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

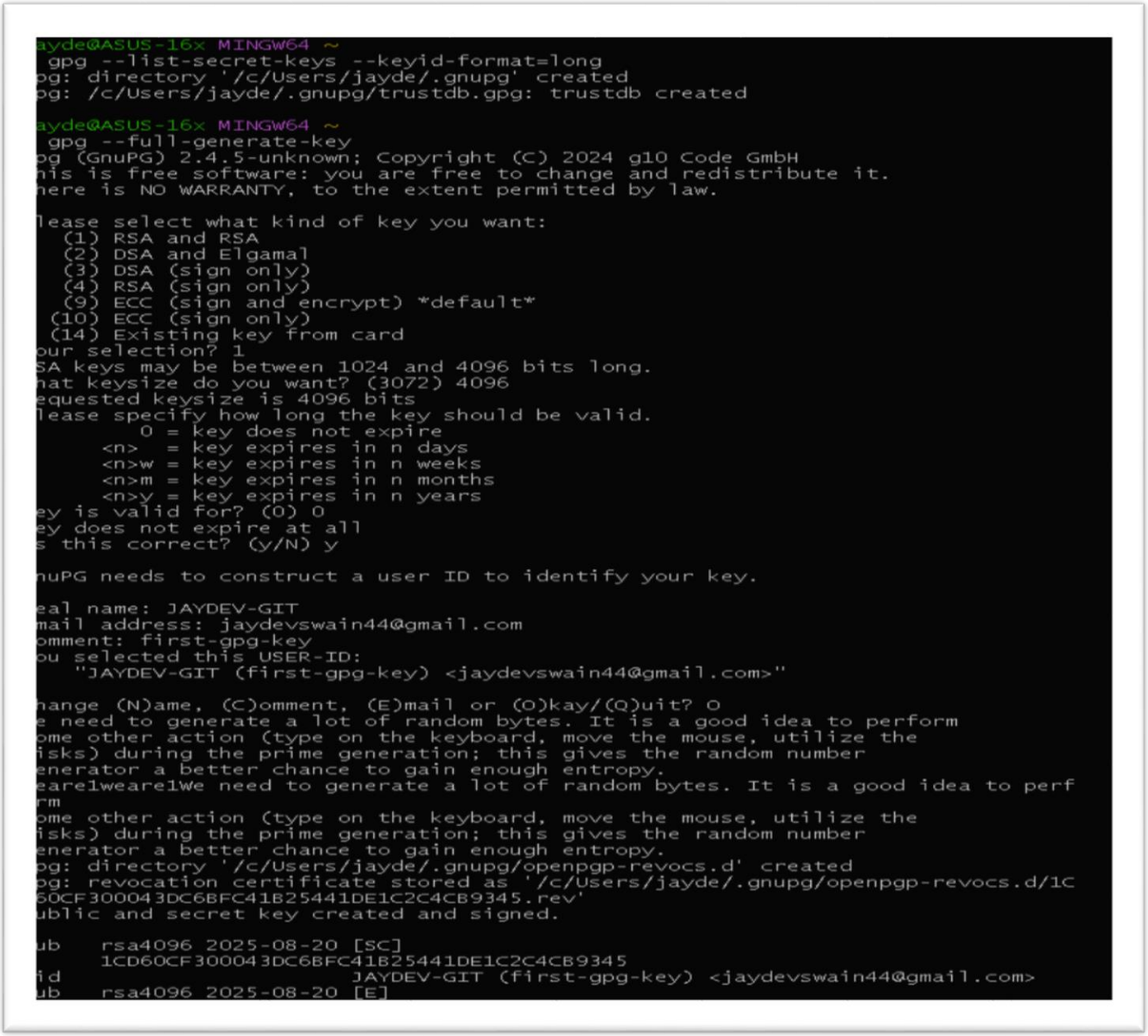
Step 1 – Generate or Use an Existing GPG Key

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
```

```
jayde@ASUS-16x MINGW64 ~
$ 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   rsa4096/441DE1C2C4CB9345 2025-08-20 [SC]
      1CD60CF300043DC6BFC41B25441DE1C2C4CB9345
uid                 [ultimate] JAYDEV-GIT (first-gpg-key) <jaydevswain44@gmail.com>
ssb   rsa4096/A0AFE2F6FA631423 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:

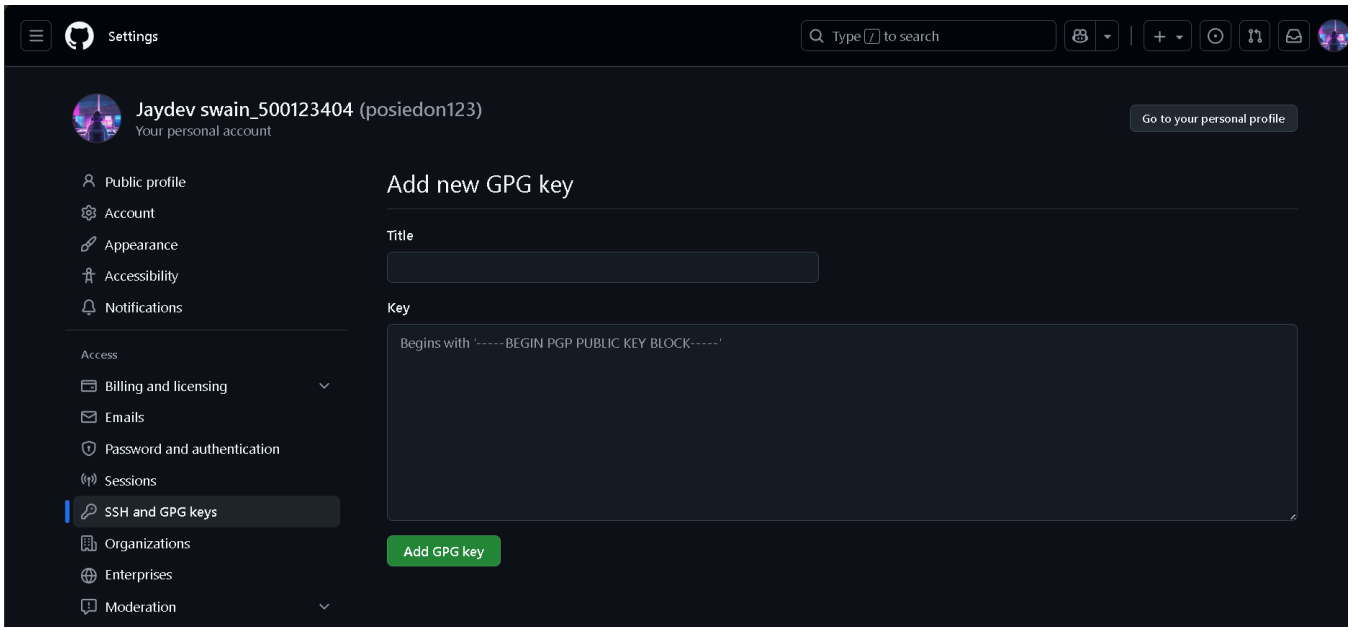
```
gpg --armor --export YOUR_KEY_ID
```

```
jayde@ASUS-16x MINGW64 ~
$ gpg --armor --export YOUR_KEY_ID
gpg: WARNING: nothing exported

jayde@ASUS-16x MINGW64 ~
$ gpg --armor --export 441DE1C2C4CB9345
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGimMHsBEADdNQODkKeZyT2oAi/Yz/yxQ22mO9yyKkbXfe77Tvy9h13nWHEH
lNjsOZ3Zof7IxUq6TaavQ2XGyUcghzDYn8XVtRIXK3j+6sEQEQoz6/khiDjUfUBg
dsf4zL3gtrkctxAbNiDul4NF6cb3eBVlz/KMQRznf+T3/o4qcCaNiftgIgjtPCfH
gM7MEW7TBJ9NKO8KK5Le3geWjcteelJ+WvqjjNf6W5MTsv65XT/vRayQK89zvU6h
vsFaeROP1rir4OLn12Vt3o/vFbqzGEoQ39Yr4isNaPzbdwGPe+qsd5vJPsa4agUN
eEYHwh7xxN2rOzQe6a9ADi7Mm5r37UbbyqL41RstPlYvFPOp3DnsJYnfaX4+3cIe
LSwPj6a53VC9OQpKBgviA857DSmxObEGs1AIt35mXztuQlPXBbsEajM2CHc+eHMw
Mxlm68ZWrzj5eq+eUSYJhgWUwifMoJ5v443OXgb4upENURwZYmOAmJLhu/wozLsX
TTiOOpL1iaiR+kc2I4a8udKIx7LUGn6W3oS1NsgrIIBCSE5hAGjXzz/sFH6OmMjg
A3HFpuYovXLVI/RL/oe/U2XxIqHNXJwpqYSMxQyC1UvQOmgfjghIaCO2k0WgpdNj
eFBPLjmZ2H9GfMJiVF33hiubVLluOWwOr7/Jyclp9bwRU9UhZSaGOapEzwARAQAB
tDRKQVlERVYtROlUIChmaXJzdC1ncGcta2V5KSA8amF5ZGV2c3dhaW4ONEBnbWFp
bC5jb2O+iQJRBBMBCAA7FiEEHNYM8wAEPca/xBslRB3hwsTLkOUFAmimMHsCGwMF
CwkIBwICIgIGFQoJCAsCBBYCAwECHgcCF4AACgkQRB3hwsTLkOWCtxAAnpJfYhYE
EO0+yBayfBjUXMavrnTjGggGkRdbzZrgrx3CEXv1Gn8QBEomePw9NrOypXwRkPqo
2bBIZdWSKiQfPOMjpgNlU2ibiVvvao8LxUoUiAtTGzb5zLjIZV8StO1ORoREkqlq
rUZqnKvARvNl+lJmua2ZqrCpDxlc+WnoPbN4KUd7bbgjbcimR8ZlF1bXsUj8wXOy
G1DlO9iA6cSRWMXM8kL/1wdxe+/ezkpdua+rDLwjkPNEymEk9fRB/zU4a+VjDBIP
qz8y3PW6qR+cKSy7cZa2Q6/qBDDOCSULAvCrbP3ayOxDcxbOGSjWmS32GWlR4xiY
4+M63EepiCgh9EfTGSKp+skZnAeOSZoaPO/31VS9EEXUm4TNAwq7LiH9fk/U8qpo
hyAktJEXCFYimAjQMhOsFO+3ybsrudcitvWXtBm8jYcVLK8gUZ87gAvgFwqf9CBE
UziVonn7RRWGv6UHK5uV+NCECxwYoe7PqKHhmY29WnRv1kGx6gymLXTiDoKNvt5H
ZoVJADn/8LihRbXKOv4eGiSkhVV6CGCEZuKPixUROig3vcu2wOcLG77KK7dBDZ41
NR3KyfCjTrbg2kvZIZqyMZ2I1bpnQIV7kcwBRYCj5pCGxorgNFhKCUxFrRI8k8q3
OLZVZNNwz1NM7Oe1UAjctpHaxf/hRFehD3e5AgOEaKYwewEQAKkMiYKXOibc/If6
LTn8l9b+tlGbsYaut/9yQtCkJOqa4Oy7hcYBVFKw2fvqNdQdJ1POiezv1/yeDrTG
XAloSqku5LrdTTm4j11VRdJc9GbUwF6iiLo3wKU7CowlOwofNySGLf1qPULHi+q+
ItNoi4N4pL9482VUc6KlTYV/X2o5DpLvrd8p3JI6YzwXEdjD+x7fiynkouXt2Kud
PLNmFtj4Z2VGIVdllkUCRQY5t/BUoTsGdNw2Z1KVs8Kjba+47ORJpAC+61ueEv2B
Ags+CdU3fhwrN5j7Tq4wh1gXdSXt3FRo6LwvO3xIXmJkgP+IZn+E+NyVCQn5kbCE
6tHkYbzcxqC65pSHNljw9ZTwsqVIpdqqpZTt6N+h92BSOLFuWB4NqE7P1mqeXyDI
c2Vwbr4MTV0axYFP36SLGIqVSbgrYrif1nQjnZcYv4NqdoCqvSy1Ov1UHyNWPOuU
wePa3/AODIxoL9u78cKzLmx2Yg39KbPXQL35HAfdwKmrigFHcOYq3LX4YAJlFCkM
aTd2lpKLOQu5IzVWmKBKvU0gYmzM+UOgHJIuLkSTIvNOM1Mlt8TmxKM3H1WtwDOQ
A/qaiH1G2azHSsAsuHZZE45pIhtj+9J0CknRk+loajcHr2mZnUGnUMqYgpjDMyYi
v1+rNoAXyO0kUfjkTpA5zGVtOVFFABEBAAGJAjYEGAEIACAWIQQc1gzzAAQ9xr/E
GyVEHeHCxMuTRQUCaKYwewIbDAAKCRBEHeHCxMuTRRx8D/wMBff5RxH8LpCA1+UT
ETogWebPzuGfDeIB4R7zdJbgxmA6xikJ8yceJjy97oXeqQNqgome12/1H5ByWOBN
mzIJWB7KLhj187u+KKNSe3Ibqx5O9ewvTAKG9d0T5jPgH7o1JLwOlde2ow1OtJhC
UFMIZN+XadLsisiZ1iLRxg7vu+MkEPeDFkuLpyu9sxC6DY+AW9wm3HuoJt2+p0St
x4Vw3pR/jHNlfT9SNouAQEkIQHiEBFFLNIsGlYYZfc7RFLJSPxdqTfpvXXHXGXyU
6aNdlJRYOwNnJDY+up3s7am7LuJcJCbQtEZ78zV1Wh1fFPRbcsgSdVOXigtr6mhS
DcaV7ZWvH81BvrX/NS/z8buJoNa8ihwgVSWBR+KHN4JlDxJVloAX6xrNROyYEvKk
qPDLeEGZYTt6EOFRpR/EPNv5JaiMUfAcgqdxfGNiX6OhYXL6rn+3LmA/S84XAxsi
xzWFt9I+1aj7LO1KW1fv5HSlPWfl/MCrDaLJHNu/US9LaEaii1i4aWjrvc9qhhBy
dw4GSV3Y1gzGCOlIvRQZY7nFTeWWrYBvI5D1CaqtOSP0qaXk7Eego1lHxyDJ2ogq
O7SCnY5zqhzPBhyx43ATLZWUXvQdT7Or/TBfix3r9/ZFTY3YJXVLab5n17pps4Oz
8PwLdG3bFYLrOY7iZvkTyWDwug==
=Kcy9
-----END PGP PUBLIC KEY BLOCK-----

jayde@ASUS-16x MINGW64 ~
$
```

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
```

```
jayde@ASUS-16x MINGW64 ~
$ git config --global user.signingkey  441DE1C2C4CB9345

jayde@ASUS-16x MINGW64 ~
$ git config --global commit.gpgsign true

jayde@ASUS-16x MINGW64 ~
$
```

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>
```

```
jayde@ASUS-16x MINGW64 ~
$ git clone https://github.com/posiedon123/GPG-KEY
Cloning into 'GPG-KEY'...
warning: You appear to have cloned an empty repository.

jayde@ASUS-16x MINGW64 ~
$
```

2. Edit or create a file:

```
jayde@ASUS-16x MINGW64 ~
$ cd GPG-KEY

jayde@ASUS-16x MINGW64 ~/GPG-KEY (main)
$ echo "Secure commit test" >> secure.txt

jayde@ASUS-16x MINGW64 ~/GPG-KEY (main)
$ git add secure.txt
warning: in the working copy of 'secure.txt', LF will be replaced by CRLF the next time Git touches it

jayde@ASUS-16x MINGW64 ~/GPG-KEY (main)
$ git commit -S -m "Add secure commit test file"
[main (root-commit) 73ab1e9] Add secure commit test file
 1 file changed, 1 insertion(+)
 create mode 100644 secure.txt

jayde@ASUS-16x MINGW64 ~/GPG-KEY (main)
$
```

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.