

PKCS#11 Wrapper – GPG2, Debian & RPM Signing Integration Document

GPG2, or GNU Privacy Guard 2, is a free, open-source implementation of the OpenPGP standard designed to provide powerful encryption and signing capabilities. Combined with Encryption Consulting's PKCS#11 Wrapper, the signing key is held in your HSM and reached through GPG's smartcard daemon, so the private key never leaves the HSM and every signing operation is recorded centrally.

Debian and RPM package signing both delegate to GPG2 for the actual signature, so all three signing types share the same foundation. Sections 1 to 10 build that foundation on an Ubuntu machine. After that, go to Section 11 for GPG2 signing, Section 12 for Debian signing, or Section 13 for RPM signing — each of the last two adds only the extra steps specific to that package format.

NOTE: To perform Debian and RPM signings, you will still need to configure and set up GPG on your machine, which means completing Sections 1 to 10 first.

Prerequisites: An Ubuntu machine with root access, access to the CodeSign Secure portal, and a sample file, .deb, or .rpm package to sign.

1. Prerequisites and Root Access

The whole of this workflow is performed as the root user, because the GnuPG home directory used throughout is /root/.gnupg.

Steps:

Step 1: Log In as the Root User

- First, you need to be logged in as the root user before we move forward with GPG2 signing.

```
sudo su
```

```
aryan@Signing:~$ sudo su
[sudo] password for aryan:
root@Signing:/home/aryan#
```

Step 2: Install the Wrapper Dependencies

- To set up the PKCS#11 Wrapper, you will need to run the following commands and install some dependencies and packages on your Ubuntu machine:

```
sudo apt-get install curl  
sudo apt-get install liblog4cxx12
```

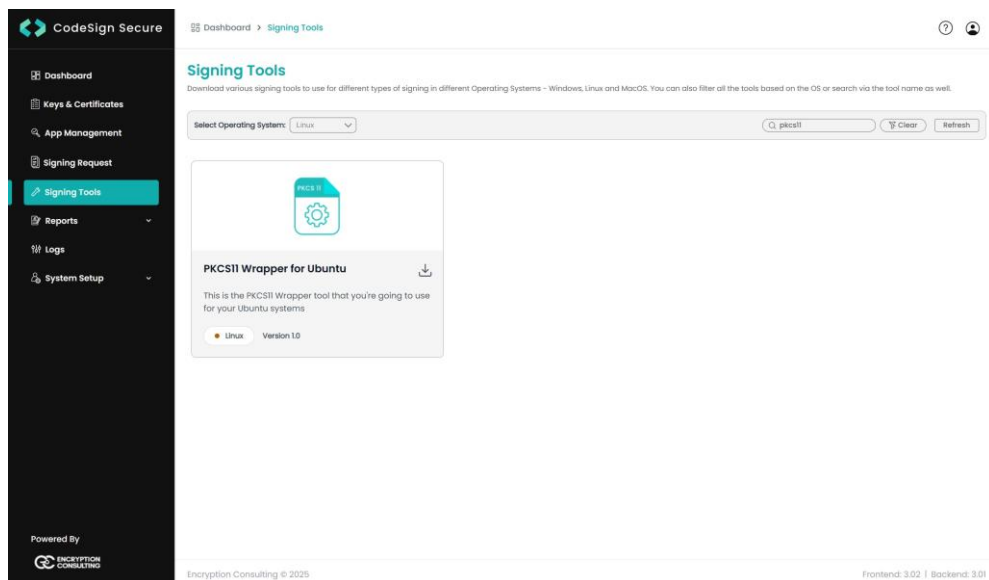
2. Install the EC PKCS#11 Wrapper

The wrapper exposes your HSM keys through the standard PKCS#11 interface, which is what GPG's smartcard daemon will talk to later in this guide.

Steps:

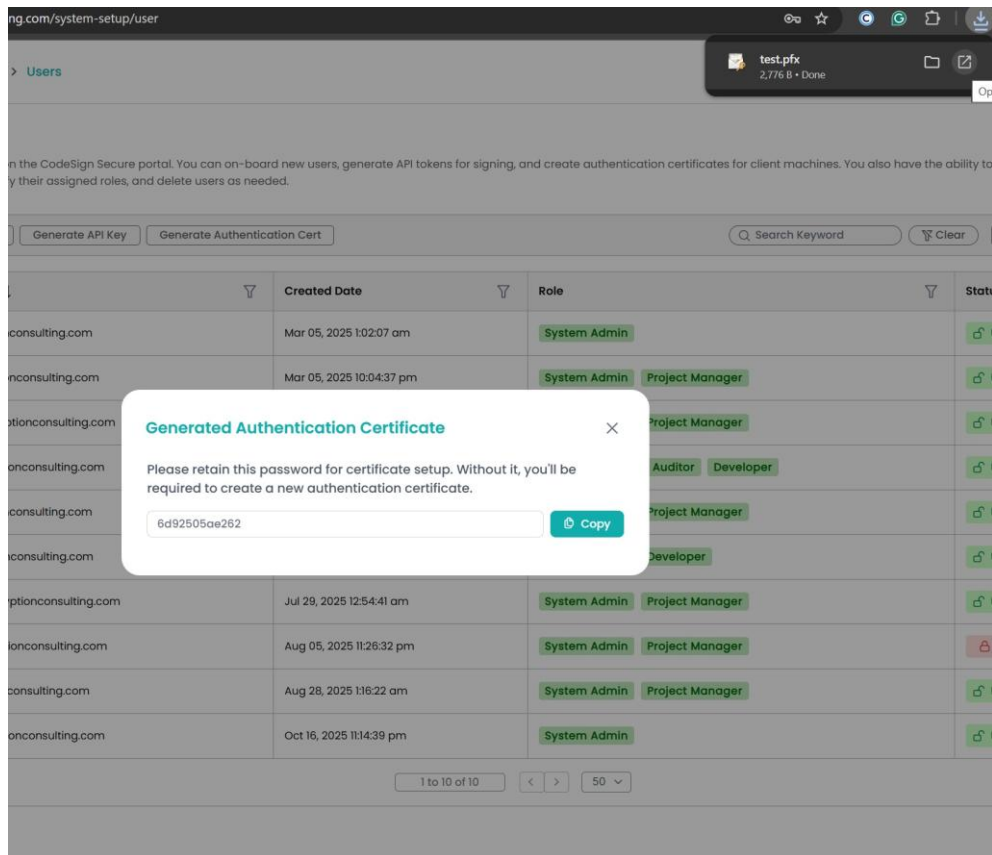
Step 1: Download the PKCS#11 Wrapper for Ubuntu

- Go to EC CodeSign Secure v3.02's Signing Tools section and download the PKCS11 Wrapper for Ubuntu.



Step 2: Generate a P12 Authentication Certificate

- After that, generate a P12 Authentication certificate from the System Setup > User > Generate Authentication Certificate dropdown.



NOTE: The certificate password is displayed only once, so copy and store it safely. Both the certificate path and its password are entered into `ec_pkcs11client.ini` in the next step.

Step 3: Edit the Configuration Files

- Go to your Ubuntu client system and edit the configuration files (`ec_pkcs11client.ini` and `pkcs11properties.cfg`) downloaded in the PKCS11 Wrapper.

```

aryan@test:~/PKCS_Wrapper_Ubuntu$ cat ec_pkcs11client.ini
[logconfig_file]
name=/home/aryan/PKCS_Wrapper_Ubuntu/EC_PKCS11_CLIENT-LogConfig.xml

[server_url]
url=https://devcodesignsecure.encryptionconsulting.com

[pfxfile_path]
path=/home/aryan/PKCS_Wrapper_Ubuntu/Pkcs11Test.pfx

[pfxfile_passwd]
passwd=de48b6924cb8

[login_token]
username=Aryan@encryptionconsulting.com
passcode=8@!$U3QXS5Xtu2vf
aryan@test:~/PKCS_Wrapper_Ubuntu$ cat pkcs11properties.cfg
name=signingmanager
library=/home/aryan/PKCS_Wrapper_Ubuntu/ec_pkcs11client.so
attributes=compatibility
attributes(*, *, *) = {
CKA_TOKEN=true
}
aryan@test:~/PKCS_Wrapper_Ubuntu$ █

```

NOTE: When using Luna HSM, add the slot number in the pkcs11properties.cfg file as shown below.

```

aryan@test:~/PKCS_Wrapper_Ubuntu$ cat pkcs11properties.cfg
name=signingmanager
library=/home/aryan/PKCS_Wrapper_Ubuntu/ec_pkcs11client.so
slot=3
attributes=compatibility
attributes(*, *, *) = {
CKA_TOKEN=true
}

```

Step 4: Set the EC_INI_FILE_PATH Environment Variable

- Now we need to add the environment variable for the pkcs11 client library.
- Run the below command to open the bashrc file:

```
nano ~/.bashrc
```

```

root@Signing:/home/aryan# nano ~/.bashrc
root@Signing:/home/aryan# █

```

- Now add the EC_INI_FILE_PATH variable with the path of the .ini at the end of the bashrc file, like:

```
export EC_INI_FILE_PATH=\
/home/aryan/pkcs11-client/ec_pkcs11client.ini
```

```
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#   . /etc/bash_completion
#fi
export EC_INI_FILE_PATH=/home/aryan/pkcs11-client/ec_pkcs11client.ini
```

Help Write Out Where Is Cut Execute
Exit Read File Replace Paste Justify Go

- Press Ctrl+X, then enter Y, and then press Enter to save.
- Reload the environment variables:

```
source ~/.bashrc
```

```
root@Signing:/home/aryan# source ~/.bashrc
root@Signing:/home/aryan#
```

- Check whether the variable has been set correctly:

```
echo $EC_INI_FILE_PATH
```

```
root@Signing:/home/aryan# echo $EC_INI_FILE_PATH
/home/aryan/pkcs11-client/ec_pkcs11client.ini
root@Signing:/home/aryan#
```

NOTE: There must be no space between the dollar sign and the variable name, or the shell will print a literal dollar sign instead of the value. If you cannot see the file location from the echo command, open a new terminal and try again.

3. Install GnuPG 2.5.4

GnuPG 2.5.4 is required. If your distribution does not package that version, it must be built from source, which is what the remainder of this section and Section 4 cover.

Steps:

Step 1: Check the Packaged Version

- Check the default version of gnupg in your ubuntu machine:

```
sudo apt-cache policy gnupg
```

```
root@Signing:/home/aryan# sudo apt-cache policy gnupg
gnupg:
  Installed: 2.4.4-2ubuntu17
  Candidate: 2.4.4-2ubuntu17
  Version table:
 *** 2.4.4-2ubuntu17 500
      500 http://in.archive.ubuntu.com/ubuntu noble/main amd64 Packages
      500 http://in.archive.ubuntu.com/ubuntu noble/main i386 Packages
      100 /var/lib/dpkg/status
root@Signing:/home/aryan#
```

- If the 2.5.4 version is present in your system, you can run the below command and go to Section 5:

```
sudo apt install gnupg
```

- Since this version is not present, you would need to perform the following steps.

Step 2: Install the Build Tools

- Install the required build tools and dependencies for the gnupg package.

```
sudo apt install build-essential bzip2 libassuan-dev \
  libgcrypt20-dev libgpg-error-dev libksba-dev libnpth0-dev
```

```
root@Signing:/home/aryan# sudo apt install build-essential bzip2 libassuan-dev libgcrypt20-dev libgpg-error-dev libksba-dev libnpth0-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bzip2 is already the newest version (1.0.8-5.1build0.1).
bzip2 set to manually installed.
The following package was automatically installed and is no longer required:
  libllvm17t64
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu dpkg-dev fakeroot g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc
```

Step 3: Download and Extract GnuPG

- Download the 2.5.4 version of gnupg using the below command:

```
wget https://www.gnupg.org/ftp/gcrypt/gnupg/gnupg-2.5.4.tar.bz2
```

```

root@Signing:/home/aryan# wget https://www.gnupg.org/ftp/gcrypt/gnupg/gnupg-2.5.4.tar.bz2
--2025-03-18 05:05:47-- https://www.gnupg.org/ftp/gcrypt/gnupg/gnupg-2.5.4.tar.bz2
Resolving www.gnupg.org (www.gnupg.org)... 217.69.76.60, 2001:aa8:fff1:2100::60
Connecting to www.gnupg.org (www.gnupg.org)|217.69.76.60|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8174777 (7.8M) [text/plain]
Saving to: 'gnupg-2.5.4.tar.bz2'

gnupg-2.5.4.tar.bz2      100%[=====] 7.80M  370KB/s  in 22s

2025-03-18 05:06:11 (369 KB/s) - 'gnupg-2.5.4.tar.bz2' saved [8174777/8174777]

```

- Extract the Archive:

```
tar -xjvf gnupg-2.5.4.tar.bz2
```

```

root@Signing:/home/aryan# tar -xjvf gnupg-2.5.4.tar.bz2
gnupg-2.5.4/
gnupg-2.5.4/g13/
gnupg-2.5.4/g13/all-tests.scm
gnupg-2.5.4/g13/g13tuple.c
gnupg-2.5.4/g13/sh-cmd.c
gnupg-2.5.4/g13/be-dmccrypt.c
gnupg-2.5.4/g13/be-encfs.c
gnupg-2.5.4/g13/Makefile.in
gnupg-2.5.4/g13/sh-blockdev.c
gnupg-2.5.4/g13/be-truecrypt.c
gnupg-2.5.4/g13/g13-common.h
gnupg-2.5.4/g13/suspend.c

```

- Go to the newly created gnupg-2.5.4 directory:

```
cd gnupg-2.5.4
```

```

root@Signing:/home/aryan# cd gnupg-2.5.4
root@Signing:/home/aryan/gnupg-2.5.4# ls
ABOUT-NLS  AUTHORS  ChangeLog-2011  COPYING  COPYING.other  INSTALL  NEWS  scd  tools
acinclude.m4  autogen.rc  common  COPYING.CC0  dirmgr  kbx  po  sm  tpm2d
aclocal.m4  autogen.sh  config.h.in  COPYING.GPL2  doc  m4  README  tests  VERSION
agent  build-aux  configure  COPYING.LGPL21  g10  Makefile.am  README.GIT  THANKS
am  ChangeLog  configure.ac  COPYING.LGPL3  g13  Makefile.in  regexp  TODO
root@Signing:/home/aryan/gnupg-2.5.4#

```

Step 4: Run Configure and Read the Output

- Run the configure file:

```
./configure
```

```

root@Signing:/home/aryan/gnupg-2.5.4# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
configure: autobuild project... gnupg

```

- If this release of GnuPG requires some newer shared libraries than you have installed, the output of the configure command will tell you, as shown in the image below.

```

configure:
***
*** You need libgpg-error to build this program.
** This library is for example available at
***   https://gnupg.org/ftp/gcrypt/gpg-error/
*** (at least version 1.51 is required.)
***
configure:
***
*** You need libgcrypt to build this program.
** This library is for example available at
***   https://gnupg.org/ftp/gcrypt/libgcrypt/
*** (at least version 1.11.0 (API 1) is required.)
***
configure:
***
*** You need libassuan to build this program.
*** This library is for example available at
***   https://gnupg.org/ftp/gcrypt/libassuan/
*** (at least version 3.0.0 (API 3) is required.)
***
configure: error:
***
*** Required libraries not found. Please consult the above messages
*** and install them before running configure again.
***

```

- The above output of the configure command tells us we need to build and install the latest version of the libgpg-error, libgcrypt, and libassuan libraries. These are installed in Section 4.

4. Install the Additional Dependencies for GPG2

Each of the libraries below is built the same way: download, extract, configure, make, install, refresh the linker cache, and return to the GnuPG directory. Install them in the order given, because later libraries depend on earlier ones.

Steps:

Step 1: Install libpgp-error

- Install the libpgp-error version from the link provided as the configure command output.

-

```
wget https://gnupg.org/ftp/gcrypt/gpgmt/libpgp-error-1.51.tar.bz2
```

```
root@Signing:/home/aryan/gnupg-2.5.4# wget https://gnupg.org/ftp/gcrypt/gpgmt/libpgp-error-1.51.tar.bz2
--2025-03-18 05:09:37-- https://gnupg.org/ftp/gcrypt/gpgmt/libpgp-error-1.51.tar.bz2
Resolving gnupg.org (gnupg.org)... 217.69.76.60, 2001:aa8:fff1:2100::60
Connecting to gnupg.org (gnupg.org)[217.69.76.60]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1085510 (1.0M) [text/plain]
Saving to: 'libpgp-error-1.51.tar.bz2'

libpgp-error-1.51.tar.bz2      100%[=====] 1.04M  290KB/s  in 3.7s

2025-03-18 05:09:42 (290 KB/s) - 'libpgp-error-1.51.tar.bz2' saved [1085510/1085510]
```

- Extract the archive:

```
tar xf libpgp-error-1.51.tar.bz2
```

```
root@Signing:/home/aryan/gnupg-2.5.4# tar xf libpgp-error-1.51.tar.bz2
root@Signing:/home/aryan/gnupg-2.5.4# ls
ABOUT-NLS  autogen.rc  config.h.in  COPYING.GPL2  g10  m4  README.GIT  TODO
acinclude.m4  autogen.sh  config.log  COPYING.LGPL21  g13  Makefile.am  regexp  tools
aclocal.m4  build-aux  configure  COPYING.LGPL3  INSTALL  Makefile.in  scd  tpm2d
agent  ChangeLog  configure.ac  COPYING.other  kbx  NEWS  sm  VERSION
am  ChangeLog-2011  COPYING  dirnagr  libpgp-error-1.51  po  tests
AUTHORS  common  COPYING.CC0  doc  libpgp-error-1.51.tar.bz2  README  THANKS
```

- Go to the newly created libpgp-error-1.51 directory:

```
cd libpgp-error-1.51/
```

```
root@Signing:/home/aryan/gnupg-2.5.4# cd libpgp-error-1.51/
root@Signing:/home/aryan/gnupg-2.5.4/libpgp-error-1.51# ls
ABOUT-NLS  autogen.rc  ChangeLog  configure  COPYING.LIB  lang  m4  mkinstalldirs  README  THANKS
aclocal.m4  autogen.sh  ChangeLog-2011  configure.ac  doc  libpgp-error.spec  Makefile.am  NEWS  src  VERSION
AUTHORS  build-aux  config.h.in  COPYING  INSTALL  libpgp-error.spec.in  Makefile.in  po  tests
```

- Run the configure file:

```
./configure
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libgpg-error-1.51# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... no
checking whether make supports nested variables... (cached) yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
configure: autobuild project... libgpg-error
configure: autobuild revision... 1.51
configure: autobuild hostname... Signing

```

```

config.status: creating src/gpg-error-config-old
config.status: creating src/gpg-error-config
config.status: creating src/gpg-error-config-test.sh
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile

libgpg-error v1.51 has been configured as follows:

Revision: b0bb926 (45243)
Platform: x86_64-pc-linux-gnu

```

- Run the make command:

```
make
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libgpg-error-1.51# make
make all-recursive
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libgpg-error-1.51'
Making all in m4
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libgpg-error-1.51/m4'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/libgpg-error-1.51/m4'
Making all in src
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libgpg-error-1.51/src'
mawk -f ./mkerrnos.awk ./errnos.in >code-to-errno.h
mawk -f ./mkerrcodes1.awk ./errnos.in >_mkerrcodes.h
gcc -E -P _mkerrcodes.h | grep GPG_ERR_ | \
    mawk -f ./mkerrcodes.awk >mkerrcodes.h
rm _mkerrcodes.h

```

- Install the package:

```
sudo make install
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libgpg-error-1.51# sudo make install
Making install in m4
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libgpg-error-1.51/m4'
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libgpg-error-1.51/m4'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/libgpg-error-1.51/m4'
make[1]: Leaving directory '/home/aryan/gnupg-2.5.4/libgpg-error-1.51/m4'
Making install in src
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libgpg-error-1.51/src'
make install-am
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libgpg-error-1.51/src'
make[3]: Entering directory '/home/aryan/gnupg-2.5.4/libgpg-error-1.51/src'
/usr/bin/mkdir -p '/usr/local/lib'

```

- Update the system's dynamic linker cache using the below command:

```
sudo ldconfig
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libgpg-error-1.51# sudo ldconfig
root@Signing:/home/aryan/gnupg-2.5.4/libgpg-error-1.51#

```

- Go back one directory (../gnupg-2.5.4) to install other libraries:

```
cd ..
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libgpg-error-1.51# cd ..
root@Signing:/home/aryan/gnupg-2.5.4# ls
ABOUT-NLS      autogen.rc      config.h.in     COPYING.GPL2    g10             m4              README.GIT      TODO
acinclude.m4    autogen.sh      config.log      COPYING.LGPL21  g13             Makefile.am     regexp          tools
aclocal.m4      build-aux       configure       COPYING.LGPL3   INSTALL         Makefile.in     scd             tpm2d
agent           ChangeLog       configure.ac    COPYING.other   kbx             NEWS            sm              VERSION
api             ChangeLog-2011  COPYING        dirnmgr         libgpg-error-1.51  po             tests
AUTHORS         common          COPYING.CC0    doc             libgpg-error-1.51.tar.bz2  README          THANKS

```

Step 2: Install libgcrypt

- Install the libgcrypt version from the link provided as the configure command output.

-

```
wget https://gnupg.org/ftp/gcrypt/libgcrypt/libgcrypt-1.11.0.tar.bz2
```

```

root@Signing:/home/aryan/gnupg-2.5.4# wget https://gnupg.org/ftp/gcrypt/libgcrypt/libgcrypt-1.11.0.tar.bz2
--2025-03-18 05:12:58-- https://gnupg.org/ftp/gcrypt/libgcrypt/libgcrypt-1.11.0.tar.bz2
Resolving gnupg.org (gnupg.org)... 217.69.76.60, 2001:aa8:fff1:2100::60
Connecting to gnupg.org (gnupg.org)[217.69.76.60]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4180345 (4.0M) [text/plain]
Saving to: 'libgcrypt-1.11.0.tar.bz2'

libgcrypt-1.11.0.tar.bz2  100%[=====] 3.99M  351KB/s  in 13s

2025-03-18 05:13:12 (321 KB/s) - 'libgcrypt-1.11.0.tar.bz2' saved [4180345/4180345]

```

- Extract the archive:

```
tar xf libgcrypt-1.11.0.tar.bz2
```

```
root@Signing:/home/aryan/gnupg-2.5.4# tar xf libgcrypt-1.11.0.tar.bz2
root@Signing:/home/aryan/gnupg-2.5.4# ls
ABOUT-NLS      autogen.rc      config.h.in     COPYING.GPL2    g10             libgpg-error-1.51  po             tests
acinclude.m4    autogen.sh      config.log      COPYING.LGPL21  g13             libgpg-error-1.51.tar.bz2  README         THANKS
aclocal.m4      build-aux       configure       COPYING.LGPL3   INSTALL        md               README.GIT     TODO
agent          ChangeLog       configure.ac    COPYING.other   kbx            Makefile.am       regexp         tools
an             ChangeLog-2011  COPYING        dirmngr        libgcrypt-1.11.0  Makefile.in       scd           tpm2d
AUTHORS        common         COPYING.CC0    doc            libgcrypt-1.11.0.tar.bz2  NEWS           sm            VERSION
```

- Go to the newly created libgcrypt-1.11.0 directory:

```
cd libgcrypt-1.11.0/
```

```
root@Signing:/home/aryan/gnupg-2.5.4# cd libgcrypt-1.11.0/
root@Signing:/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0# ls
acinclude.m4    autogen.sh      cipher          configure.ac    INSTALL        Makefile.in      random         tests
aclocal.m4      build-aux       compat         COPYING        LICENSES       mkinstalldirs   README         THANKS
AUTHORS         ChangeLog       config.h.in    COPYING.LIB     m4            mpi             README.GIT     TODO
autogen.rc      ChangeLog-2011  configure      doc            Makefile.am    NEWS           src            VERSION
```

- Run the configure file:

```
./configure
```

```
root@Signing:/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking whether to enable maintainer-specific portions of Makefiles... no
```

```

Platform: GNU/Linux (x86_64-pc-linux-gnu)
Hardware detection module: libgcrypt_la-hwf-x86
Enabled cipher algorithms: arcfour blowfish cast5 des aes twofish
                        serpent rfc2268 seed camellia idea salsa20
                        gost28147 chacha20 sm4 aria
Enabled digest algorithms: crc gostr3411-94 md4 md5 rmd160 sha1
                        sha256 sha512 sha3 tiger whirlpool sribog
                        blake2 sm3
Enabled kdf algorithms: s2k pkdf2 scrypt
Enabled pubkey algorithms: dsa elgamal rsa ecc
Random number generator: default
Try using jitter entropy: yes
Using linux capabilities: no
FIPS module version:
Try using Padlock crypto: yes
Try using AES-NI crypto: yes
Try using Intel SHAEXT: yes
Try using Intel PCLMUL: yes
Try using Intel SSE4.1: yes
Try using DRNG (RDRAND): yes
Try using Intel AVX: yes
Try using Intel AVX2: yes
Try using Intel AVX512: yes
Try using Intel GFNI: yes
Try using ARM NEON: n/a
Try using ARMv8 crypto: n/a
Try using ARMv8 SVE: n/a
Try using ARMv9 SVE2: n/a
Try using PPC crypto: n/a

```

- Run the make command:

```
make
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0# make
make all-recursive
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0'
Making all in compat
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0/compat'
/bin/bash ../libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I.. -I../src -I../src -I/usr/local/include -g -O2 -fvisibility=hidden -fno-delete-null-pointer-checks -Wall -MT compat.lo -MD -MP -MF .deps/compat.Tpo -c -o compat.lo compat.c
libtool: compile: gcc -DHAVE_CONFIG_H -I. -I.. -I../src -I../src -I/usr/local/include -g -O2 -fvisibility=hidden -fno-delete-null-pointer-checks -Wall -MT compat.lo -MD -MP -MF .deps/compat.Tpo -c compat.c -fPIC -DPIC -o .libs/compat.o
mv -f .deps/compat.Tpo .deps/compat.Plo
/bin/bash ../libtool --tag=CC --mode=link gcc -g -O2 -fvisibility=hidden -fno-delete-null-pointer-checks -Wall -o libcompat.la compat.lo
libtool: link: gcc -g -O2 -fvisibility=hidden -fno-delete-null-pointer-checks -Wall -o libcompat.la compat.o

```

- Install the package:

```
sudo make install
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0# sudo make install
Making install in compat
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0/compat'
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0/compat'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0/compat'
make[1]: Leaving directory '/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0/compat'
Making install in mpi
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0/mpi'
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0/mpi'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0/mpi'

```

- Update the system's dynamic linker cache using the below command:

```
sudo ldconfig
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0# sudo ldconfig
root@Signing:/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0#

```

- Go back one directory (../gnupg-2.5.4) to install other libraries:

```
cd ..
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libgcrypt-1.11.0# cd ..
root@Signing:/home/aryan/gnupg-2.5.4# ls
ABOUT-NLS      autogen.rc      config.h.in     COPYING.GPL2    g10             libgpg-error-1.51  po             tests
acinclude.m4    autogen.sh      config.log      COPYING.LGPL21  g13             libgpg-error-1.51.tar.bz2  README        THANKS
aclocal.m4      build-aux       configure       COPYING.LGPL3   INSTALL         m4                 README.GIT    TODO
agent           ChangeLog       configure.ac    COPYING.other   kbx             Makefile.am        regex         tools
am              ChangeLog-2011  COPYING       dirnmgr         libgcrypt-1.11.0  Makefile.in        scd           tpm2d
AUTHORS         common          COPYING.CC0    doc             libgcrypt-1.11.0.tar.bz2  NEWS          sm           VERSION

```

Step 3: Install libassuan

- Install the libassuan version from the link provided as the configure command output.
-

```
wget https://gnupg.org/ftp/gcrypt/libassuan/libassuan-3.0.0.tar.bz2
```

```

root@Signing:/home/aryan/gnupg-2.5.4# wget https://gnupg.org/ftp/gcrypt/libassuan/libassuan-3.0.0.tar.bz2
--2025-03-18 05:20:08-- https://gnupg.org/ftp/gcrypt/libassuan/libassuan-3.0.0.tar.bz2
Resolving gnupg.org (gnupg.org)... 217.69.76.60, 2001:aa8:fff1:2100::60
Connecting to gnupg.org (gnupg.org)|217.69.76.60|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 592353 (578K) [text/plain]
Saving to: 'libassuan-3.0.0.tar.bz2'

libassuan-3.0.0.tar.bz2      100%[=====] 578.47K  297KB/s   in 1.9s
2025-03-18 05:20:11 (297 KB/s) - 'libassuan-3.0.0.tar.bz2' saved [592353/592353]

```

- Extract the archive:

```
tar xf libassuan-3.0.0.tar.bz2
```

```
root@Signing:/home/aryan/gnupg-2.5.4# tar xf libassuan-3.0.0.tar.bz2
root@Signing:/home/aryan/gnupg-2.5.4# ls
ABOUT-NLS  autogen.sh  configure  COPYING.other  libassuan-3.0.0  Makefile.am  scd  VERSION
acinclude.m4  build-aux  configure.ac  dirnagr  libassuan-3.0.0.tar.bz2  Makefile.in  sn
aclocal.m4  ChangeLog  COPYING  doc  libgcrypt-1.11.0  NEWS  tests
agent  ChangeLog-2011  COPYING.CC0  g10  libgcrypt-1.11.0.tar.bz2  po  THANKS
an  common  COPYING.GPL2  g13  libgpg-error-1.51  README  TODO
AUTHORS  config.h.in  COPYING.LGPL21  INSTALL  libgpg-error-1.51.tar.bz2  README.GIT  tools
autogen.rc  config.log  COPYING.LGPL3  kbx  m4  regexp  tpm2d
```

- Go to the newly created libassuan-3.0.0 directory:

```
cd libassuan-3.0.0/
```

```
root@Signing:/home/aryan/gnupg-2.5.4# cd libassuan-3.0.0/
root@Signing:/home/aryan/gnupg-2.5.4/libassuan-3.0.0# ls
aclocal.m4  autogen.sh  ChangeLog-2011  configure.ac  doc  Makefile.am  NEWS  src  TODO
AUTHORS  build-aux  config.h.in  COPYING  INSTALL  Makefile.in  README  tests  VERSION
autogen.rc  ChangeLog  configure  COPYING.LIB  m4  mkinstalldirs  README.GIT  THANKS
```

- Run the configure file:

```
./configure
```

```
root@Signing:/home/aryan/gnupg-2.5.4/libassuan-3.0.0# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... no
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
```

```

config.status: creating src/libassuan-config
config.status: creating src/versioninfo.rc
config.status: creating src/libassuan.pc
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands

    Libassuan v3.0.0 has been configured as follows:

    Revision: 0351ecf (849)
    Platform: x86_64-pc-linux-gnu

```

- Run the make command:

```
make
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libassuan-3.0.0# make
make all-recursive
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0'
Making all in m4
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0/m4'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0/m4'
Making all in src
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0/src'
gcc \
    -I. -I. -o mkheader ./mkheader.c
./mkheader linux-gnu ./assuan.h.in \
    3.0.0 0x030000 >assuan.h
make all-am
make[3]: Entering directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0/src'
/bin/bash ../libtool --tag=CC --mode=compile gcc -DHAVE_CONFIG_H -I. -I.. -I.. -I/usr/local/include

```

- Install the package:

```
sudo make install
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libassuan-3.0.0# sudo make install
Making install in m4
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0/m4'
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0/m4'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0/m4'
make[1]: Leaving directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0/m4'
Making install in src
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0/src'
make install-am
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0/src'
make[3]: Entering directory '/home/aryan/gnupg-2.5.4/libassuan-3.0.0/src'
/usr/bin/mkdir -p '/usr/local/lib'

```

- Update the system's dynamic linker cache using the below command:

```
sudo ldconfig
```

```
root@Signing:/home/aryan/gnupg-2.5.4/libassuan-3.0.0# sudo ldconfig
root@Signing:/home/aryan/gnupg-2.5.4/libassuan-3.0.0#
```

- Go back one directory (../gnupg-2.5.4) to install other libraries:

```
cd ..
```

```
root@Signing:/home/aryan/gnupg-2.5.4/libassuan-3.0.0# cd ..
root@Signing:/home/aryan/gnupg-2.5.4# ls
ABOUT-NLS      autogen.sh      configure        COPYING.other   libassuan-3.0.0  Makefile.am     scd             VERSION
acinclude.m4    build-aux       configure.ac     dirmngr        libassuan-3.0.0.tar.bz2  Makefile.in     sm
aclocal.m4      ChangeLog       COPYING         doc            libgcrypt-1.11.0  NEWS           tests
agent          ChangeLog-2011  COPYING.CC0     g10            libgcrypt-1.11.0.tar.bz2  po             THANKS
an             common         COPYING.GPL2    g13            libgpg-error-1.51  README         TODO
AUTHORS         config.h.in     COPYING.LGPL21  INSTALL        libgpg-error-1.51.tar.bz2  README.GIT     tools
autogen.rc      config.log     COPYING.LGPL3   kbx            m4                 regexp         tpm2d
```

Step 4: Run Configure Again

- Run the configure file from the gnupg-2.5.4 directory:

```
./configure
```

```
root@Signing:/home/aryan/gnupg-2.5.4# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
configure: autobuild project... gnupg
configure: autobuild revision... 2.5.4
```

- Check the output of the above command.

```

configure:
***
*** You need libksba to build this program.
*** This library is for example available at
*** https://gnupg.org/ftp/gcrypt/libksba/
*** (at least version 1.6.3 using API 1 is required).
***
configure: error:
***
*** Required libraries not found. Please consult the above messages
*** and install them before running configure again.
***

```

- The above output of the configure command again tells us we need to build and install another library with the latest version, for libksba.

Step 5: Install libksba

- Install the libksba version from the link provided as the configure command output.
-

```
wget https://gnupg.org/ftp/gcrypt/libksba/libksba-1.6.3.tar.bz2
```

```

root@Signing:/home/aryan/gnupg-2.5.4# wget https://gnupg.org/ftp/gcrypt/libksba/libksba-1.6.3.tar.bz2
--2025-03-18 05:24:50-- https://gnupg.org/ftp/gcrypt/libksba/libksba-1.6.3.tar.bz2
Resolving gnupg.org (gnupg.org)... 217.69.76.60, 2001:aa8:fff1:2100::60
Connecting to gnupg.org (gnupg.org)|217.69.76.60|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 668287 (653K) [text/plain]
Saving to: 'libksba-1.6.3.tar.bz2'

libksba-1.6.3.tar.bz2      100%[=====] 652.62K  264KB/s   in 2.5s
2025-03-18 05:24:53 (264 KB/s) - 'libksba-1.6.3.tar.bz2' saved [668287/668287]

```

- Extract the archive:

```
tar xf libksba-1.6.3.tar.bz2
```

```

root@Signing:/home/aryan/gnupg-2.5.4# tar xf libksba-1.6.3.tar.bz2
root@Signing:/home/aryan/gnupg-2.5.4# ls
ABOUT-NLS  autogen.sh  configure  COPYING.other  libassuan-3.0.0  libksba-1.6.3.tar.bz2  README.GIT  tools
acinclude.m4  build-aux  configure.ac  dirnng  libassuan-3.0.0.tar.bz2  m4  Makefile.am  regex
aclocal.m4  ChangeLog  COPYING  doc  libgcrypt-1.11.0  NEWS  Makefile.in  scd
agent  ChangeLog-2011  COPYING.CC0  g10  libgcrypt-1.11.0.tar.bz2  NEWS  po  sm
am  common  COPYING.GPL2  g13  libgpg-error-1.51  NEWS  po  tests
AUTHORS  config.h.in  COPYING.LGPL21  INSTALL  libgpg-error-1.51.tar.bz2  po  README  THANKS
autogen.rc  config.log  COPYING.LGPL3  kbx  libksba-1.6.3  README  TODO

```

- Go to the newly created libksba-1.6.3 directory:

```
cd libksba-1.6.3/
```

```
root@Signing:/home/aryan/gnupg-2.5.4# cd libksba-1.6.3/
root@Signing:/home/aryan/gnupg-2.5.4/libksba-1.6.3# ls
aclocal.m4  autogen.sh  ChangeLog-2011  configure.ac  COPYING.GPLv3  gl  Makefile.am  NEWS  tests
AUTHORS    build-aux   config.h.in     COPYING      COPYING.LGPLv3  INSTALL  Makefile.in  README  THANKS
autogen.rc  ChangeLog  configure       COPYING.GPLv2  doc            m4      mkinstalldirs  src    TODO
```

- Run the configure file:

```
./configure
```

```
root@Signing:/home/aryan/gnupg-2.5.4/libksba-1.6.3# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
```

```
config.status: executing depfiles commands
config.status: executing libtool commands
configure:
***
*** Note: The installed yacc version is not GNU Bison.  You need
*** to install Bison if you want to change any grammar (.y) file.
***

Libksba v1.6.3 has been configured as follows:

Revision:  bffa9b3  (49146)
Platform:  x86_64-pc-linux-gnu
```

- Run the make command:

```
make
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libksba-1.6.3# make
make all-recursive
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libksba-1.6.3'
Making all in m4
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libksba-1.6.3/m4'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/libksba-1.6.3/m4'
Making all in gl
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libksba-1.6.3/gl'
cp ./alloca_.h alloca.h-t
mv alloca.h-t alloca.h
make all-am

```

- Install the package:

```
sudo make install
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libksba-1.6.3# sudo make install
Making install in m4
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libksba-1.6.3/m4'
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/libksba-1.6.3/m4'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/libksba-1.6.3/m4'
make[1]: Leaving directory '/home/aryan/gnupg-2.5.4/libksba-1.6.3/m4'
Making install in gl
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/libksba-1.6.3/gl'
make install-am

```

- Update the system's dynamic linker cache using the below command:

```
sudo ldconfig
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libksba-1.6.3# sudo ldconfig
root@Signing:/home/aryan/gnupg-2.5.4/libksba-1.6.3#

```

- Go back one directory (../gnupg-2.5.4) to install other libraries:

```
cd ..
```

```

root@Signing:/home/aryan/gnupg-2.5.4/libksba-1.6.3# cd ..
root@Signing:/home/aryan/gnupg-2.5.4# ls
ABOUT-NLS      autogen.sh      configure      COPYING.other  libassuan-3.0.0  libksba-1.6.3.tar.bz2  README.GIT  tools
acinclude.m4    build-aux       configure.ac   dirnagr       libassuan-3.0.0.tar.bz2  m4                    regexp      tpm2d
aclocal.m4      ChangeLog       COPYING       doc           libgcrypt-1.11.0  Makefile.am          scd         VERSION
agent          ChangeLog-2011  COPYING.CC0   g10          libgcrypt-1.11.0.tar.bz2  Makefile.in          sm         tests
am             common         COPYING.GPL2  g13          libgpg-error-1.51  NEWS                 tests       THANKS
AUTHORS        config.h.in     COPYING.LGPL21  INSTALL      libgpg-error-1.51.tar.bz2  po                   TODO
autogen.rc     config.log      COPYING.LGPL3  kbx         libksba-1.6.3      README

```

5. Complete the GnuPG Installation

With every dependency in place, the configure command should now finish without reporting a missing library, and GnuPG can be built and installed.

Steps:

Step 1: Run Configure Once More

- Run the configure file again from the gnupg-2.5.4 directory:

```
./configure
```

```
root@Signing:/home/aryan/gnupg-2.5.4# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
configure: autobuild project... gnupg
configure: autobuild revision... 2.5.4
configure: autobuild hostname... Signing
configure: autobuild timestamp... 20250318-052817
```

- Check the output of the above command.

```
Revision: ec6c2e38f (60524)
Platform: GNU/Linux (x86_64-pc-linux-gnu)

OpenPGP: yes
S/MIME: yes
Agent: yes
Smartcard: yes (without internal CCID driver)
TPM: no
G13: no
Dirmngr: no
Keyboxd: no
Gpgtar: yes
WKS tools: yes

Protect tool: (default)
LDAP wrapper: (default)
Default agent: (default)
Default pinentry: (default)
Default scdaemon: (default)
Default keyboxd: (default)
Default tpm2daemon: (default)
Default dirmngr: (default)

Dirmngr auto start: yes
Readline support: no
LDAP support: n/a
TLS support: no
TOFU support: no
Tor support: only .onion
```

- Since there is no error of package version, we can move forward with the installation.

Step 2: Build and Install GnuPG

- Run the make command:

```
make
```

```
root@Signing:/home/aryan/gnupg-2.5.4# make
make all-recursive
make[1]: Entering directory '/home/aryan/gnupg-2.5.4'
Making all in m4
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/m4'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/m4'
Making all in common
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/common'
make all-am
make[3]: Entering directory '/home/aryan/gnupg-2.5.4/common'
gcc -DHAVE_CONFIG_H -I. -I.. -DLOCALEDIR="/usr/local/share/locale" -DGNUPG_BINDIR="/usr/local/bin" -DGNUPG_LIBEXECDIR="/usr/local/libexec" -DGNUPG_LIBDIR="/usr/local/lib/gnupg" -DGNUPG_DATADIR="/usr/local/share/gnupg" -DGNUPG_SYSCONFDIR="/usr/l
```

- Install the package:

```
sudo make install
```

```

root@Signing:/home/aryan/gnupg-2.5.4# sudo make install
Making install in m4
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/m4'
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/m4'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/m4'
make[1]: Leaving directory '/home/aryan/gnupg-2.5.4/m4'
Making install in common
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/common'
make install-am
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/common'
make[3]: Entering directory '/home/aryan/gnupg-2.5.4/common'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/home/aryan/gnupg-2.5.4/common'
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/common'
make[1]: Leaving directory '/home/aryan/gnupg-2.5.4/common'

```

Step 3: Verify the Installation

- Check whether gpg has been installed correctly or not:

```
gpg --version
```

```

root@Signing:/home/aryan/gnupg-2.5.4# gpg --version
gpg (GnuPG) 2.5.4
libgcrypt 1.11.0
Copyright (C) 2025 g10 Code GmbH
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /root/.gnupg
Supported algorithms:
Pubkey: RSA, Kyber, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed

```

- Check whether gpg-agent has been installed correctly or not:

```
gpg-agent --version
```

```

root@Signing:/home/aryan/gnupg-2.5.4# gpg-agent --version
gpg-agent (GnuPG) 2.5.4
libgcrypt 1.11.0
Copyright (C) 2025 g10 Code GmbH
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

```

6. Install gnupg-pkcs11-scd 0.11.0

gnupg-pkcs11-scd replaces GnuPG's own smartcard daemon with one that speaks PKCS#11, which is how GPG reaches the keys in your HSM through the Encryption Consulting wrapper.

Steps:

Step 1: Check the Packaged Version

- Check the default version of gnupg-pkcs11-scd in your ubuntu machine:

```
sudo apt-cache policy gnupg-pkcs11-scd
```

```
root@Signing:/home/aryan/gnupg-2.5.4# sudo apt-cache policy gnupg-pkcs11-scd
gnupg-pkcs11-scd:
  Installed: (none)
  Candidate: 0.10.0-3build2
  Version table:
     0.10.0-3build2 500
                  500 http://in.archive.ubuntu.com/ubuntu noble/universe amd64 Packages
root@Signing:/home/aryan/gnupg-2.5.4#
```

- If the required version, 0.11.0, is installed on your machine, run the below command:

```
sudo apt install gnupg-pkcs11-scd
```

- Otherwise, you would need to perform the following steps.

Step 2: Install the Build Tools

- Install the required build tools and dependencies for the gnupg-pkcs11-scd package.

```
sudo apt install libpcsclite-dev libssl-dev pkg-config \
  libpkcs11-helper1-dev
```

```
root@Signing:/home/aryan/gnupg-2.5.4# sudo apt install libpcsclite-dev libssl-dev pkg-config libpkcs11-helper1-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm17t64
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  javascript-common libjs-jquery libpkgconf3 pkgconf pkgconf-bin
Suggested packages:
  apache2 | lighttpd | httpd libssl-doc
The following NEW packages will be installed:
```

Step 3: Download, Build and Install

- Download the required package:

```
wget https://github.com/alonbl/gnupg-pkcs11-scd/releases/download/gnupg-pkcs11-scd-0.11.0/gnupg-pkcs11-scd-0.11.0.tar.bz2
```

```
root@Signing:/home/aryan/gnupg-2.5.4# wget https://github.com/alonbl/gnupg-pkcs11-scd/releases/download/gnupg-pkcs11-scd-0.11.0/gnupg-pkcs11-scd-0.11.0.tar.bz2
--2025-03-18 05:44:02-- https://github.com/alonbl/gnupg-pkcs11-scd/releases/download/gnupg-pkcs11-scd-0.11.0/gnupg-pkcs11-scd-0.11.0.tar.bz2
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/2983000/ec15664e-a32e-4c91-bc3a-5077fe644060?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250318%2Fus-east-1%2F%3Faws4_request&X-Amz-Date=20250318T001402Z&X-Amz-Expires=300&X-Amz-Signature=6cc3d51a9bf2047466bd0cd89afe6c5fiabd4c75784f1caa624ae79589083498X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dgnupg-pkcs11-scd-0.11.0.tar.bz2&response-content-type=application%2Foctet-stream [following]
--2025-03-18 05:44:03-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/2983000/ec15664e-a32e-4c91-bc3a-5077fe644060?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250318%2Fus-east-1%2F%3Faws4_request&X-Amz-Date=20250318T001402Z&X-Amz-Expires=300&X-Amz-Signature=6cc3d51a9bf2047466bd0cd89afe6c5fiabd4c75784f1caa624ae79589083498X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dgnupg-pkcs11-scd-0.11.0.tar.bz2&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.108.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 154497 (151K) [application/octet-stream]
Saving to: 'gnupg-pkcs11-scd-0.11.0.tar.bz2'

gnupg-pkcs11-scd-0.11.0.tar.bz2 100%[=====] 150.88K --.-KB/s in 0.1s

2025-03-18 05:44:04 (1.27 MB/s) - 'gnupg-pkcs11-scd-0.11.0.tar.bz2' saved [154497/154497]
```

- Extract the Archive:

```
tar xf gnupg-pkcs11-scd-0.11.0.tar.bz2
```

```
root@Signing:/home/aryan/gnupg-2.5.4# tar xf gnupg-pkcs11-scd-0.11.0.tar.bz2
root@Signing:/home/aryan/gnupg-2.5.4# ls
ABOUT-NLS      build-aux      configure.ac    etc            libexec        Makefile.am    stamp-h1
acinclude.m4    ChangeLog      COPYING        g10           libgcrypt-1.11.0  Makefile.in    tests
aclocal.m4      ChangeLog-2011 COPYING.CC0     g13           libgcrypt-1.11.0.tar.bz2  NEWS          THANKS
agent          common        COPYING.GPL2    gnupg-pkcs11-scd-0.11.0  libgpg-error-1.51  po            TODO
am             config.h       COPYING.LGPL21  gnupg-pkcs11-scd-0.11.0.tar.bz2  libgpg-error-1.51.tar.bz2  README        tools
AUTHORS        config.h.in    COPYING.LGPL3   INSTALL       libksba-1.6.3     README.GIT     tpm2d
autogen.rc     config.log     COPYING.other   kbx           libksba-1.6.3.tar.bz2  regexp        VERSION
autogen.sh     config.status  dirnmgr        libassuan-3.0.0      m4               scd
bin            configure      doc            libassuan-3.0.0.tar.bz2  Makefile         sm
```

- Go to the newly created gnupg-pkcs11-scd-0.11.0 directory:

```
cd ./gnupg-pkcs11-scd-0.11.0/
```

```
root@Signing:/home/aryan/gnupg-2.5.4# cd ./gnupg-pkcs11-scd-0.11.0/
root@Signing:/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0# ls
aclocal.m4      compile      config.sub    COPYING      gnupg-pkcs11-scd      install-sh    Makefile.in  THANKS
AUTHORS        config.guess configure      depcomp     gnupg-pkcs11-scd-proxy  m4           missing
ChangeLog      config.h.in  configure.ac  distro       INSTALL              Makefile.am  README
```

- Run the configure file:

```
./configure
```

```

root@Signing:/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether make supports the include directive... yes (GNU style)
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out

```

- Check the output of the above command.

```

checking whether LOCAL_PEERCREATED is declared... no
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating gnupg-pkcs11-scd-proxy/Makefile
config.status: creating gnupg-pkcs11-scd-proxy/gnupg-pkcs11-scd-proxy.service.in
config.status: creating gnupg-pkcs11-scd/Makefile
config.status: creating distro/Makefile
config.status: creating distro/debian/Makefile
config.status: creating distro/rpm/Makefile
config.status: creating distro/rpm/gnupg-pkcs11-scd.spec
config.status: creating config.h
config.status: executing depfiles commands

```

- Since there is no error of package version, we can move forward with the installation.
- Run the make command:

```
make
```

```

root@Signing:/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0# make
make all-recursive
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0'
Making all in gnupg-pkcs11-scd
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0/gnupg-pkcs11-scd'
gcc -DHAVE_CONFIG_H -I. -I.. -I/usr/local/include -I/usr/local/include -pthread -g -O2 -MT scdaemon.o -MD -MP -MF .deps/scdaemon.Tpo -c -o scdaemon.o scdaemon.c
mv -f .deps/scdaemon.Tpo .deps/scdaemon.Po
gcc -DHAVE_CONFIG_H -I. -I.. -I/usr/local/include -I/usr/local/include -pthread -g -O2 -MT common.o -MD -MP -MF .deps/common.Tpo -c -o common.o common.c
mv -f .deps/common.Tpo .deps/common.Po
gcc -DHAVE_CONFIG_H -I. -I.. -I/usr/local/include -I/usr/local/include -pthread -g -O2 -MT strgetopt.o -MD -MP -MF .deps/strgetopt.Tpo -c -o strgetopt.o strgetopt.c

```

- Install the package:

```
sudo make install
```

```

root@Signing:/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0# sudo make install
Making install in gnupg-pkcs11-scd
make[1]: Entering directory '/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0/gnupg-pkcs11-scd'
make[2]: Entering directory '/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0/gnupg-pkcs11-scd'
  /usr/bin/mkdir -p '/usr/local/bin'
  /usr/bin/install -c gnupg-pkcs11-scd '/usr/local/bin'
  /usr/bin/mkdir -p '/usr/local/share/doc/gnupg-pkcs11-scd'
  /usr/bin/install -c -m 644 gnupg-pkcs11-scd.conf.example '/usr/local/share/doc/gnupg-pkcs11-scd'
  /usr/bin/mkdir -p '/usr/local/share/man/man1'
  /usr/bin/install -c -m 644 gnupg-pkcs11-scd.1 '/usr/local/share/man/man1'
make[2]: Leaving directory '/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0/gnupg-pkcs11-scd'

```

- Update the system's dynamic linker cache using the below command:

```
sudo ldconfig
```

```

root@Signing:/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0# sudo ldconfig
root@Signing:/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0#

```

- Check whether gnupg-pkcs11-scd has been installed correctly or not:

```
gnupg-pkcs11-scd --version
```

```

root@Signing:/home/aryan/gnupg-2.5.4/gnupg-pkcs11-scd-0.11.0# gnupg-pkcs11-scd --version
gnupg-pkcs11-scd 0.11.0

Copyright (c) 2006-2007 Zeljko Vrba <zvrba@globalnet.hr>
Copyright (c) 2006-2017 Alon Bar-Lev <alon.barlev@gmail.com>

This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

```

7. Create the GnuPG Directory and Environment Variables

GnuPG keeps its configuration and sockets in a home directory. Because this workflow runs as root, that directory is /root/.gnupg.

Steps:

Step 1: Create the GnuPG Directory

- Check whether the /root/.gnupg directory exists or not in your system:

```
ls -ld /root/.gnupg
```

```

root@Signing:/home/aryan# ls -ld /root/.gnupg
ls: cannot access '/root/.gnupg': No such file or directory
root@Signing:/home/aryan# █

```

- If not, then use the below command to make the directory:

```
mkdir /root/.gnupg
```

```
root@Signing:/home/aryan# mkdir /root/.gnupg
root@Signing:/home/aryan# ls -ld /root/.gnupg
drwxr-xr-x 2 root root 4096 Mar 18 06:49 /root/.gnupg
root@Signing:/home/aryan#
```

Step 2: Add the GPG Environment Variables

- Add the environment variables for the GPG Agent and the gnupg-pkcs11 socket directory.
- Open the bashrc file using the below command:

```
nano ~/.bashrc
```

```
root@Signing:/home/aryan# nano ~/.bashrc
root@Signing:/home/aryan#
```

- Paste the below commands as per the file locations in your system, at the end of the file, like:

```
export GPG_AGENT_INFO=/root/.gnupg
export GNUPG_PKCS11_SOCKETDIR=/root/.gnupg/
```

```
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#    . /etc/bash_completion
#fi
export EC_INI_FILE_PATH=/home/aryan/pkcs11-client/ec_pkcs11client.ini
export GPG_AGENT_INFO=/root/.gnupg
export GNUPG_PKCS11_SOCKETDIR=/root/.gnupg/
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

- Press Ctrl+X, then enter Y, and then press Enter to save.
- Reload the bashrc file:

```
source ~/.bashrc
```

```
root@Signing:/home/aryan# source ~/.bashrc
root@Signing:/home/aryan#
```

- To check these variables, run:

```
echo $GPG_AGENT_INFO
echo $GNUPG_PKCS11_SOCKETDIR
```

```
root@Signing:/home/aryan# echo $GPG_AGENT_INFO
/root/.gnupg
root@Signing:/home/aryan# echo $GNUPG_PKCS11_SOCKETDIR
/root/.gnupg/
root@Signing:/home/aryan#
```

- If you cannot see the file location from the echo commands, open a new terminal and try again.

8. Create the Configuration Files

Five configuration files are created in the `/root/.gnupg` directory. Together they tell GPG to use an agent, tell that agent to use the PKCS#11 smartcard daemon, and tell the daemon which wrapper library to load.

NOTE: Each directive below belongs on its own line in the file. On the published documentation page these appear run together on a single line.

Steps:

Step 1: Create `gnupg-pkcs11-scd.conf`

- Create a `gnupg-pkcs11-scd.conf` file in the `/root/.gnupg` directory.
- Run the following command to create or edit the file:

```
nano /root/.gnupg/gnupg-pkcs11-scd.conf
```

```
root@Signing:/home/aryan# nano /root/.gnupg/gnupg-pkcs11-scd.conf
root@Signing:/home/aryan#
```

- Modify and configure the file as per the below content and your system's file locations:

```
verbose
debug-all
providers ec
provider-ec-library /home/aryan/pkcs11-client/ec_pkcs11client.so
```

```
GNU nano 7.2 /root/.gnupg/gnupg-pkcs11-scd.conf *
verbose
debug-all
providers ec
provider-ec-library /home/aryan/pkcs11-client/ec_pkcs11client.so
█
```

- Press Ctrl+X, then enter Y, and then press Enter to save.

Step 2: Create gpg-agent.conf

- Create a gpg-agent.conf file in the /root/.gnupg directory.
- Run the following command to create or edit the file:

```
nano /root/.gnupg/gpg-agent.conf
```

```
root@Signing:/home/aryan# nano /root/.gnupg/gpg-agent.conf
root@Signing:/home/aryan# █
```

- Modify and configure the file as per the below content and your system's file locations:

```
verbose
debug-all
log-file /tmp/gpg-agent.log
scdaemon-program /usr/local/bin/gnupg-pkcs11-scd
pinentry-program /usr/bin/pinentry
```

```
GNU nano 7.2 /root/.gnupg/gpg-agent.conf *
verbose
debug-all
log-file /tmp/gpg-agent.log
scdaemon-program /usr/local/bin/gnupg-pkcs11-scd
pinentry-program /usr/bin/pinentry
```

- Press Ctrl+X, then enter Y, and then press Enter to save.

Step 3: Create gpg.conf

- Create a gpg.conf file in the /root/.gnupg directory.
- Run the following command to create or edit the file:

```
nano /root/.gnupg/gpg.conf
```

```
root@Signing:/home/aryan# nano /root/.gnupg/gpg.conf
root@Signing:/home/aryan#
```

- Modify and configure the file as per the below content and your system's file locations:

```
use-agent
#log-file /tmp/gpg.log
```

NOTE: You can uncomment the second line, by removing the # symbol, to generate a gpg.log file.

```
GNU nano 7.2 /root/.gnupg/gpg.conf
use-agent
#log-file /tmp/gpg.log
```

- Press Ctrl+X, then enter Y, and then press Enter to save.

Step 4: Create pkcs11-scd.conf

- Create a pkcs11-scd.conf file in the /root/.gnupg directory.
- Run the following command to create or edit the file:

```
nano /root/.gnupg/pkcs11-scd.conf
```

```
root@Signing:/home/aryan# nano /root/.gnupg/pkcs11-scd.conf
root@Signing:/home/aryan#
```

- Modify and configure the file as per the below content and your system's file locations:

```
module /home/aryan/pkcs11-client/ec_pkcs11client.so
```

```
GNU nano 7.2 /root/.gnupg/pkcs11-scd.conf *
module /home/aryan/pkcs11-client/ec_pkcs11client.so
```

- Press Ctrl+X, then enter Y, and then press Enter to save.

Step 5: Create scdaemon.conf

- Create a scdaemon.conf file in the /root/.gnupg directory.
- Run the following command to create or edit the file:

```
nano /root/.gnupg/scdaemon.conf
```

```
root@Signing:/home/aryan# nano /root/.gnupg/scdaemon.conf
root@Signing:/home/aryan#
```

- Add the below directive to this file:

```
disable-ccid
```

```
GNU nano 7.2 /root/.gnupg/scdaemon.conf *
disable-ccid
```

- Press Ctrl+X, then enter Y, and then press Enter to save.

9. Start the GPG Agent

Only one gpg-agent may run at a time. Any agent started before the configuration files existed will not have picked them up, so it must be stopped and replaced.

Steps:

Step 1: Check for a Running Agent

- To check if gpg-agent is running on your ubuntu machine:

```
ps -aef | grep gpg
```

```
root@Signing:/home/aryan# ps -aef | grep gpg
root      134028      1    0 06:07 ?        00:00:00 gpg-agent --homedir /var/lib/fwupd/gnupg --daemon
root      136706     2739    0 06:53 ?        00:00:00 gpg-agent --verbose --debug-level advanced --daemon
root      136730     4354    0 06:54 pts/1    00:00:00 grep --color=auto gpg
root@Signing:/home/aryan#
```

- If any agent is already running, you will need to kill that instance and start a new one.

Step 2: Stop Any Existing Agent

- To kill an instance of the agent, use the following command with the changes to the PID number, which is the number after the “root” user in the above image, as per your machine.

```
kill -9 134028
```

```
root@Signing:/home/aryan# kill -9 134028
root@Signing:/home/aryan# ps -aef | grep gpg
root      136706     2739    0 19:08 ?        00:00:00 gpg-agent --verbose --debug-level advanced --daemon
root      141385     4354    0 20:45 pts/1    00:00:00 grep --color=auto gpg
```

- Repeat for any other agent process listed:

```
kill -9 136706
```

```

root@Signing:/home/aryan# kill -9 136706
root@Signing:/home/aryan# ps -aef | grep gpg
root      141561    4354  0 20:53 pts/1    00:00:00 grep --color=auto gpg

```

NOTE: The PIDs above are from the example machine. Use the values shown in your own ps output.

Step 3: Start a New Agent

- Start a new gpg agent:

```
sudo gpg-agent --verbose --debug-level advanced --daemon
```

```

root@Signing:/home/aryan# sudo gpg-agent --verbose --debug-level advanced --daemon
gpg-agent[136705]: enabled debug flags: ipc
gpg-agent[136705]: directory '/root/.gnupg/private-keys-v1.d' created
root@Signing:/home/aryan# gpg-agent --verbose --debug-level advanced --daemon
gpg-agent[136711]: enabled debug flags: ipc
gpg-agent: a gpg-agent is already running - not starting a new one
gpg-agent: secmem usage: 0/32768 bytes in 0 blocks

```

- Check again that only one gpg-agent is running on your ubuntu machine:

```
ps -aef | grep gpg
```

```

root@Signing:/home/aryan/pkcs11-client# ps -aef | grep gpg
root      141564    2739  0 20:53 ?        00:00:00 gpg-agent --verbose --debug-level advanced --daemon
root      142391    4354  0 21:29 pts/1    00:00:00 grep --color=auto gpg

```

10. Retrieve and Import the Keys into GPG2

The keys themselves stay in the HSM. What is imported into GPG2 is a reference to the key on the token, which is why the key generation dialog is told that the key already exists on a card.

Steps:

Step 1: Reload the Agent

- Reload the agent:

```

gpg-connect-agent <<EOF
RELOADAGENT
EOF

```

```
root@Signing:/home/aryan# gpg-connect-agent << EOF
RELOADAGENT
EOF
OK
root@Signing:/home/aryan#
```

NOTE: This is a shell here-document: type the first line, press Enter, type RELOADAGENT, press Enter, then type EOF and press Enter. The same applies to the SCD LEARN command below.

Step 2: Retrieve the Card Status

- Retrieve card status:

```
gpg --debug-all --card-status
```

```
root@Signing:/home/aryan# gpg --debug-all --card-status
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg: reading options from '/root/.gnupg/gpg.conf'
gpg: reading options from '[cmdline]'
Reader .....: [none]
Application ID ...: D2760001240111503131216DD75C1111
Application type ..: OpenPGP
Version .....: 11.50
Manufacturer .....: ?
Serial number ....: 216DD75C
Name of cardholder: [not set]
Language prefs ...: [not set]
Salutation .....:
URL of public key : [not set]
Login data .....: [not set]
Signature PIN ....: forced
Key attributes ...: rsa48 rsa48 rsa48
Max. PIN lengths ..: 0 0 0
PIN retry counter : 0 0 0
Signature counter : 0
Signature key ....: [none]
Encryption key....: [none]
Authentication key: [none]
General key info..: [none]

Please try command "openpgp" if the listing does not look correct
```

Step 3: List the Key Friendly Names

- Get the key friendly names and certificates:

```
gpg-connect-agent <<EOF
```

```
SCD LEARN
EOF
```

```
root@Signing:/home/aryan# gpg-connect-agent << EOF
SCD LEARN
EOF

S SERIALNO D27600012401115031312160075C1111
S APPATYPE PKCS11
S KEY-FRIENDLY 7F1F389609D127BD27CD59B30285B6A778AC0E37 /DC=loc/DC=test/DC=encon/CN=Users/CN=Administrator on loadshared accelerator
S CERTINFO 101 nCIPHER\x20Corp\x2E\x20Ltd///loadshared\x20accelerator/185F5E5FAE11DAB878BA1879A7F383F8B4CF8E6017EE88004CB92F44D0ABA1DCEF2AD0FF1A88670527C1277DAB969AFA65917BDEB1D55617BC599E5457FDAB62
S KEYPAIRINFO 7F1F389609D127BD27CD59B30285B6A778AC0E37 nCIPHER\x20Corp\x2E\x20Ltd///loadshared\x20accelerator/185F5E5FAE11DAB878BA1879A7F383F8B4CF8E6017EE88004CB92F44D0ABA1DCEF2AD0FF1A88670527C1277DAB969AFA65917BDEB1D55617BC599E5457FDAB62
S KEY-FRIENDLY 7C914E99C2B17936483BC0E5B7A3151CCDCD28B7 /C=US/ST=Dallas/L=Texas/O=Encryption Consulting LLC/OU=CodeSign/CN=test1 on loadshared accelerator
S CERTINFO 101 nCIPHER\x20Corp\x2E\x20Ltd///loadshared\x20accelerator/BD29E360D19B50B0C62B65CAD3ED0FA55B5CEBE2B2AEAD0ED020679E5FAB4D51918B85C2F8459BFF31B03198F1DB2A9D443BC4E49AC4E7B85CBB678A290C5344
S KEYPAIRINFO 7C914E99C2B17936483BC0E5B7A3151CCDCD28B7 nCIPHER\x20Corp\x2E\x20Ltd///loadshared\x20accelerator/BD29E360D19B50B0C62B65CAD3ED0FA55B5CEBE2B2AEAD0ED020679E5FAB4D51918B85C2F8459BFF31B03198F1DB2A9D443BC4E49AC4E7B85CBB678A290C5344
OK
```

- Copy the required KEY FRIENDLY Id. You can compare the respective CKA_ID of the key you want to use for signing from the CodeSign Secure portal and copy the KEY FRIENDLY Id.

```
root@Signing:/home/aryan# gpg-connect-agent << EOF
SCD LEARN
EOF

S SERIALNO D27600012401115031312160075C1111
S APPATYPE PKCS11
S KEY-FRIENDLY 7F1F389609D127BD27CD59B30285B6A778AC0E37 /DC=loc/DC=test/DC=encon/CN=Users/CN=Administrator on loadshared accelerator
S CERTINFO 101 nCIPHER\x20Corp\x2E\x20Ltd///loadshared\x20accelerator/185F5E5FAE11DAB878BA1879A7F383F8B4CF8E6017EE88004CB92F44D0ABA1DCEF2AD0FF1A88670527C1277DAB969AFA65917BDEB1D55617BC599E5457FDAB62
S KEYPAIRINFO 7F1F389609D127BD27CD59B30285B6A778AC0E37 nCIPHER\x20Corp\x2E\x20Ltd///loadshared\x20accelerator/185F5E5FAE11DAB878BA1879A7F383F8B4CF8E6017EE88004CB92F44D0ABA1DCEF2AD0FF1A88670527C1277DAB969AFA65917BDEB1D55617BC599E5457FDAB62
S KEY-FRIENDLY 7C914E99C2B17936483BC0E5B7A3151CCDCD28B7 /C=US/ST=Dallas/L=Texas/O=Encryption Consulting LLC/OU=CodeSign/CN=test1 on loadshared accelerator
S CERTINFO 101 nCIPHER\x20Corp\x2E\x20Ltd///loadshared\x20accelerator/BD29E360D19B50B0C62B65CAD3ED0FA55B5CEBE2B2AEAD0ED020679E5FAB4D51918B85C2F8459BFF31B03198F1DB2A9D443BC4E49AC4E7B85CBB678A290C5344
S KEYPAIRINFO 7C914E99C2B17936483BC0E5B7A3151CCDCD28B7 nCIPHER\x20Corp\x2E\x20Ltd///loadshared\x20accelerator/BD29E360D19B50B0C62B65CAD3ED0FA55B5CEBE2B2AEAD0ED020679E5FAB4D51918B85C2F8459BFF31B03198F1DB2A9D443BC4E49AC4E7B85CBB678A290C5344
OK
```

Step 4: Import the Key into GPG2

- Import the key into GPG 2:

```
gpg --expert --full-generate-key
```

```
root@Signing:/home/aryan# gpg --expert --full-generate-key

gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg (GnuPG) 2.5.4; Copyright (C) 2025 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.
```

- This command will ask you for various inputs to properly import and set the permissions for the key in your machine.
 - First, it will ask you to select a number from the list. You will need to enter 13 as the key was already created from the CodeSign Secure portal.

```
Please select what kind of key you want:
```

- (1) RSA and RSA
- (2) DSA and Elgamal
- (3) DSA (sign only)
- (4) RSA (sign only)
- (7) DSA (set your own capabilities)
- (8) RSA (set your own capabilities)
- (9) ECC (sign and encrypt) *default*
- (10) ECC (sign only)
- (11) ECC (set your own capabilities)
- (13) Existing key
- (14) Existing key from card
- (16) ECC and Kyber

```
Your selection? 13
```

- Then, you will need to enter the KEY FRIENDLY Id that you had taken from the earlier steps.

```
Your selection? 13
```

```
Enter the keygrip: 7C914E99C2B17936483BCE0587A3151CCDCD28B7
```

- It will then ask you to confirm the permissions for this key. Type “Q”.

```
Possible actions for this RSA key: Sign Certify Encrypt Authenticate  
Current allowed actions: Sign Certify Encrypt
```

- (S) Toggle the sign capability
- (E) Toggle the encrypt capability
- (A) Toggle the authenticate capability
- (Q) Finished

```
Your selection? Q
```

- Now, you will need to set the key expiry date in your system. The default selection is “key does not expire”.

```

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

```

- Finally, it will prompt you to enter the key identification details for easier system access. Provide the name and email address for that key and confirm your settings by entering “O”.

```

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

Real name: Test1
Email address: aryan@encryptionconsulting.com
Comment:
You selected this USER-ID:
    "Test1 <aryan@encryptionconsulting.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O

```

- Your key will successfully get imported into GPG2 and now can be used for signing.

```

public and secret key created and signed.

pub   rsa3072 2025-03-18 [SCE]
      8A361E0C157B120C20595E738311968FA5629A34
uid           Test1 <aryan@encryptionconsulting.com>

```

NOTE: Now you have GPG2 set up and installed on your machine. Continue with Section 11 for GPG2 signing, Section 12 for Debian signing, or Section 13 for RPM signing.

11. GPG2 Signing and Verification

With the key imported, signing is a single gpg command that produces a detached .gpg signature file alongside the input.

Steps:

Step 1: Find the Public Key ID

- List all the keys available for GPG2 signing:

```
gpg --list-keys
```

```
root@Signing:/home/aryan# gpg --list-keys
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
/root/.gnupg/pubring.kbx
-----
pub   rsa3072 2025-03-18 [SCE]
      8A361E0C157B120C20595E738311968FA5629A34
uid           [ultimate] Test1 <aryan@encryptionconsulting.com>
```

- Copy the pub key ID for the key you want to use for signing.

```
root@Signing:/home/aryan# gpg --list-keys
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
/root/.gnupg/pubring.kbx
-----
pub   rsa3072 2025-03-18 [SCE]
      8A361E0C157B120C20595E738311968FA5629A34
uid           [ultimate] Test1 <aryan@encryptionconsulting.com>
```

Step 2: Sign the File

- Look for the file that you will sign using the key we just imported into GPG2.

```
root@Signing:/home/aryan/pkcs11-client# ls
ec_pkcs11client.ini  EC_PKCS11_CLIENT-LogConfig.xml  ec_pkcs11client.so  New_Auth_Cert.pfx  pkcs11properties.cfg  testfile
```

- Run the signing command:

```
gpg --sign \
  --default-key <PUB Key ID> \
  <File Path that needs to be signed>
```

- A sample command is provided below:

```
gpg --sign \
  --default-key 8A361E0C157B120C20595E738311968FA5629A34 \
```

```
/home/aryan/pkcs11-client/testfile
```

```
root@Signing:/home/aryan/pkcs11-client# gpg --sign --default-key 8A361E0C157B120C20595E738311968FA5629A34 /home/aryan/pkcs11-client/testfile
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg: using "8A361E0C157B120C20595E738311968FA5629A34" as default secret key for signing
root@Signing:/home/aryan/pkcs11-client# ls
ec_pkcs11client.ini  EC_PKCS11_CLIENT-LogConfig.xml  ec_pkcs11client.so  New_Auth_Cert.pfx  pkcs11properties.cfg  testfile  testfile.gpg
root@Signing:/home/aryan/pkcs11-client#
```

- It has successfully generated a signature file (testfile.gpg).

Step 3: Verify the Signature

- Run the verification command:

```
gpg --verify <Generated Signature Path>
```

- A sample command is provided below:

```
gpg --verify /home/aryan/pkcs11-client/testfile.gpg
```

```
root@Signing:/home/aryan/pkcs11-client# gpg --verify /home/aryan/pkcs11-client/testfile.gpg
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg: Signature made Wed 19 Mar 2025 12:16:45 AM IST
gpg:                using RSA key 8A361E0C157B120C20595E738311968FA5629A34
gpg: Good signature from "Test1 <aryan@encryptionconsulting.com>" [ultimate]
root@Signing:/home/aryan/pkcs11-client#
```

Step 4: Cross-Check in CodeSign Secure

- Open the CodeSign Secure portal and navigate to Reports > Signing Request Report, where every signing request performed through the wrapper is recorded for audit purposes.

12. Additional Steps for Debian Signing and Verification

Debian signing relies on GPG2 (GNU Privacy Guard) keys, so before proceeding you will need to have completed Sections 1 to 10. Unlike GPG2 signing, the key is not named on the command line — it is set as the default key in gpg.conf.

Steps:

Step 1: Find the Public Key ID

- List all the keys available for GPG2 signing:

```
gpg --list-keys
```

```

root@Signing:/home/aryan# gpg --list-keys
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
/root/.gnupg/pubring.kbx
-----
pub   rsa3072 2025-03-18 [SCE]
      8A361E0C157B120C20595E738311968FA5629A34
uid           [ultimate] Test1 <aryan@encryptionconsulting.com>

```

- Copy the pub key ID for the key you want to use for signing.

```

root@Signing:/home/aryan# gpg --list-keys
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
/root/.gnupg/pubring.kbx
-----
pub   rsa3072 2025-03-18 [SCE]
      8A361E0C157B120C20595E738311968FA5629A34
uid           [ultimate] Test1 <aryan@encryptionconsulting.com>

```

Step 2: Set the Default Key in gpg.conf

- Mention the pub key ID in the gpg.conf file that we created earlier.
- Open the gpg.conf using the below command:

```
nano /root/.gnupg/gpg.conf
```

```

root@Signing:/home/aryan/pkcs11-client# nano /root/.gnupg/gpg.conf
root@Signing:/home/aryan/pkcs11-client#

```

- This should be the content inside this file.

```

GNU nano 7.2 /root/.gnupg/gpg.conf
use-agent
#log-file /tmp/gpg.log

```

- Now, add the respective pub key ID as the default key at the end of the file, like:

```
default-key 8A361E0C157B120C20595E738311968FA5629A34
```

```
GNU nano 7.2 /root/.gnupg/gpg.conf *
use-agent
#log-file /tmp/gpg.log
default-key 8A361E0C157B120C20595E738311968FA5629A34
```

- Press Ctrl+X, then enter Y, and then press Enter to save.

Step 3: Install the dpkg-sig Utility

- Check whether the Debian package utility is available:

```
sudo apt-cache policy dpkg-sig
```

```
root@Signing:/home/aryan# sudo apt-cache policy dpkg-sig
N: Unable to locate package dpkg-sig
root@Signing:/home/aryan#
```

- If the above gives an error or returns an empty output, refer to the below steps to download and install the dpkg-sig package.

- Download the dpkg-sig package:

```
wget https://old-releases.ubuntu.com/ubuntu/pool/universe/d/\
dpkg-sig/dpkg-sig_0.13.1+nmu2ubuntu1_all.deb
```

```
root@Signing:/home/aryan# rm -f dpkg-sig_0.13.1+nmu4.tar.gz
root@Signing:/home/aryan# wget https://old-releases.ubuntu.com/ubuntu/pool/universe/d/dpkg-sig/dpkg-sig_0.13.1+nmu2ubuntu1_all.deb
--2025-03-19 04:18:35-- https://old-releases.ubuntu.com/ubuntu/pool/universe/d/dpkg-sig/dpkg-sig_0.13.1+nmu2ubuntu1_all.deb
Resolving old-releases.ubuntu.com (old-releases.ubuntu.com)... 185.125.190.40, 185.125.190.37, 91.189.91.123, ...
Connecting to old-releases.ubuntu.com (old-releases.ubuntu.com)|185.125.190.40|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33978 (33K) [application/x-debian-package]
Saving to: 'dpkg-sig_0.13.1+nmu2ubuntu1_all.deb'

dpkg-sig_0.13.1+nmu2ubuntu1_all.deb 100%[=====] 33.18K 184KB/s in 0.2s

2025-03-19 04:18:37 (184 KB/s) - 'dpkg-sig_0.13.1+nmu2ubuntu1_all.deb' saved [33978/33978]

root@Signing:/home/aryan# ls
Desktop Downloads ec_pkcs11_client.log gnupg-2.5.4.tar.bz2 Pictures Public Templates
Documents dpkg-sig_0.13.1+nmu2ubuntu1_all.deb gnupg-2.5.4 Music pkcs11-client snap Videos
```

- Install the package:

```
sudo apt install ./dpkg-sig_0.13.1+nmu2ubuntu1_all.deb
```

```

root@Signing:/home/aryan# sudo apt install ./dpkg-sig_0.13.1+nmu2ubuntu1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'dpkg-sig' instead of './dpkg-sig_0.13.1+nmu2ubuntu1_all.deb'
The following packages were automatically installed and are no longer required:
  libllvm17t64 python3-netifaces
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libconfig-file-perl
Suggested packages:
  ssh libterm-readkey-perl
The following NEW packages will be installed:
  dpkg-sig libconfig-file-perl
0 upgraded, 2 newly installed, 0 to remove and 13 not upgraded.

```

NOTE: Use the same filename you downloaded. The published documentation page downloads one filename and installs a different one, so the install step will fail if copied literally.

- Check whether dpkg-sig has installed correctly in your system:

```
dpkg-sig
```

```

root@Signing:/home/aryan# dpkg-sig
Short usage:
If not stated otherwise, all commands have one or more .deb or .changes
files as argument. In the case of .changes files, dpkg-sig will work on
all .deb files named in there.

-s,--sign <SIGNING_NAME>      Sign files
-c,--verify                    Verify signatures on files
-l,--list                      List signatures on files

--get-hashes                  Get hashes file for files
--sign-hashes <HASHES_FILE>   Sign hashes file
--write-hashes <HASHES_FILE> Write sigs from signed hashes file

Options:
-m,--e <MAINTAINER_NAME>     Specify maintainer name to use when signing
-k <KEYID>                    Specify keyid to use when signing
-v,--verbose                  Makes dpkg-sig more verbose
--also-v2-sig                 Verify sigs from dpkg-sig 0.2 and earlier
--also-v3-sig                 Verify sigs from dpkg-sig 0.3-0.10
-a,--sign-changes <no|auto|yes|full|force_full>
    Tells whether also sign the changes and dsc-files. The default is auto.

-g,--gpgoptions <GPG_OPTIONS> DANGEROUS: Specify custom gpg options.
-p,--cache-passphrase         INSECURE: Caches gpg passphrase in dpkg-sig
-f,--passphrase-file <FILE>   INSECURE: Let gpg use passphrase from <FILE>
root@Signing:/home/aryan#

```

Step 4: Sign the Debian Package

- Locate the Debian file, which you need to sign using the key that you had set in the gpg.conf file as the default key.

```
root@Signing:/home/aryan/pkcs11-client# ls
ec_pkcs11client.ini  EC_PKCS11_CLIENT-LogConfig.xml  ec_pkcs11client.so  New_Auth_Cert.pfx  pkcs11properties.cfg  testfile  testfile.gpg  testsample.deb
root@Signing:/home/aryan/pkcs11-client#
```

- Run the signing command:

```
dpkg-sig --sign builder <File Path you want to sign>
```

- A sample command is provided below:

```
dpkg-sig --sign builder /home/aryan/pkcs11-client/testsample.deb
```

```
root@Signing:/home/aryan/pkcs11-client# dpkg-sig --sign builder /home/aryan/pkcs11-client/testsample.deb
Processing /home/aryan/pkcs11-client/testsample.deb...
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg: using "8A361E0C157B120C20595E738311968FA5629A34" as default secret key for signing
Signed deb /home/aryan/pkcs11-client/testsample.deb
```

Step 5: Verify the Signature

- Run the verification command:

```
dpkg-sig --verify <File Path which got signed>
```

- A sample command is provided below:

```
dpkg-sig --verify /home/aryan/pkcs11-client/testsample.deb
```

```
root@Signing:/home/aryan/pkcs11-client# dpkg-sig --verify /home/aryan/pkcs11-client/testsample.deb
Processing /home/aryan/pkcs11-client/testsample.deb...
GOODSIG _gpgbuilder 8A361E0C157B120C20595E738311968FA5629A34 1742339342
root@Signing:/home/aryan/pkcs11-client#
```

13. Additional Steps for RPM Signing and Verification

RPM signing also delegates to GPG2, so Sections 1 to 10 must be complete. RPM needs the public key exported and imported into its own keyring, and a .rpmmacros file that names the key to use.

Steps:

Step 1: Install RPM and gnupg2

- Install the rpm package on your machine:

```
sudo apt install rpm
```

```
root@Signing:/home/aryan# sudo apt install rpm
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libllvm17t64 python3-netifaces
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  debugedit libfsverity0 liblua5.3-0 librpm9t64 librpmbuild9t64 librpmio9t64 librpmjsign9t64 rpm-common rpm2cpio
Suggested packages:
  alien elfutils rpmlint rpm-i18n
The following NEW packages will be installed:
  debugedit libfsverity0 liblua5.3-0 librpm9t64 librpmbuild9t64 librpmio9t64 librpmjsign9t64 rpm rpm-common rpm2cpio
0 upgraded, 10 newly installed, 0 to remove and 13 not upgraded.
```

- Check whether rpm has been installed correctly:

```
rpm --version
```

```
root@Signing:/home/aryan# rpm --version
RPM version 4.18.2
root@Signing:/home/aryan#
```

- Install the gnupg2 package:

```
sudo apt install gnupg2
```

```
root@Signing:/home/aryan# sudo apt install gnupg2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libllvm17t64 python3-netifaces
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  gnupg2
0 upgraded, 1 newly installed, 0 to remove and 13 not upgraded.
Need to get 4,748 B of archives.
```

- Check whether gpg2 has installed correctly:

```
gpg2 --version
```

```

root@Signing:/home/aryan# gpg2 --version
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
gpg (GnuPG) 2.4.4
libgcrypt 1.11.0
Copyright (C) 2024 g10 Code GmbH
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /root/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2

```

Step 2: Export the Public Key

- List all the keys available for GPG2 signing:

```
gpg --list-keys
```

```

root@Signing:/home/aryan# gpg --list-keys
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
/root/.gnupg/pubring.kbx
-----
pub   rsa3072 2025-03-18 [SCE]
      8A361E0C157B120C20595E738311968FA5629A34
uid           [ultimate] Test1 <aryan@encryptionconsulting.com>

```

- Copy the set pub key uid name for the key you want to use for signing.

```

root@Signing:/home/aryan# gpg --list-keys
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
/root/.gnupg/pubring.kbx
-----
pub   rsa3072 2025-03-18 [SCE]
      8A361E0C157B120C20595E738311968FA5629A34
uid           [ultimate] Test1 <aryan@encryptionconsulting.com>

```

- Export the respective key to a file with the pub key uid name as per your setup, like:

```
gpg --export -a "Test1" > ~/RPM-GPG-KEY
```

```
root@Signing:/home/aryan# gpg --export -a "Test1" > ~/RPM-GPG-KEY
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
```

NOTE: The uid name used here, Test1, is the name you set when importing the key in Section 10. It must match exactly, including capitalisation.

Step 3: Import the Key into RPM

- Import this file into RPM:

```
rpm --import ~/RPM-GPG-KEY
```

```
root@Signing:/home/aryan# rpm --import ~/RPM-GPG-KEY
root@Signing:/home/aryan#
```

Step 4: Create the .rpmmacros File

- Create a .rpmmacros file, which will tell the RPM package which GPG key to use for signing.
- Run the following command to create or edit the file:

```
nano ~/.rpmmacros
```

```
root@Signing:/home/aryan# nano ~/.rpmmacros
root@Signing:/home/aryan#
```

- Modify the ~/.rpmmacros with file locations as per the paths set in your ubuntu machine and the required pub key uid name.

```
%_gpg_name Test1
%_signature gpg
%_gpg_path /root/.gnupg/
%_gpgbin /usr/bin/gpg2
%__gpg_check_password_cmd /bin/true
```

```
GNU nano 7.2 /root/.rpmmacros *
%_gpg_name Test1
%_signature gpg
%_gpg_path /root/.gnupg/
%_gpgbin /usr/bin/gpg2
%__gpg_check_password_cmd /bin/true
```

- Press Ctrl+X, then enter Y, and then press Enter to save.

Step 5: Sign the RPM Package

- Locate the RPM file that you need to sign using the key we just imported into RPM.

```
root@Signing:/home/aryan/pkcs11-client# ls
ec_pkcs11client.ini      ec_pkcs11client.so  pkcs11properties.cfg  testfile      testsample.deb
EC_PKCS11_CLIENT-LogConfig.xml  New_Auth_Cert.pfx  RPMsample.rpm         testfile.gpg
root@Signing:/home/aryan/pkcs11-client#
```

- Run the signing command:

```
rpm --verbose --addsign <File Path to be signed>
```

- A sample command is provided below:

```
rpm --verbose --addsign /home/aryan/pkcs11-client/RPMsample.rpm
```

```
root@Signing:/home/aryan/pkcs11-client# rpm --verbose --addsign /home/aryan/pkcs11-client/RPMsample.rpm
/home/aryan/pkcs11-client/RPMsample.rpm:
gpg: WARNING: unsafe permissions on homedir '/root/.gnupg'
root@Signing:/home/aryan/pkcs11-client#
```

Step 6: Verify the Signature

- Run the verification command:

```
rpm --verbose --checksig <File Path which was signed>
```

- A sample command is provided below:

```
rpm --verbose --checksig /home/aryan/pkcs11-client/RPMsample.rpm
```

Step 7: Cross-Check in CodeSign Secure

- Open the CodeSign Secure portal and navigate to Reports > Signing Request Report, where every signing request performed through the wrapper is recorded for audit purposes.