

Report: Directory Encryption using eCryptfs

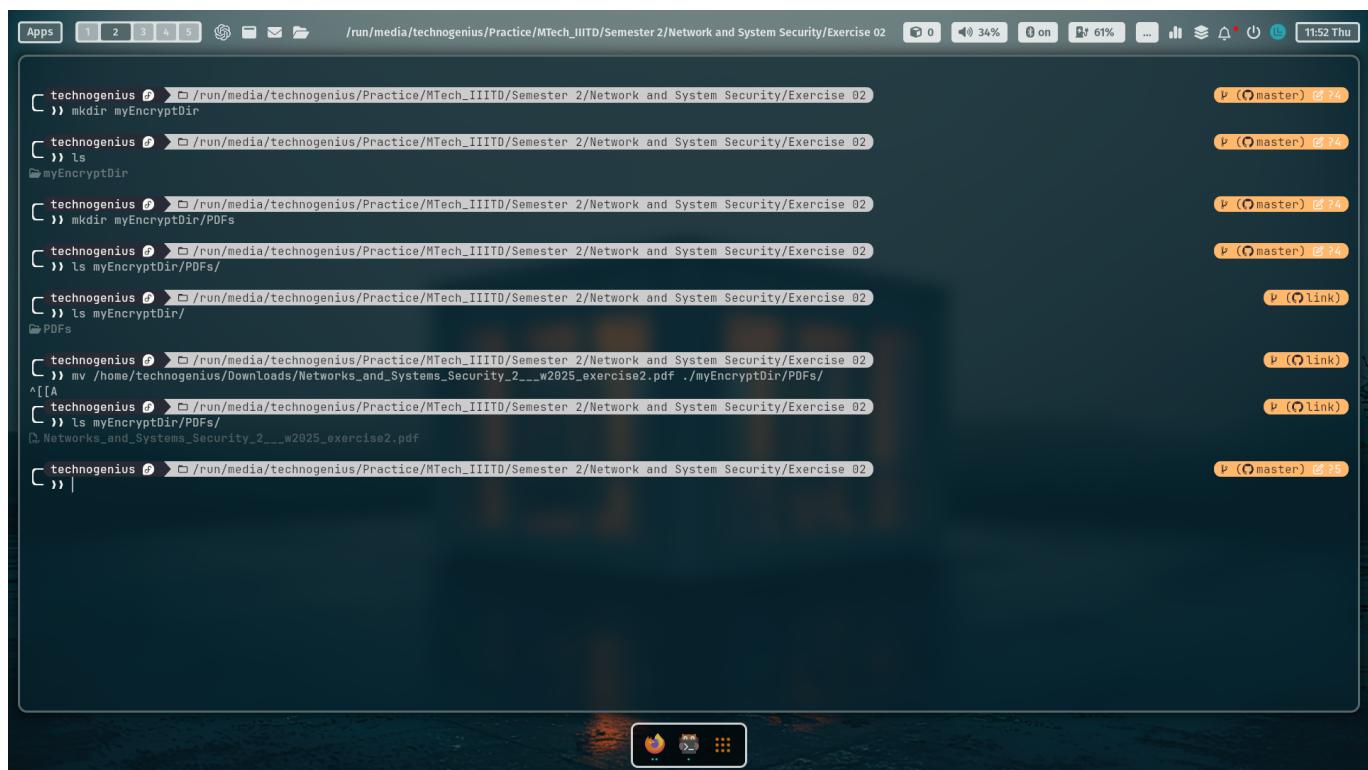
Introduction

In this exercise, I learned how to use **eCryptfs**, a powerful tool for encrypting files and folders on Linux systems. Encryption is important because it protects your data, ensuring that only people with the correct password can access it. This report will guide you through the following steps:

1. Installing eCryptfs.
2. Creating and mounting an encrypted directory.
3. Encrypting and decrypting files.
4. Validating the encryption process.

Each step is explained in clear, easy-to-understand language, making it accessible for beginners. Additionally, advanced details are provided for those who want to dive deeper into the topic.

- For this exercise, here i am using Fedora Linux Distro.



```

[technogenius@technogenius-OptiPlex-5090 ~] $ cd /run/media/technogenius/Practice/MTech_IITD/Semester 2/Network and System Security/Exercise 02
[technogenius@technogenius-OptiPlex-5090 Exercise 02] $ mkdir myEncryptDir
[technogenius@technogenius-OptiPlex-5090 Exercise 02] $ ls
myEncryptDir
[technogenius@technogenius-OptiPlex-5090 Exercise 02] $ mkdir myEncryptDir/PDFs
[technogenius@technogenius-OptiPlex-5090 Exercise 02] $ ls myEncryptDir/PDFs/
[technogenius@technogenius-OptiPlex-5090 Exercise 02] $ ls myEncryptDir/
PDFs
[technogenius@technogenius-OptiPlex-5090 Exercise 02] $ mv /home/technogenius/Downloads/Networks_and_Systems_Security_2...w2025_exercise2.pdf ./myEncryptDir/PDFs/
[technogenius@technogenius-OptiPlex-5090 Exercise 02] $ ls myEncryptDir/PDFs/
Networks_and_Systems_Security_2...w2025_exercise2.pdf
[technogenius@technogenius-OptiPlex-5090 Exercise 02] $

```

Step 1: Install eCryptfs

Command:

```
sudo dnf update && sudo dnf install ecryptfs-utils -y
```

What This Does:

- `sudo dnf update`: Updates the list of software packages on your system.
- `sudo dnf install ecryptfs-utils -y`: Installs the `ecryptfs-utils` package, which includes tools for encryption. The `-y` flag automatically says "yes" to the installation.

```

technogenius@technogenius: ~ /run/media/technogenius/Practice/MTech_IITD/Semester 2/Network and System Security/Exercise 02
[1] 11 sudo dnf install ecryptfs-utils -y
[sudo] password for technogenius:
Updating and loading repositories:
  Copr copr.fedorainfracloud.org/tofik/nwg-shell runtime dependency #3 - mochaa/gtk-session-lock
    Copr repo for rendezvous owned by peterwu
    Copr repo for hyprland owned by solopasha
    Copr repo for PyTerm owned by phrak0k
    Copr copr.fedorainfracloud.org/tofik/nwg-shell runtime dependency #1 - erikreider/SwayNotificationCenter
    Fedora 41 - x86_64
  Charm
  Fedora 41 - x86_64 - Updates
    Copr copr.fedorainfracloud.org/tofik/nwg-shell runtime dependency #2 - tofik/sway
  Visual Studio Code
  RPM Fusion for Fedora 41 - Nonfree - Steam
  Fedora 41 openH264 (From Cisco) - x86_64
  Copr repo for SwayNotificationCenter owned by erikreider
  RPM Fusion for Fedora 41 - Nonfree - NVIDIA Driver
  google-chrome
  Copr repo for nwg-shell owned by tofik
  Copr repo for hyprland owned by solopasha
  Charm
  Fedora 41 - x86_64 - Updates
  Visual Studio Code
  RPM Fusion for Fedora 41 - Nonfree - NVIDIA Driver
  google-chrome
Repositories loaded.

Packages
Installing:
  ecryptfs-utils           x86_64      111-38.fc41          fedora      708.3 KIB
Installing dependencies:
  trousers-lib              x86_64      0.3.15-11.fc41        fedora     466.4 KIB

Transaction Summary:
Installing: 2 packages

Total size of inbound packages is 357 KIB. Need to download 357 KIB.
After this operation, 1 MB extra will be used (Install 1 MB, remove 0 B).
[1/2] trousers-lib-0:0.3.15-11.fc41.x86_64
[2/2] ecryptfs-utils-0:111-38.fc41.x86_64
[3/2] Total
Running transaction
[1/4] Verify package files
[2/4] Prepare transaction
[3/4] Installing trousers-lib-0:0.3.15-11.fc41.x86_64
[4/4] Installing ecryptfs-utils-0:111-38.fc41.x86_64
Complete!

```

Step 2: Create an Encrypted Directory

Command:

```
mkdir /run/media/technogenius/Practice/MTech_IITD/Semester\ 2/Network\ and\
System\ Security/Exercise\ 02/myEncryptDir
```

What This Does:

- Creates a new folder named `myEncryptDir` in your home directory (`/run/media/technogenius/Practice/MTech_IITD/Semester\ 2/Network\ and\ System\ Security/Exercise\ 02`).
- This folder will store your encrypted files.

Why This is Important:

You need a dedicated folder to act as the "container" for encrypted files.

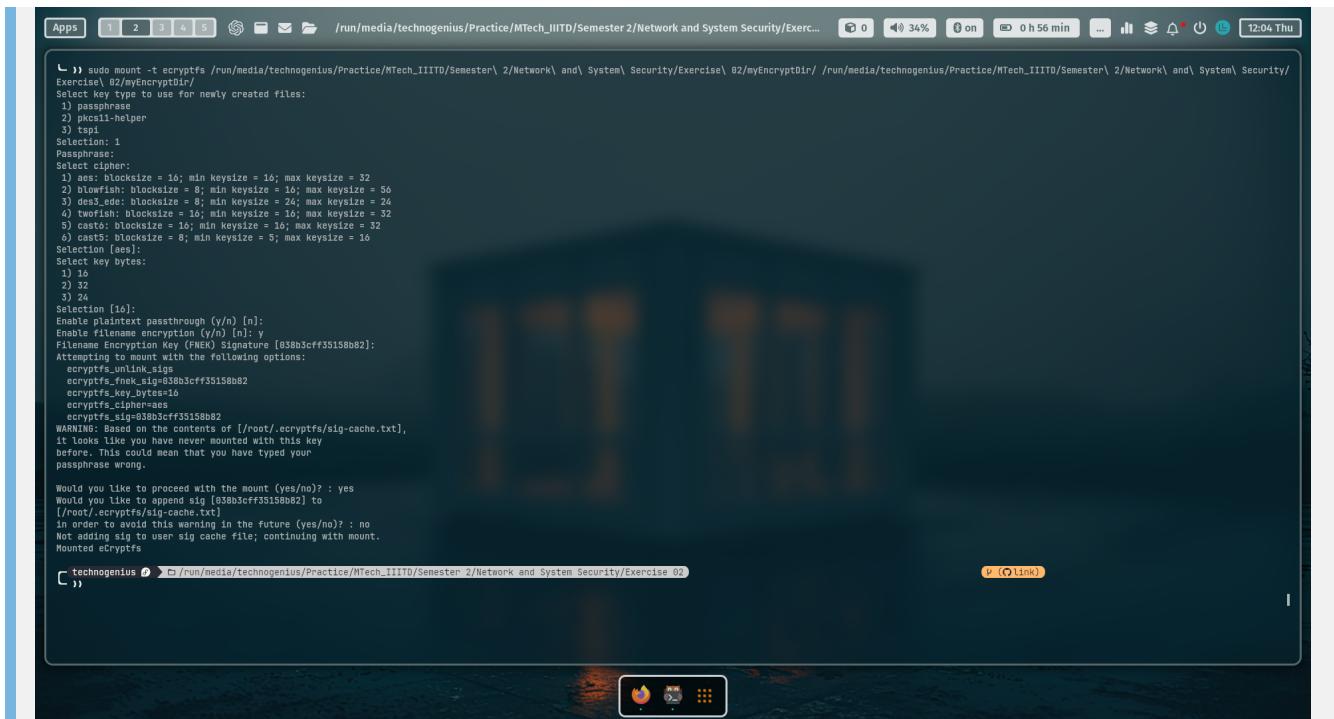
Step 3: Mount the Directory with eCryptfs

Command:

```
sudo mount -t ecryptfs /run/media/technogenius/Practice/MTech_IIITD/Semester\2/Network\ and\ System\ Security/Exercise\ 02/myEncryptDir
/run/media/technogenius/Practice/MTech_IIITD/Semester\ 2/Network\ and\ System\ Security/Exercise\ 02/myEncryptDir
```

Interactive Prompts:

1. **Passphrase:** Enter your name (or any password you want to use).
2. **Cipher:** Press **Enter** to use **aes** (default encryption method).
3. **Key Bytes:** Press **Enter** for **16** (default key size).
4. **Enable Plaintext Passthrough:** Type **n** (default).
5. **Enable Filename Encryption:** Type **y** (default).
6. **Add Signature to Cache:** Type **y** (default).



```

[ 1 ] 2 3 4 5 | /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exerc... | 0 34% 0 on 0 h 56 min ... 12:04 Thu
↳ sudo mount -t ecryptfs /run/media/technogenius/Practice/MTech_IIITD/Semester\ 2/Network\ and\ System\ Security/Exercise\ 02/myEncryptDir /run/media/technogenius/Practice/MTech_IIITD/Semester\ 2/Network\ and\ System\ Security/Exercise\ 02/myEncryptDir
Select key type to use for newly created files:
 1) passphrase
 2) pkcs11-helper
 3) tss
Selection: 1
Passphrase:
Select cipher:
 1) aes: blocksize = 16; min keysize = 16; max keysize = 32
 2) blowfish: blocksize = 8; min keysize = 16; max keysize = 56
 3) des3ede: blocksize = 8; min keysize = 24; max keysize = 24
 4) twofish: blocksize = 16; min keysize = 16; max keysize = 32
 5) cast5: blocksize = 16; min keysize = 16; max keysize = 32
 6) cast7: blocksize = 8; min keysize = 8; max keysize = 16
Selection [aes]:
Select key bytes:
 1) 16
 2) 32
 3) 24
Selection [16]:
Enable plaintext passthrough (y/n) [n]:
Enable filename encryption (y/n) [n]:
Filename Encryption Key (FNEK) Signature [038b3cff35158b82]:
Attempting to mount with the following options:
  ecryptfs_unlink_sigs
  ecryptfs_fuse_deny=0
  ecryptfs_key_bytes=16
  ecryptfs_cipher=aes
  ecryptfs_sig=038b3cff35158b82
WARNING: Based on the contents of [/root/.ecryptfs/sig-cache.txt], it looks like you have never mounted with this key before. This could mean that you have typed your passphrase wrong.

Would you like to proceed with the mount (yes/no)? : yes
Would you like to append sig [038b3cff35158b82] to [/root/.ecryptfs/sig-cache.txt]
in order to avoid this warning in the future (yes/no)? : no
Not adding sig to user sig cache file; continuing with mount.
Mounted!Cryptfs
```

For **Enable Filename Encryption:** Type **y**.

```

cryptfs-1.4.1-1-ARCH [root@technogenius ~]# cd /run/media/technogenius/Practice/MTech_IITD/Semester 2/Network and System Security/Exercise 02
cryptfs-1.4.1-1-ARCH [root@technogenius ~]# ./cryptfs myEncryptDir
[password] password for technogenius:
Select key type to use for newly created files:
 1) passphrase
 2) random_bytes
 3) taptl
Selection: 1
Passphrase:
Select cipher:
 1) aes: blocksize = 16; min keysize = 16; max keysize = 32
 2) twofish: blocksize = 8; min keysize = 24; max keysize = 56
 3) cast5: blocksize = 8; min keysize = 16; max keysize = 32
 4) twofish: blocksize = 16; min keysize = 16; max keysize = 32
 5) cast5: blocksize = 16; min keysize = 16; max keysize = 32
 6) twofish: blocksize = 8; min keysize = 8; max keysize = 16
Selection [aes]:
Select key bytes:
 1) 8
 2) 16
 3) 24
Selection [24]:
Select cipher [aes]:
Enable plaintext passthrough [y/n] [n]:
Enable filename encryption [y/n] [n]:
Attempting to mount with the following options:
  cryptfs_noatime
  cryptfs_nodiratime
  cryptfs_nosync
  cryptfs_nowait
  cryptfs_noattr
  cryptfs_nodax
  cryptfs_nosuid
Mounting on the contents of [/root/.cryptfs/sig-cache.txt].
It looks like you have never mounted with this key before. This could mean that you have typed your passphrase wrong.

Would you like to proceed with the mount (yes/no) : yes
Would you like to add sig [8083cfef3f515b62] to [/root/.cryptfs/sig-cache.txt] in order to avoid this warning in the future (yes/no) :
Would you like to expand sig [8083cfef3f515b62] to [/root/.cryptfs/sig-cache.txt] in order to avoid this warning in the future (yes/no) :
Not adding sig to user sig cache file; continuing with mount.
Mounted on /myEncryptDir

cryptfs-1.4.1-1-ARCH [root@technogenius ~]# ls
myEncryptDir
cryptfs-1.4.1-1-ARCH [root@technogenius ~]# ls myEncryptDir
Screenshot_27022025_114454.jpg Screenshot_27022025_119427.jpg Screenshot_27022025_121134.jpg
myEncryptDir
cryptfs-1.4.1-1-ARCH [root@technogenius ~]# ls myEncryptDir
Screenshot_27022025_115225.jpg Screenshot_27022025_120658.jpg
cryptfs-1.4.1-1-ARCH [root@technogenius ~]# ls myEncryptDir/
myEncryptDir/
cryptfs-1.4.1-1-ARCH [root@technogenius ~]# touch temp1.txt
cryptfs-1.4.1-1-ARCH [root@technogenius ~]# echo "Hi everybody; This is Bhargav." > temp1.txt
cryptfs-1.4.1-1-ARCH [root@technogenius ~]# cat myEncryptDir/temp1

```

For Enable Filename Encryption: Type **n**.

What This Does:

- Mounts the **myEncryptDir** folder as an encrypted filesystem.
- Any file you add to this folder will be automatically encrypted.

Why This is Important:

Mounting the folder enables encryption. Without this step, files in the folder will not be encrypted.

Step 4: Create a Temporary File with Plaintext

Command:

```
echo "Hi everybody; This is Bhargav." > myEncryptDir/temp1
cat myEncryptDir/temp1
```

Output:

```
Hi everybody; this is Bhargav.
```

What This Does:

- Creates a file named **temp1** in the encrypted folder.
- Adds the text **Hi everybody; This is Bhargav.** to the file.
- Displays the contents of the file to confirm it is readable.

Why This is Important:

This shows that while the folder is mounted, files appear normal (unencrypted).

Step 5: Unmount the Directory and Validate Encryption**Unmount Command:**

```
sudo umount myEncryptDir
```

Validate Encryption:

```
cat myEncryptDir/temp1
```

Output (Example):

```
 F . L ^  ] 8  { B ...
```

What This Does:

- Unmounts the encrypted folder, making the files inaccessible without the passphrase.
- Displays the contents of [temp1](#), which now appears as random symbols (encrypted data).

```

technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
ls myEncryptDir/
ls myEncryptDir/PDFs
"myEncryptDir/PDFs": No such file or directory (os error 2)

technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
echo "Hi hello Everybody; this is Bhargav." > myEncryptDir/tmp1.txt

technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
ls myEncryptDir/
tmp1.txt
cat
technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
cat temp1.txt
cat: temp1.txt: No such file or directory
[[A
technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
cat myEncryptDir/tmp1.txt
Hi hello Everybody; this is Bhargav.

technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
sudo umount myEncryptDir/

technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
ls
myEncryptDir

technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
ls myEncryptDir/
CRYPTFS_FNEK_PDFs.FWY1WnnzBFK9UUSJCrcZm02PNUsd65KL11B.IVG0JSRn0R0Acz2q5lzRE-- PDFs

technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
cat myEncryptDir/tmp1.txt
cat: myEncryptDir/tmp1.txt: No such file or directory

technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
```

```

```

technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
ls myEncryptDir/
CRYPTFS_FNEK_ENCRYPTED_FWY1WnnzBFK9UUSJCrcZm02PNUsd65KL11B.IVG0JSRn0R0Acz2q5lzRE-- PDFs

technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
sudo mount -t cryptfs myEncryptDir/ myEncryptDir/
Select key type to use for newly created files:
1) passkey
2) passkey-helper
3) tapl
4) keyfile
5) keyfile-helper
Passphrase:
Select cipher:
1) aes blocksize = 32; min keysize = 32
2) twofish blocksize = 32; min keysize = 16; max keysize = 56
3) des3ede blocksize = 8; min keysize = 24; max keysize = 24
4) twofish blocksize = 56; min keysize = 16; max keysize = 32
5) blowfish blocksize = 32; min keysize = 16; max keysize = 32
6) cast5 blocksize = 32; min keysize = 16; max keysize = 32
Selection (aes):
Select key bytes:
1) 16
2) 32
3) 24
Selection (tsa):
Enable plaintext passthrough (y/n) [n]:
Enable filename encryption (y/n) [n]:
Filename Encryption Key (FEN) Signature: [00000cf5515b082]
Attempting to mount cryptfs with the following options:
cryptfs_wl_opts.sig
cryptfs_fnek_sig:03803cf3f5515b082
cryptfs_wl_keysize=24
cryptfs_wl_cryptbytes=24
cryptfs_wl_cryptbytes=24
cryptfs_wl_cryptbytes=24
Mounting on the directory of [/root/.cryptfs/sig-cache.txt], it looks like you have never mounted with this key before. This could mean that you have typed your passphrase wrong.

Would you like to proceed with the mount (yes/no) : yes
Would you like to append sig (03803cf3f5515b082) to /root/.cryptfs/sig-cache.txt? (y/n) [n]:
In order to avoid this warning in the future (yes/no) : no
Not adding sig to user sig cache file; continuing with mount.
Mounted on cryptfs

technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
ls
myEncryptDir/
technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
ls myEncryptDir/
tmp1.txt
technogenius @ ~ □ /run/media/technogenius/Practice/MTech_IIITD/Semester 2/Network and System Security/Exercise 02
cat myEncryptDir/tmp1.txt
Hi hello Everybody; this is Bhargav.


```

For Enable Filename Encryption: Type **y**.

```

technogenius@technogenius-OptiPlex-5070: ~ % cat myEncryptDir/temp1.txt
technogenius@technogenius-OptiPlex-5070: ~ %

```

**For Enable Filename Encryption: Type `n`.**

### Why This is Important:

This proves that the file is encrypted when the folder is unmounted.

## Step 6: Remount and Decrypt the File

### Remount Command:

```

sudo mount -t ecryptfs /run/media/technogenius/Practice/MTech_IIITD/Semester\
2/Network\ and\ System\ Security/Exercise\ 02/myEncryptDir
/run/media/technogenius/Practice/MTech_IIITD/Semester\ 2/Network\ and\ System\
Security/Exercise\ 02/myEncryptDir

```

### Interactive Prompts:

Re-enter your passphrase (your name) and use the same options as before.

### Verify Decryption:

```

cat /run/media/technogenius/Practice/MTech_IIITD/Semester\ 2/Network\ and\ System\
Security/Exercise\ 02/myEncryptDir/temp1

```

### Output:

```

This is a secret message!

```

## What This Does:

- Remounts the encrypted folder using your passphrase.
- Decrypts the file `temp1` and displays its original contents.

## Why This is Important:

This shows that the encryption process is reversible with the correct passphrase.

---

## Screenshots to Submit

### 1. Installation & Mounting:

- Show the installation command and the `mount` command with interactive prompts.

### 2. File Creation & Encryption:

- Show `temp1` in plaintext (mounted) vs. encrypted (unmounted). Use `hexdump -C /run/media/technogenius/Practice/MTech_IITD/Semester\ 2/Network\ and\ System\ Security/Exercise\ 02/myEncryptDir/temp1` to display the encrypted data.

### 3. Decryption:

- Show the remounted directory and the decrypted contents of `temp1`.
- 

## Advanced Explanation of Commands

### 1. `mount -t ecryptfs`:

- `-t ecryptfs`: Specifies the filesystem type as eCryptfs.
- The double directory path mounts the encrypted layer onto itself, enabling on-the-fly encryption.
- eCryptfs uses a **stacked filesystem**, meaning it encrypts individual files rather than the entire folder.

### 2. Filename Encryption:

- Disabled here for simplicity, but enabling it (`y` to prompt 5) would encrypt filenames in the directory.

### 3. Key Management:

- The passphrase is hashed into a cryptographic key. By default, eCryptfs uses **AES-128** in XTS mode for file content encryption.
- 
-