Roll No 06

Name: Prasad Sunil Arote

Date: 10/09/2023

Lab Assignment 13

AIM: Explore the GPG tool of Linux to implement Email Security.

LO6: Demonstrate Network Security system using Open Source tools.

THEORY:

A "private keyring" and a "public keyring" are concepts related to cryptographic key management in applications like GPG (GNU Privacy Guard), which is used for secure communication, digital signatures, and encryption. These terms refer to collections of cryptographic keys.

Private Keyring: A private keyring is a file or database that stores private cryptographic keys. Private keys are used for operations like signing messages or decrypting data. These keys should be kept confidential because anyone with access to a private key can use it to impersonate the owner or access encrypted information.

Public Keyring: A public keyring is a file or database that stores public cryptographic keys. Public keys are shared with others and are used for operations like verifying digital signatures or encrypting data that can only be decrypted by the corresponding private key. Public keys are meant to be distributed openly.

Commands for key generation, export, and import of keys, as well as for signing and encrypting a message in GPG.

Key Generation:

To generate a new GPG key pair (public and private keys), use the following command: gpg

--gen-key

This command will prompt you to enter details such as your name, email address, and passphrase for the private key. It will generate a key pair and add it to your keyring.

Exporting and Importing Keys:

To export your public key to a file (e.g., my_public_key.asc), use:

gpg --export -a "Your Name" > my_public_key.asc To

import a public key from a file, use:

gpg --import < my_public_key.asc Signing

a Message:

To sign a message using your private key, use:

gpg --detach-sign -a my_message.txt

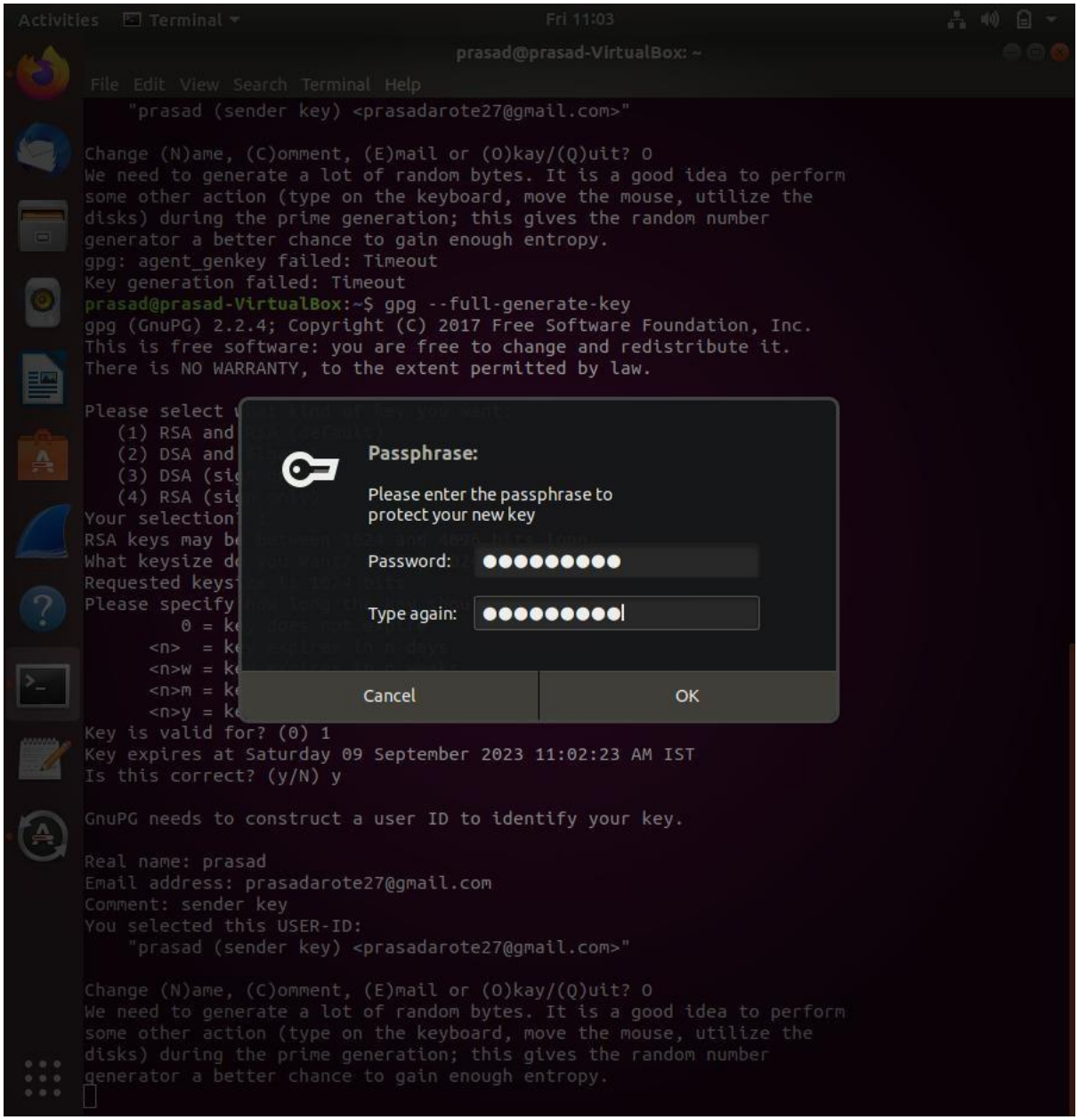
This will create a detached signature file (e.g., my_message.txt.asc) for your message.

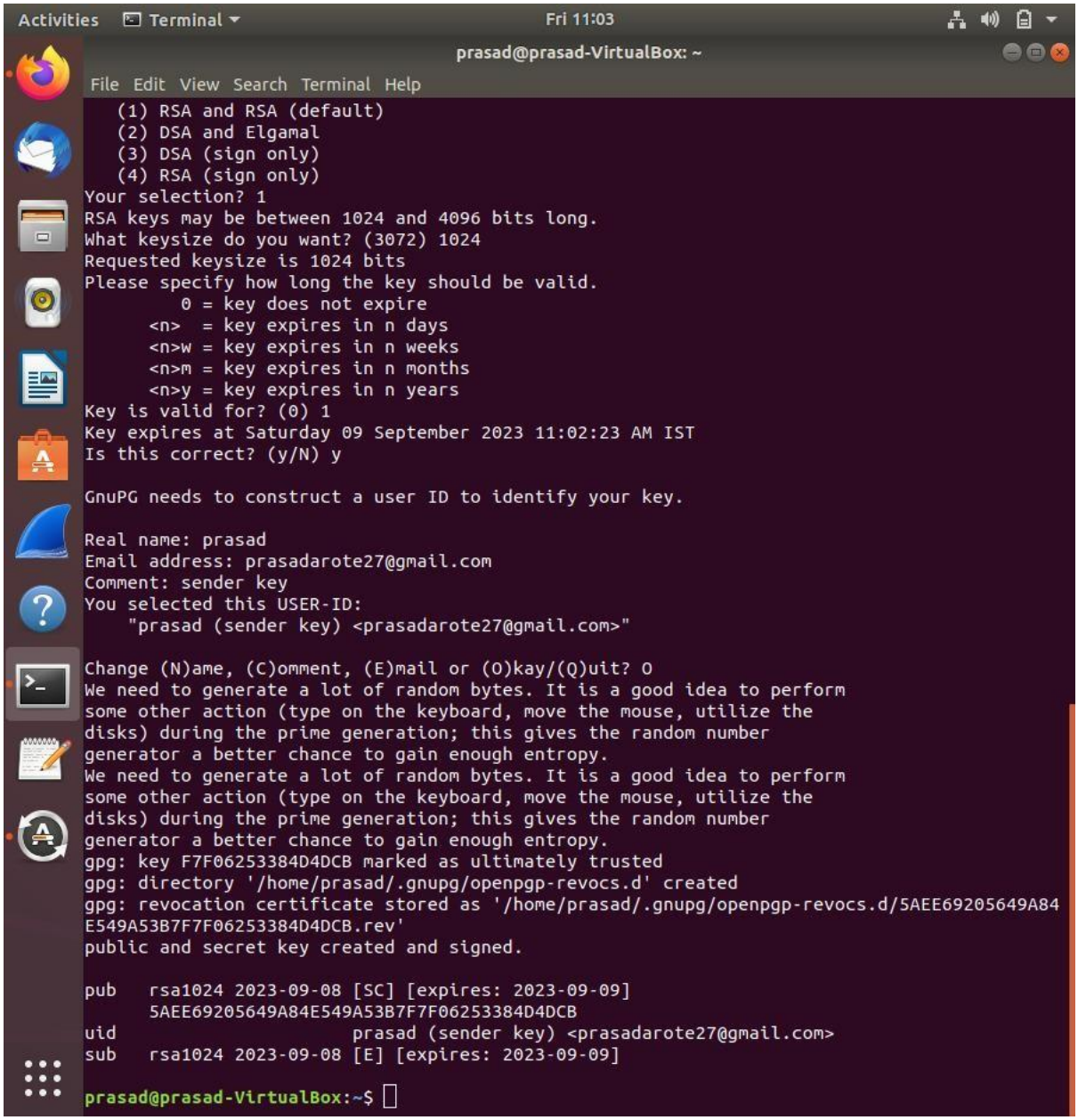
Encrypting a Message:

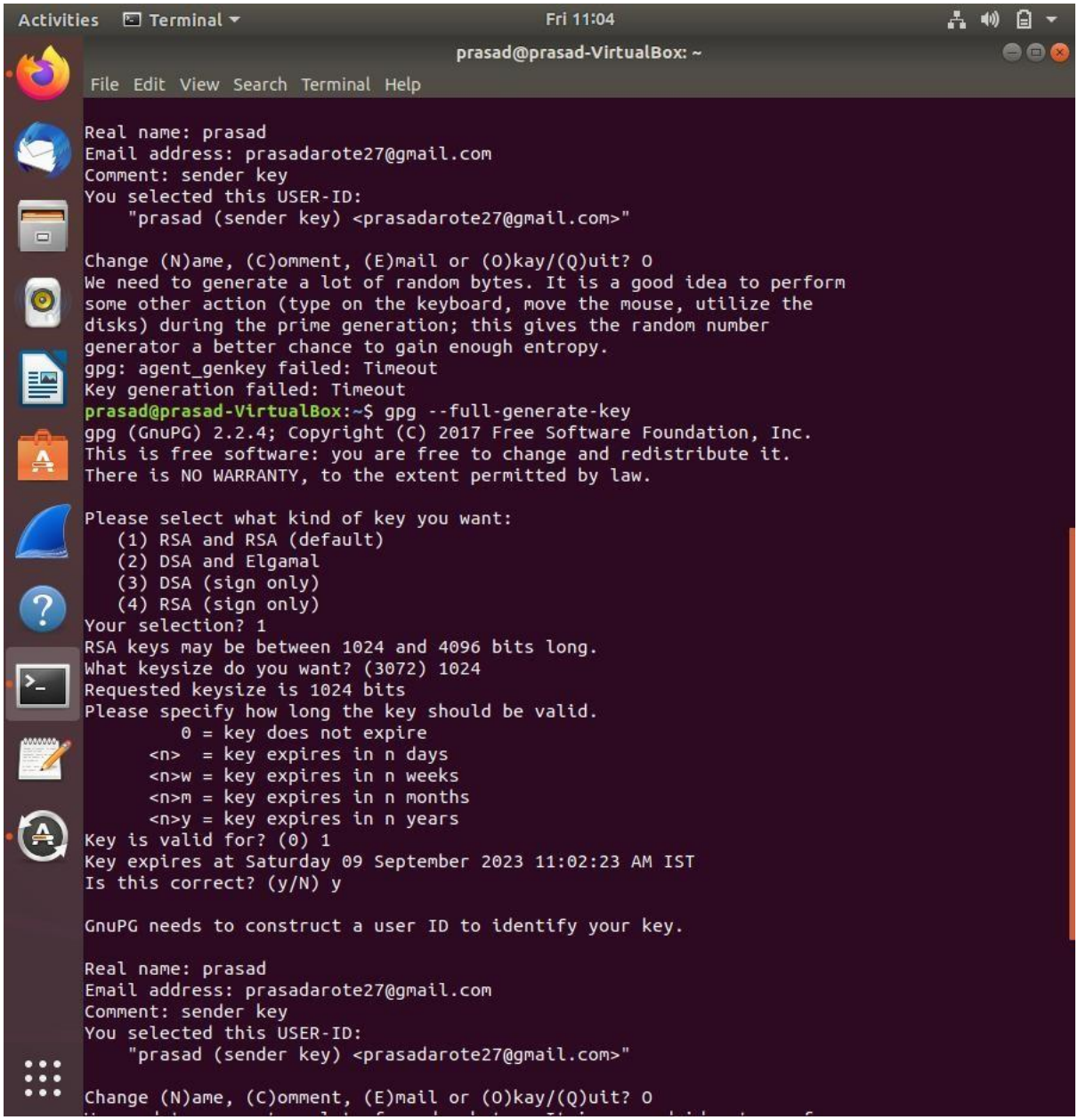
To encrypt a message for someone else using their public key, use:

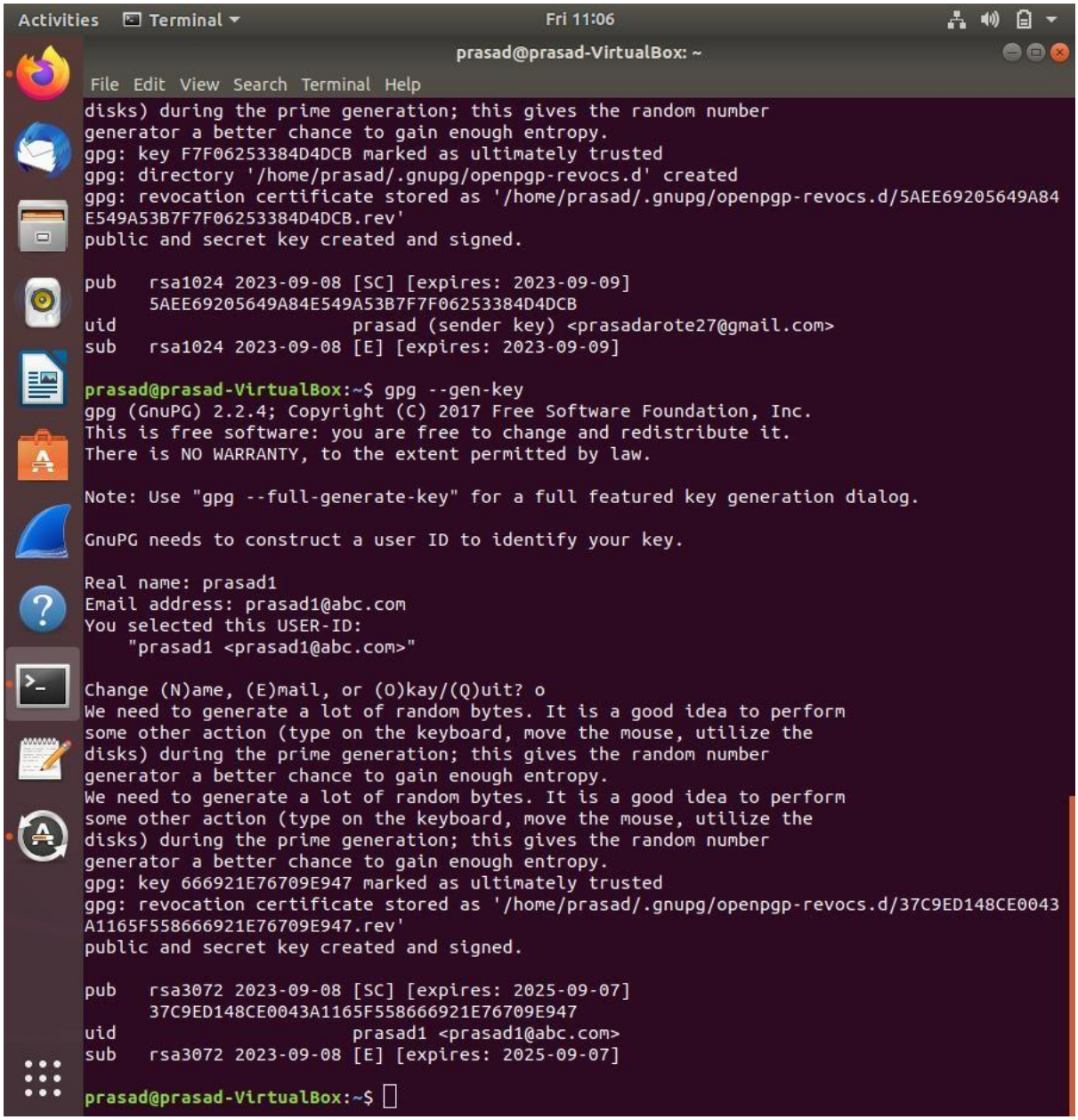
gpg --encrypt -a -r "Recipient's Name" my_message.txt

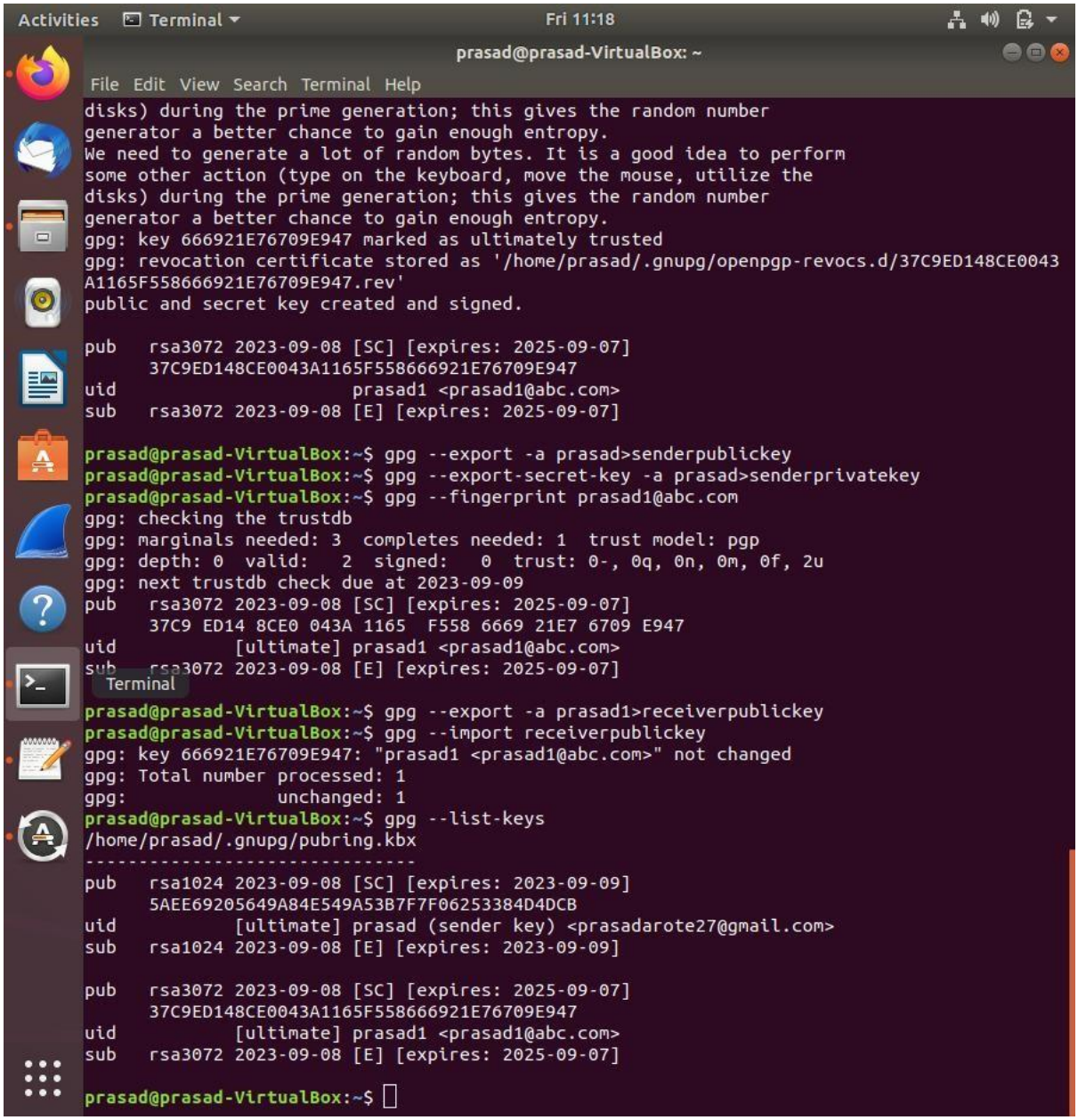
This will create an encrypted file (e.g., my_message.txt.asc) that can only be decrypted by the recipient's private key.

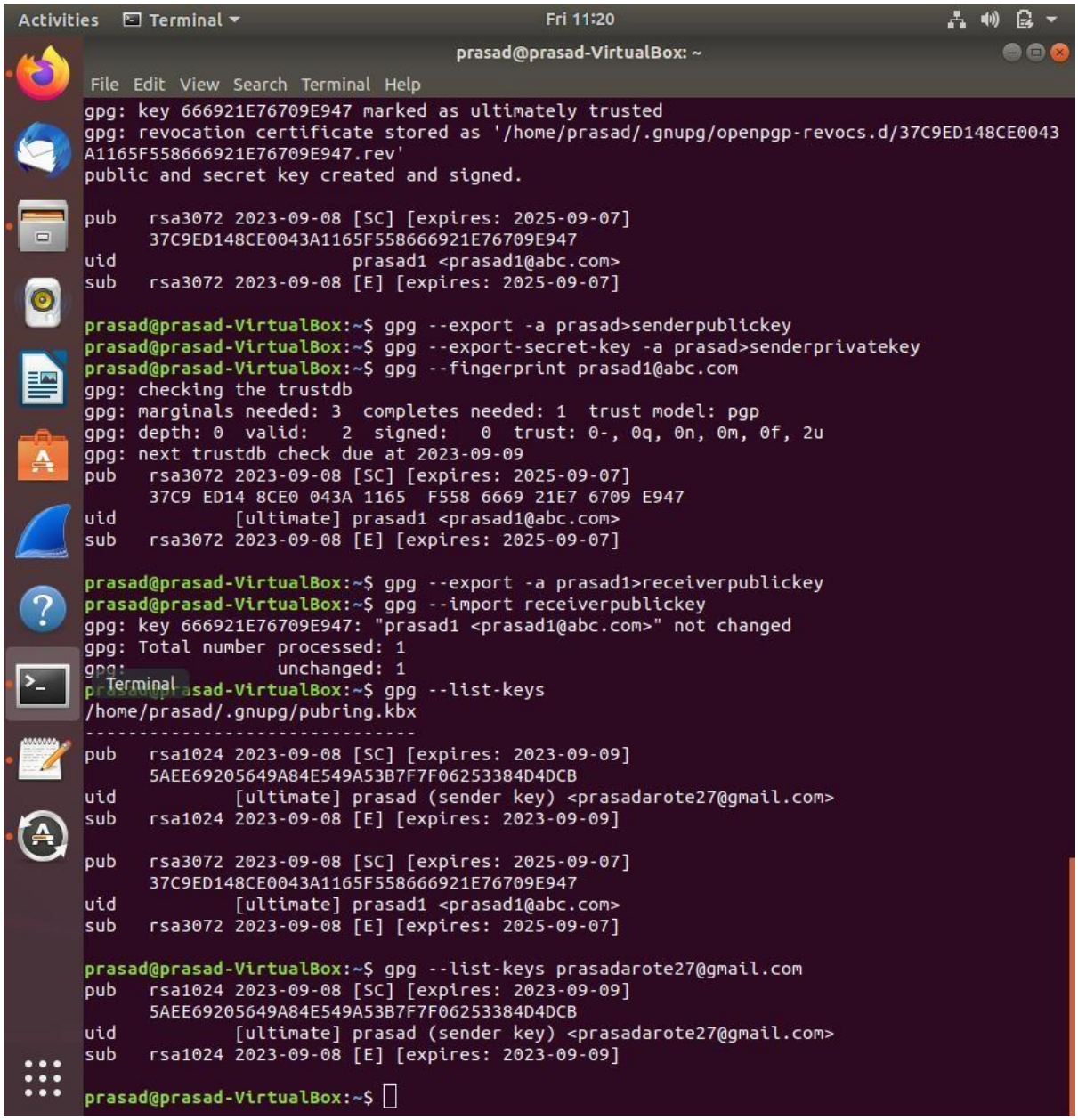

Activities
Terminal
Fri 11:03
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
"prasad (sender key) <prasadarote27@gmail.com>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: agent_genkey failed: Timeout
Key generation failed: Timeout
prasad@prasad-VirtualBox:~$ gpg --full-generate-key
gpg (GnuPG) 2.2.4; Copyright (C) 2017 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Passphrase:
Please enter the passphrase to protect your new key
Password:
Type again:
Cancel
OK
Key is valid for? (0) 1
Key expires at Saturday 09 September 2023 11:02:23 AM IST
Is this correct? (y/N) y
GnuPG needs to construct a user ID to identify your key.
Real name: prasad
Email address: prasadarote27@gmail.com
Comment: sender key
You selected this USER-ID:
"prasad (sender key) <prasadarote27@gmail.com>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.

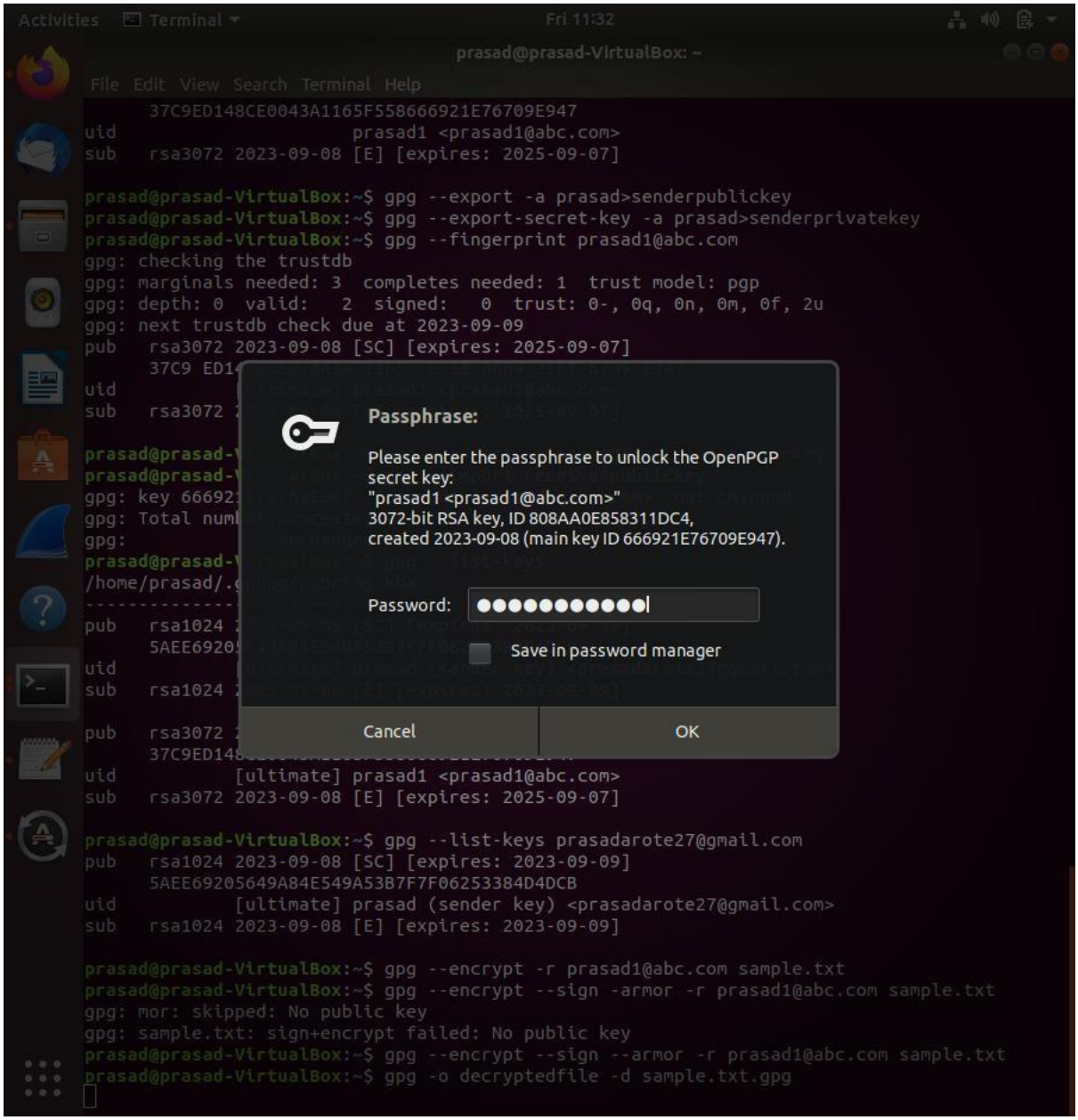

Activities
Terminal
Fri 11:03
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 1024
Requested keysize is 1024 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) 1
Key expires at Saturday 09 September 2023 11:02:23 AM IST
Is this correct? (y/N) y
GnuPG needs to construct a user ID to identify your key.
Real name: prasad
Email address: prasadarote27@gmail.com
Comment: sender key
You selected this USER-ID:
"prasad (sender key) <prasadarote27@gmail.com>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key F7F06253384D4DCB marked as ultimately trusted
gpg: directory '/home/prasad/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/prasad/.gnupg/openpgp-revocs.d/5AEE69205649A84
E549A53B7F7F06253384D4DCB.rev'
public and secret key created and signed.
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
5AEE69205649A84E549A53B7F7F06253384D4DCB
uid prasad (sender key) <prasadarote27@gmail.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]
prasad@prasad-VirtualBox:~$

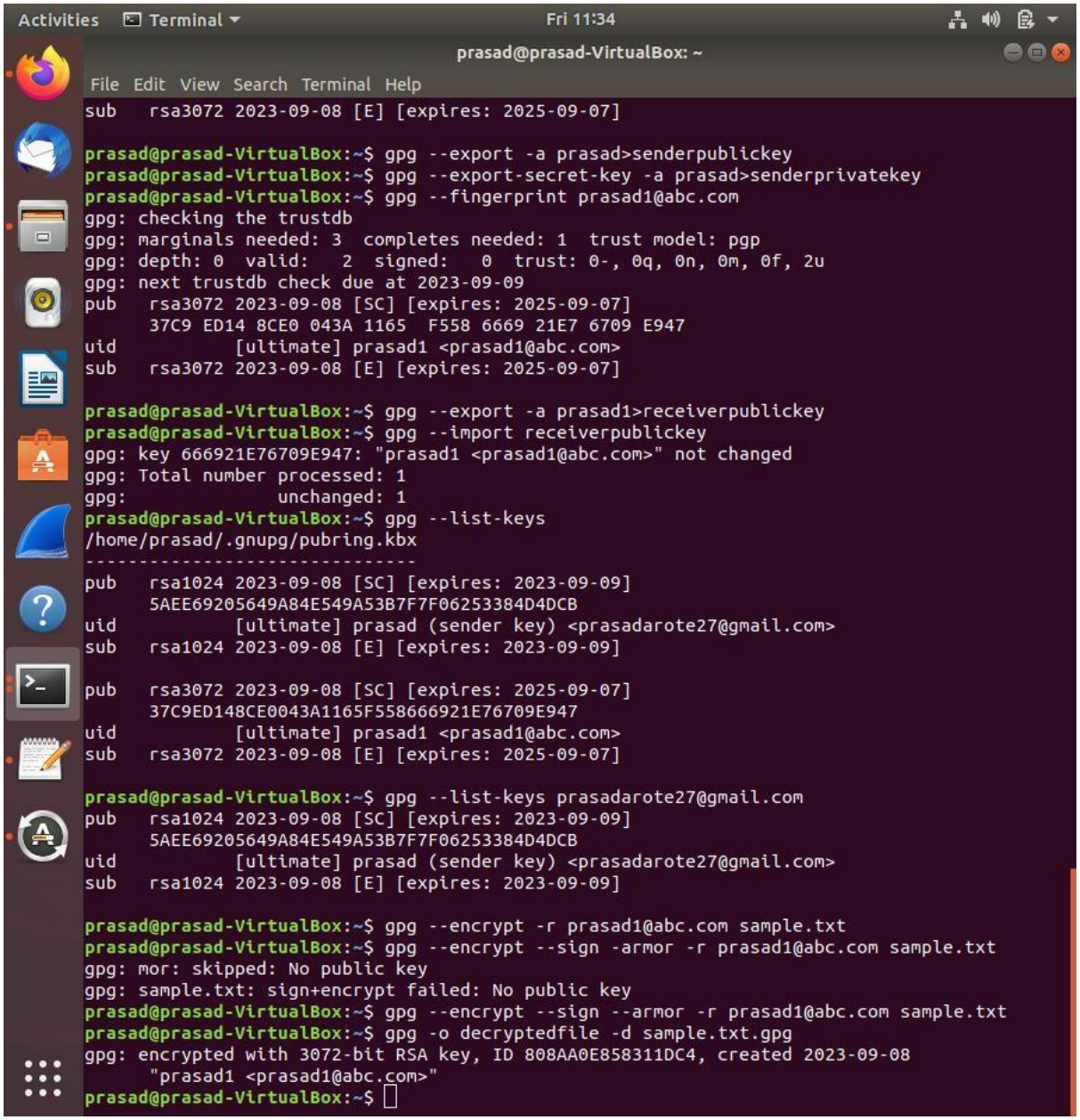

Activities
Terminal
Fri 11:04
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
Real name: prasad
Email address: prasadarote27@gmail.com
Comment: sender key
You selected this USER-ID:
"prasad (sender key) <prasadarote27@gmail.com>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: agent_genkey failed: Timeout
Key generation failed: Timeout
prasad@prasad-VirtualBox:~$ gpg --full-generate-key
gpg (GnuPG) 2.2.4; Copyright (C) 2017 Free Software Foundation, Inc.
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 (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 1024
Requested keysize is 1024 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) 1
Key expires at Saturday 09 September 2023 11:02:23 AM IST
Is this correct? (y/N) y
GnuPG needs to construct a user ID to identify your key.
Real name: prasad
Email address: prasadarote27@gmail.com
Comment: sender key
You selected this USER-ID:
"prasad (sender key) <prasadarote27@gmail.com>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O


Activities
Terminal
Fri 11:06
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key F7F06253384D4DCB marked as ultimately trusted
gpg: directory '/home/prasad/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/prasad/.gnupg/openpgp-revocs.d/5AEE69205649A84
E549A53B7F7F06253384D4DCB.rev'
public and secret key created and signed.
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
5AEE69205649A84E549A53B7F7F06253384D4DCB
uid prasad (sender key) <prasadarote27@gmail.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]
prasad@prasad-VirtualBox:~$ gpg --gen-key
gpg (GnuPG) 2.2.4; Copyright (C) 2017 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Note: Use "gpg --full-generate-key" for a full featured key generation dialog.
GnuPG needs to construct a user ID to identify your key.
Real name: prasad1
Email address: prasad1@abc.com
You selected this USER-ID:
"prasad1 <prasad1@abc.com>"
Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 666921E76709E947 marked as ultimately trusted
gpg: revocation certificate stored as '/home/prasad/.gnupg/openpgp-revocs.d/37C9ED148CE0043
A1165F558666921E76709E947.rev'
public and secret key created and signed.
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9ED148CE0043A1165F558666921E76709E947
uid prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$


Activities
Terminal
Fri 11:18
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 666921E76709E947 marked as ultimately trusted
gpg: revocation certificate stored as '/home/prasad/.gnupg/openpgp-revocs.d/37C9ED148CE0043
A1165F558666921E76709E947.rev'
public and secret key created and signed.
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9ED148CE0043A1165F558666921E76709E947
uid prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$ gpg --export -a prasad>senderpublickey
prasad@prasad-VirtualBox:~$ gpg --export-secret-key -a prasad>senderprivatekey
prasad@prasad-VirtualBox:~$ gpg --fingerprint prasad1@abc.com
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
gpg: next trustdb check due at 2023-09-09
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9 ED14 8CE0 043A 1165 F558 6669 21E7 6709 E947
uid [ultimate] prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
Terminal
prasad@prasad-VirtualBox:~$ gpg --export -a prasad1>receiverpublickey
prasad@prasad-VirtualBox:~$ gpg --import receiverpublickey
gpg: key 666921E76709E947: "prasad1 <prasad1@abc.com>" not changed
gpg: Total number processed: 1
gpg: unchanged: 1
prasad@prasad-VirtualBox:~$ gpg --list-keys
/home/prasad/.gnupg/pubring.kbx
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
5AEE69205649A84E549A53B7F7F06253384D4DCB
uid [ultimate] prasad (sender key) <prasadarote27@gmail.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9ED148CE0043A1165F558666921E76709E947
uid [ultimate] prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$


Activities
Terminal
Fri 11:20
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
gpg: key 666921E76709E947 marked as ultimately trusted
gpg: revocation certificate stored as '/home/prasad/.gnupg/openpgp-revocs.d/37C9ED148CE0043A1165F558666921E76709E947.rev'
public and secret key created and signed.
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9ED148CE0043A1165F558666921E76709E947
uid prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$ gpg --export -a prasad>senderpublickey
prasad@prasad-VirtualBox:~$ gpg --export-secret-key -a prasad>senderprivatekey
prasad@prasad-VirtualBox:~$ gpg --fingerprint prasad1@abc.com
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
gpg: next trustdb check due at 2023-09-09
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9 ED14 8CE0 043A 1165 F558 6669 21E7 6709 E947
uid [ultimate] prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$ gpg --export -a prasad1>receiverpublickey
prasad@prasad-VirtualBox:~$ gpg --import receiverpublickey
gpg: key 666921E76709E947: "prasad1 <prasad1@abc.com>" not changed
gpg: Total number processed: 1
gpg: unchanged: 1
Terminal
prasad@prasad-VirtualBox:~$ gpg --list-keys
/home/prasad/.gnupg/pubring.kbx
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
5AEE69205649A84E549A53B7F7F06253384D4DCB
uid [ultimate] prasad (sender key) <prasadarote27@gmail.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9ED148CE0043A1165F558666921E76709E947
uid [ultimate] prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$ gpg --list-keys prasadarote27@gmail.com
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
5AEE69205649A84E549A53B7F7F06253384D4DCB
uid [ultimate] prasad (sender key) <prasadarote27@gmail.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]
prasad@prasad-VirtualBox:~$


Activities
Terminal
Fri 11:32
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
37C9ED148CE0043A1165F558666921E76709E947
uid prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$ gpg --export -a prasad>senderpublickey
prasad@prasad-VirtualBox:~$ gpg --export-secret-key -a prasad>senderprivatekey
prasad@prasad-VirtualBox:~$ gpg --fingerprint prasad1@abc.com
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
gpg: next trustdb check due at 2023-09-09
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
Passphrase:
Please enter the passphrase to unlock the OpenPGP secret key:
"prasad1 <prasad1@abc.com>"
3072-bit RSA key, ID 808AA0E858311DC4,
created 2023-09-08 (main key ID 666921E76709E947).
Password:
Save in password manager
Cancel
OK
uid [ultimate] prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$ gpg --list-keys prasadarote27@gmail.com
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
5AEE69205649A84E549A53B7F7F06253384D4DCB
uid [ultimate] prasad (sender key) <prasadarote27@gmail.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]
prasad@prasad-VirtualBox:~$ gpg --encrypt -r prasad1@abc.com sample.txt
prasad@prasad-VirtualBox:~$ gpg --encrypt --sign -armor -r prasad1@abc.com sample.txt
gpg: mor: skipped: No public key
gpg: sample.txt: sign+encrypt failed: No public key
prasad@prasad-VirtualBox:~$ gpg --encrypt --sign --armor -r prasad1@abc.com sample.txt
prasad@prasad-VirtualBox:~$ gpg -o decryptedfile -d sample.txt.gpg



CONCLUSION:

We've explored the concepts of private keyrings and public keyrings in GPG. We've also provided commands for key generation, exporting and importing keys, signing messages, and encrypting messages using GPG. These commands are fundamental to using GPG for secure communication and data protection.