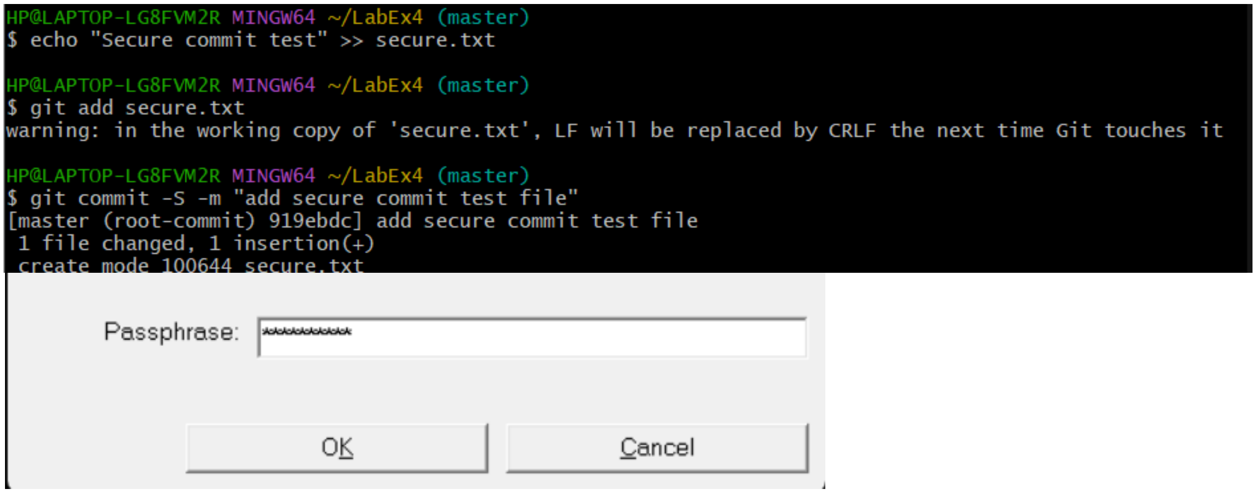
➢ Step 4 – Make a Signed Commit

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.



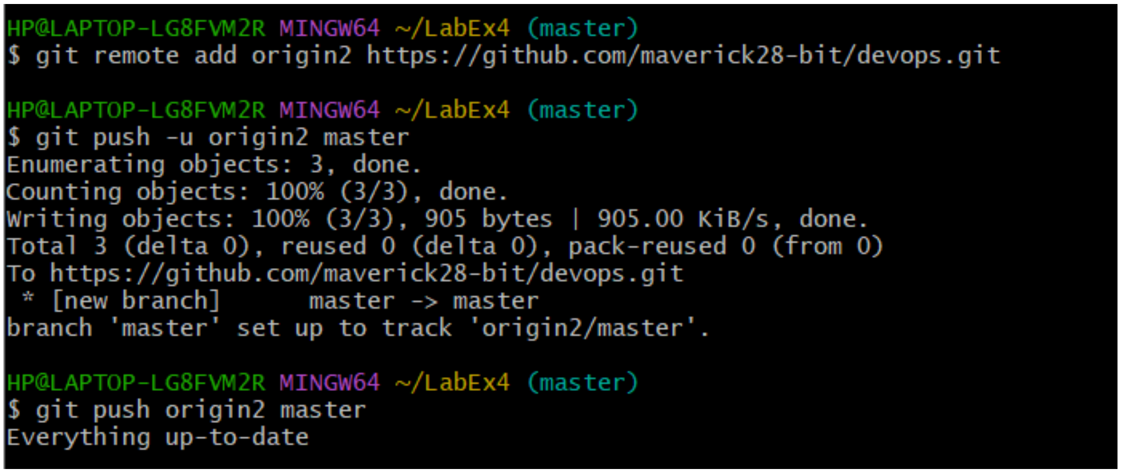
```
HP@LAPTOP-LG8FVM2R MINGW64 ~/LabEx4 (master)
$ echo "Secure commit test" >> secure.txt

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
```

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

➢ Step 5 – Push and verify on GitHub

1. **Push the commit:**

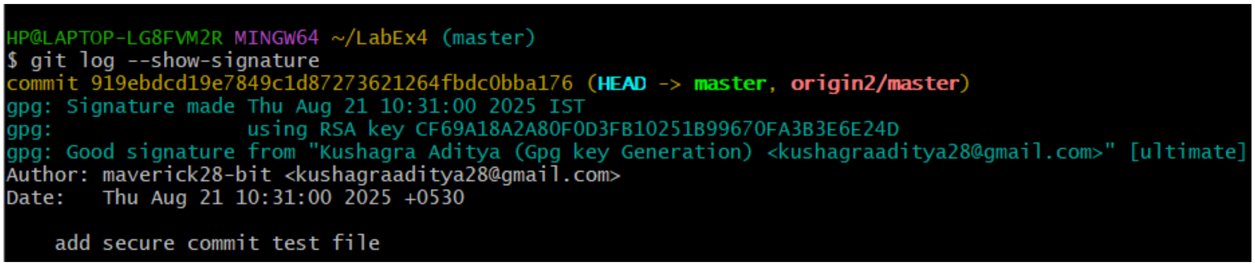
2. **Go to your repository on GitHub → Click the commit → You should see a <u>green "Verified" badge.</u>**



```
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.