Protecting User Password Keys at Rest (on the Disk)

Protecting user password keys at rest—when they are stored on disk—is crucial for data security and privacy. This involves using encryption techniques, specialized hardware and software solutions, and stringent access controls to prevent unauthorized access. Key strategies include employing Folder and File-level encryption, utilizing **Hardware security modules** (HSMs) and **Trusted platform modules** (TPMs), and implementing **key management systems** (KMS). These measures help organizations mitigate risks, ensuring the integrity and confidentiality of user data.

Features of Protecting User Password Keys at Rest (on the disk)

➢ **Encryption**:

- **Folder Encryption**: Encrypts the entire storage device to protect all data stored on it.
- **File-Level Encryption**: Encrypts individual files or directories containing sensitive information.

➢ **Hardware Security Modules (HSMs)**:

- Securely generate, store, and manage cryptographic keys.
- Provide physical and logical protection against tampering and unauthorized access.

➢ **Trusted Platform Modules (TPMs)**:

- Hardware-based security feature for secure storage of keys, passwords, and other critical data.
- Ensures keys are accessible only under specific, trusted conditions.

➢ **Key Management Systems (KMS)**:

- Centralized management of cryptographic keys, including creation, distribution, rotation, and revocation.
- Ensures secure storage and access by authorized entities only.

➢ **Decryption**:

- A secret key or password used to decrypt the encrypted data.
- Must be securely managed and protected to prevent unauthorized access.

Uses of Protecting User Password Keys at Rest (on the Disk)

- Data Security
- Compliance and Regulatory Requirements
- Risk Mitigation
- User Trust and Confidence
- Operational Security
- Encryption Key Management
- Access Control Enforcement
- Incident Response and Forensics
- Business Continuity and Disaster Recovery
- Secure Development and Testing

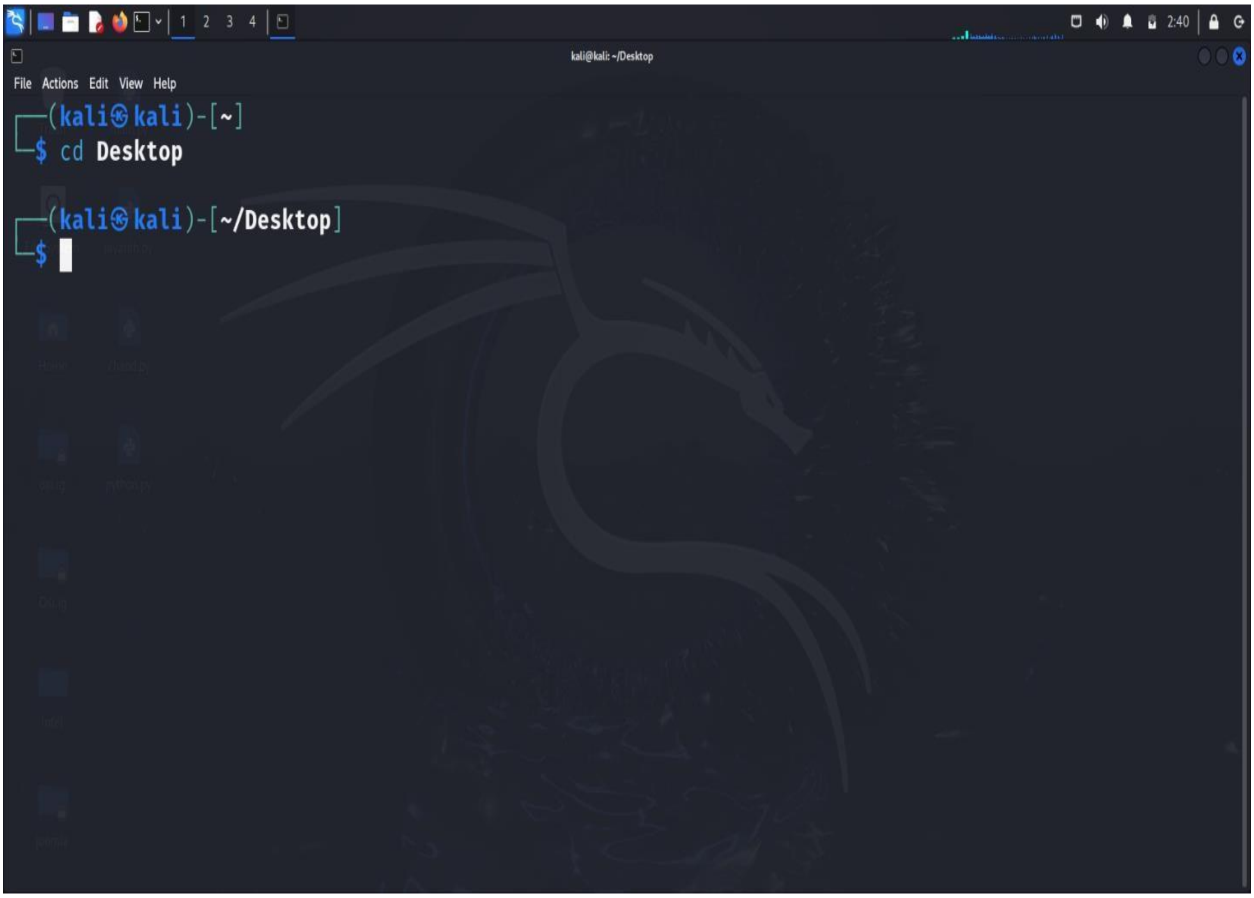
Protecting user password keys at rest is essential for maintaining the security, privacy, and integrity of sensitive information, thereby supporting regulatory compliance, risk management, and operational efficiency

Installation and step-by-step tutorial of protecting user password keys at rest:

Step 1: Open your kali Linux operating system. Move to desktop. Here you have to create a directory called intel project. In this directory, you have to install the tool.

To move to the desktop, use the following command.

```
cd Desktop
```



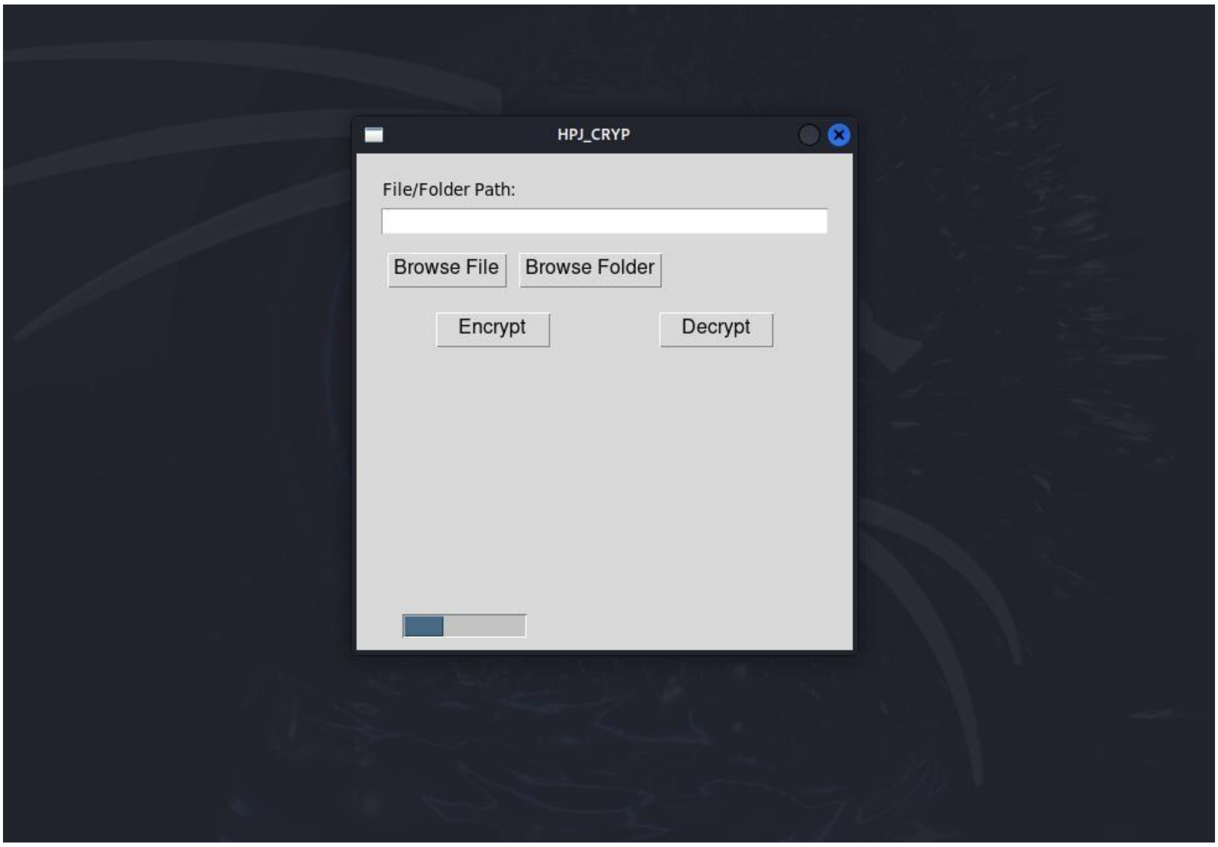
```
┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

File And Folder Encryption and Decryption Process:

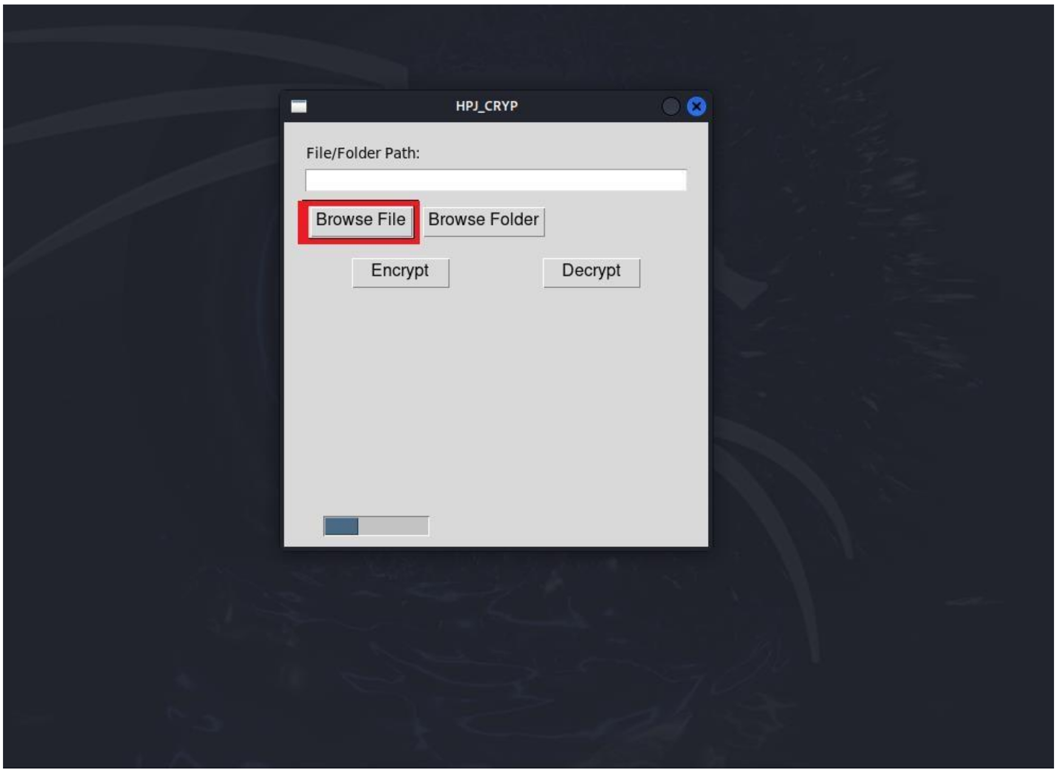
- ➢ File Encryption and Decryption
- ➢ Folder Encryption and Decryption

- **File Encryption Process:**

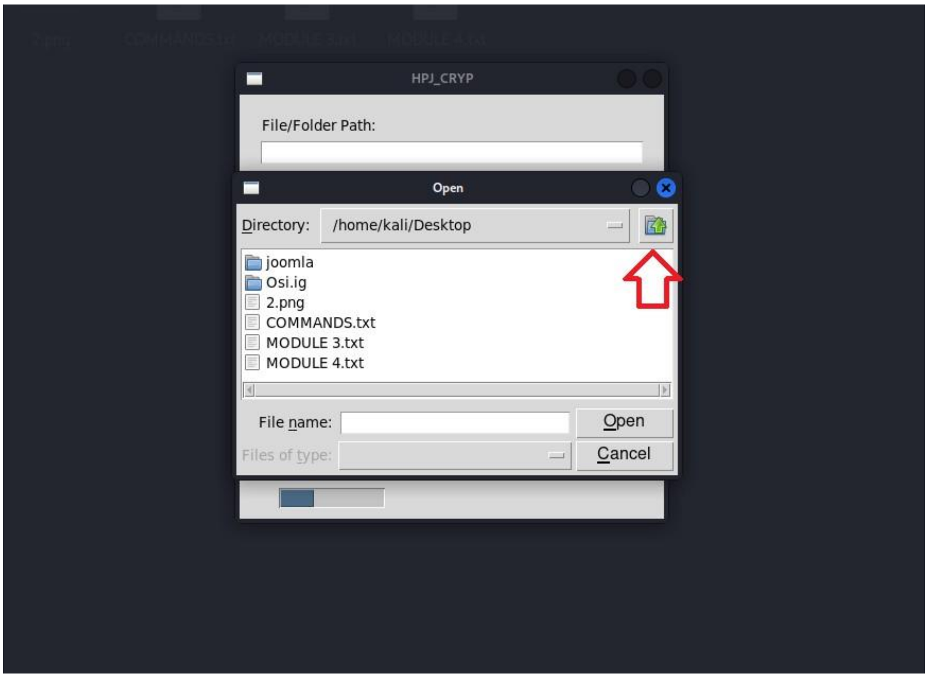
Step : Upon successful implementation of the Python program, the **HPJ_CRYP** Dialog box will display on the screen.



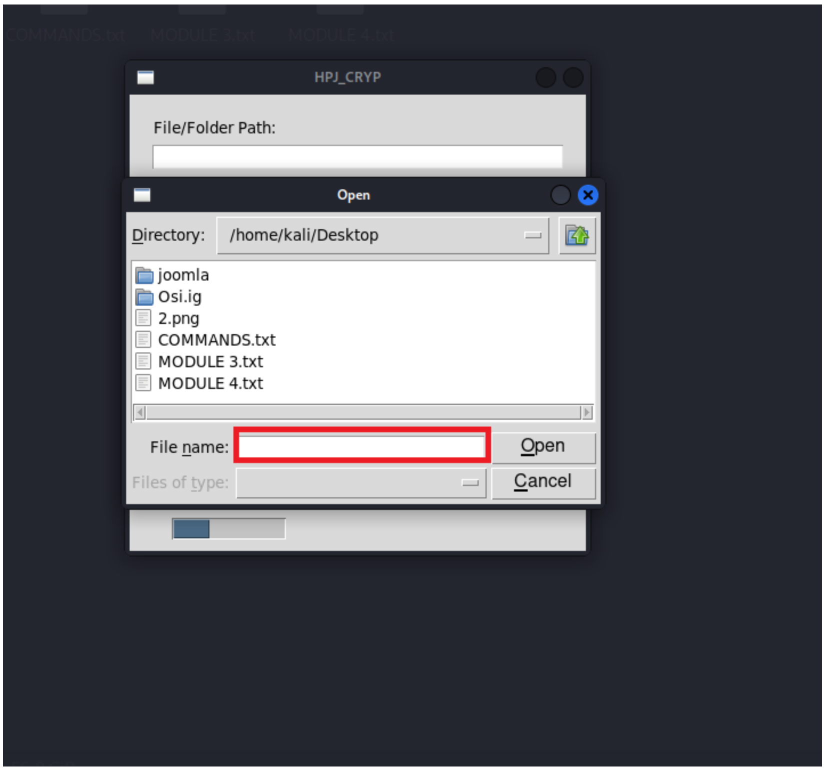
Step : Select the **Browse file** option to encrypt the necessary file.



Step : A dialog box will display with the name of **OPEN** to choose the file. Select the Folder option to acquire your necessary file.



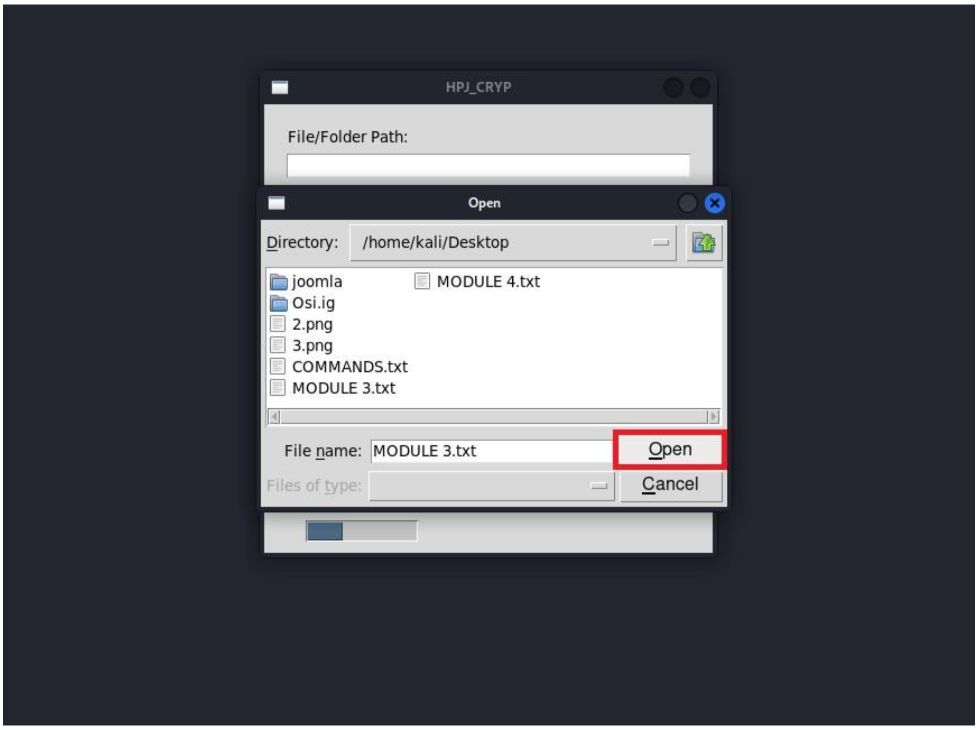
Step : Enter the file name within the File name: field rather than selecting the file. Dialog window



Directory: /home/kali/Desktop

File name:

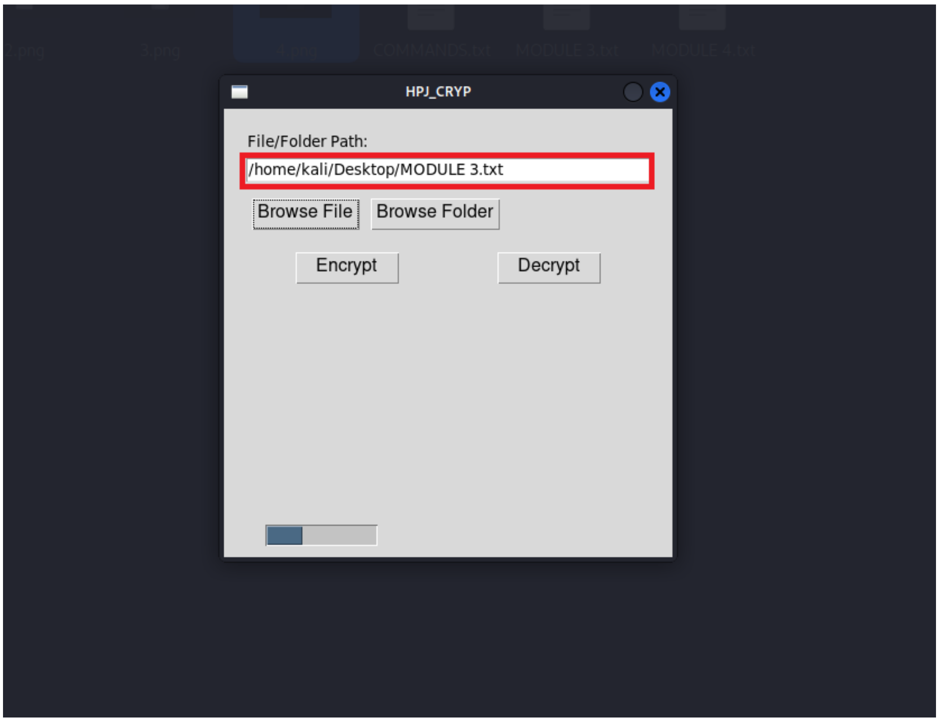
Step : Once the file has been selected, click the "open" button.



Directory: /home/kali/Desktop

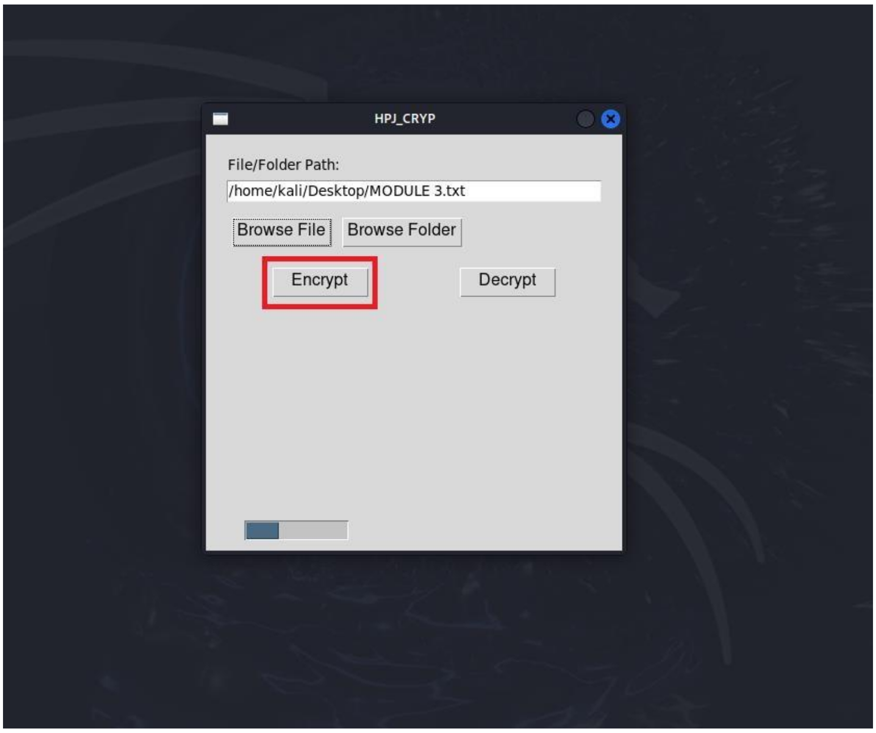
File name: MODULE 3.txt

Step : After pressing the open button, the file path will be displayed in the **File/Folder Path:** where the file is kept.



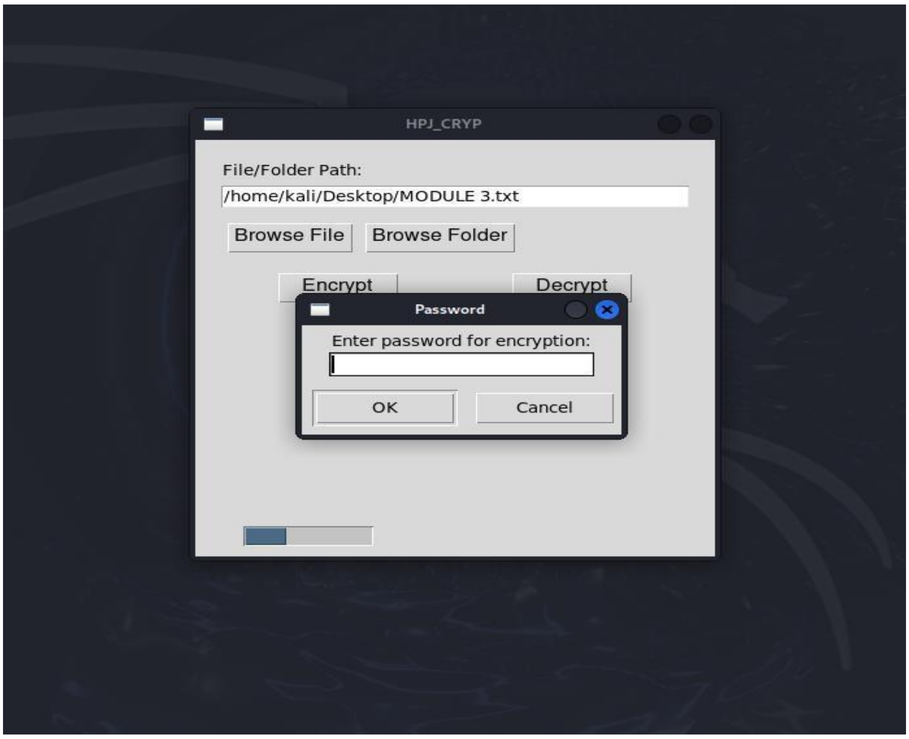
File/Folder Path:
/home/kali/Desktop/MODULE 3.txt

Step : When the file path appears, click on the Encrypt button.



File/Folder Path:
/home/kali/Desktop/MODULE 3.txt

Step : A dialog box named the Password will be presented to input the password.



HPJ_CRYP

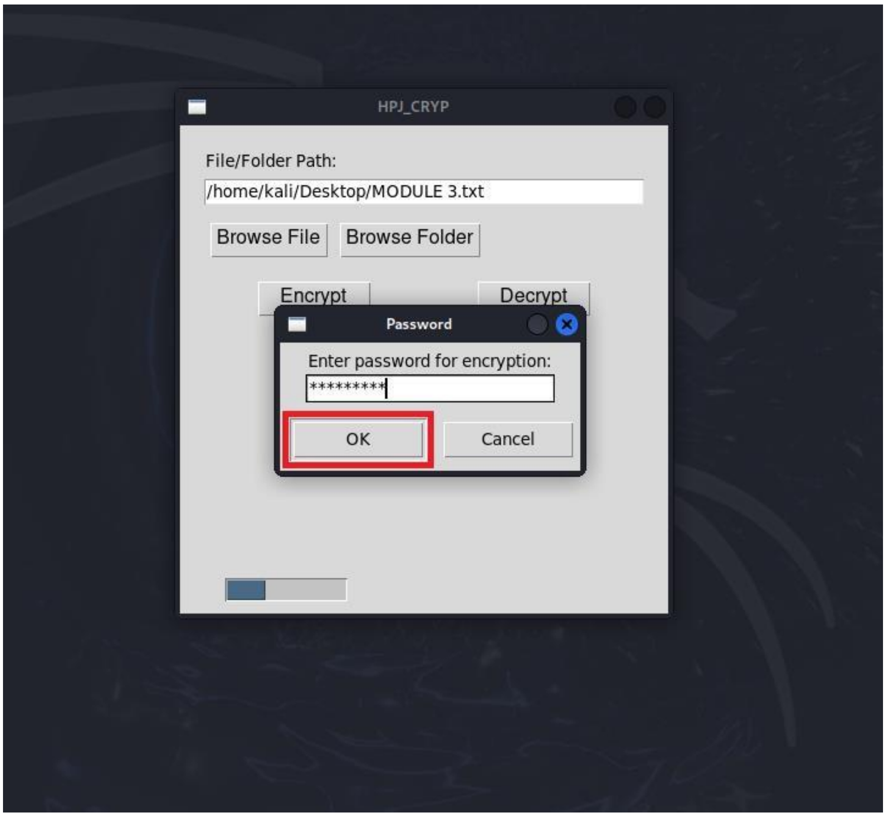
File/Folder Path:

/home/kali/Desktop/MODULE 3.txt

Password

Enter password for encryption:

Step : Enter your password and press the OK button.



HPJ_CRYP

File/Folder Path:

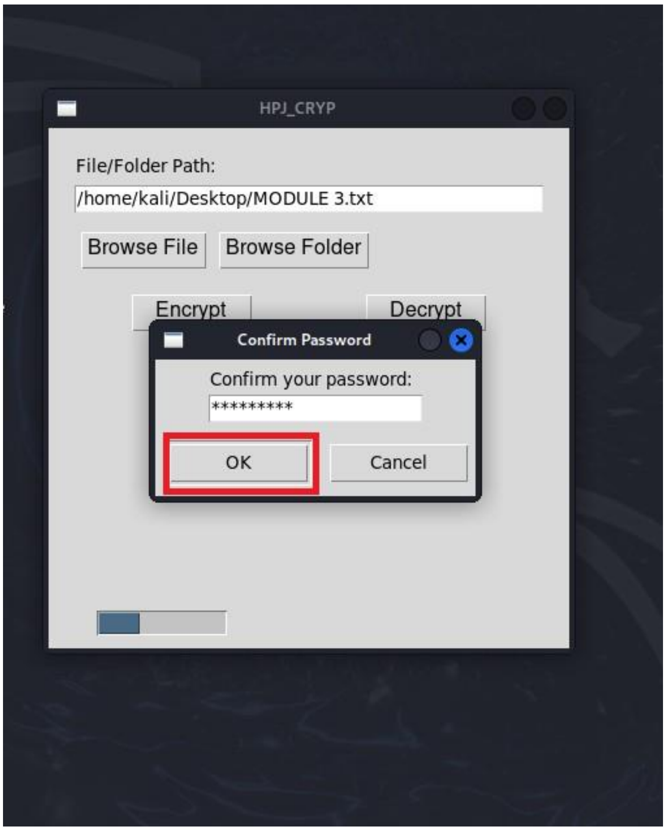
/home/kali/Desktop/MODULE 3.txt

Password

Enter password for encryption:

****

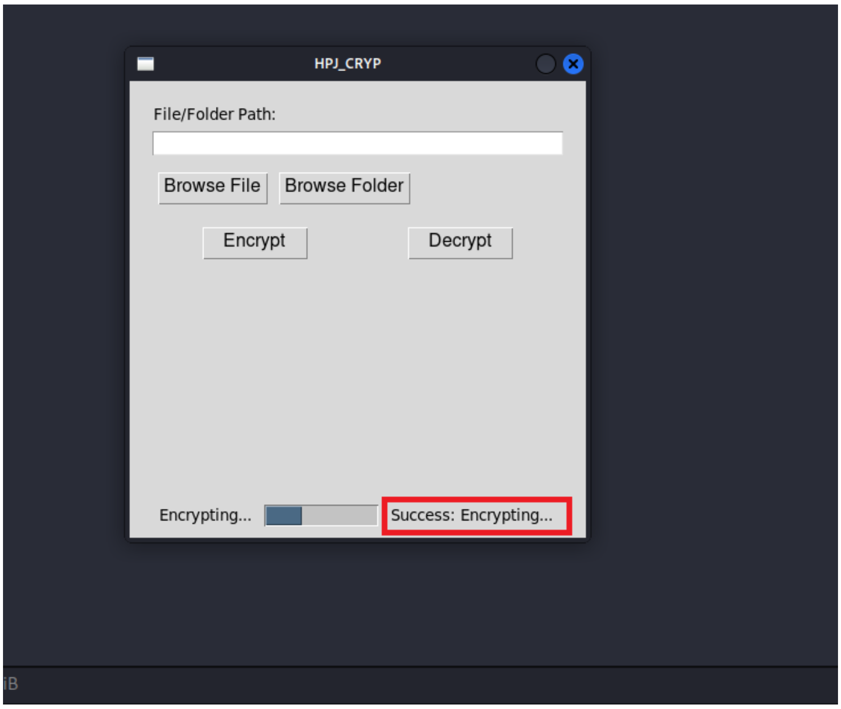
Step : After clicking OK, it will ask for the confirm password. Re-enter the password you entered and press OK.



File/Folder Path: /home/kali/Desktop/MODULE 3.txt

Confirm your password: *********

Step : You will see the success of the Encrypting file.

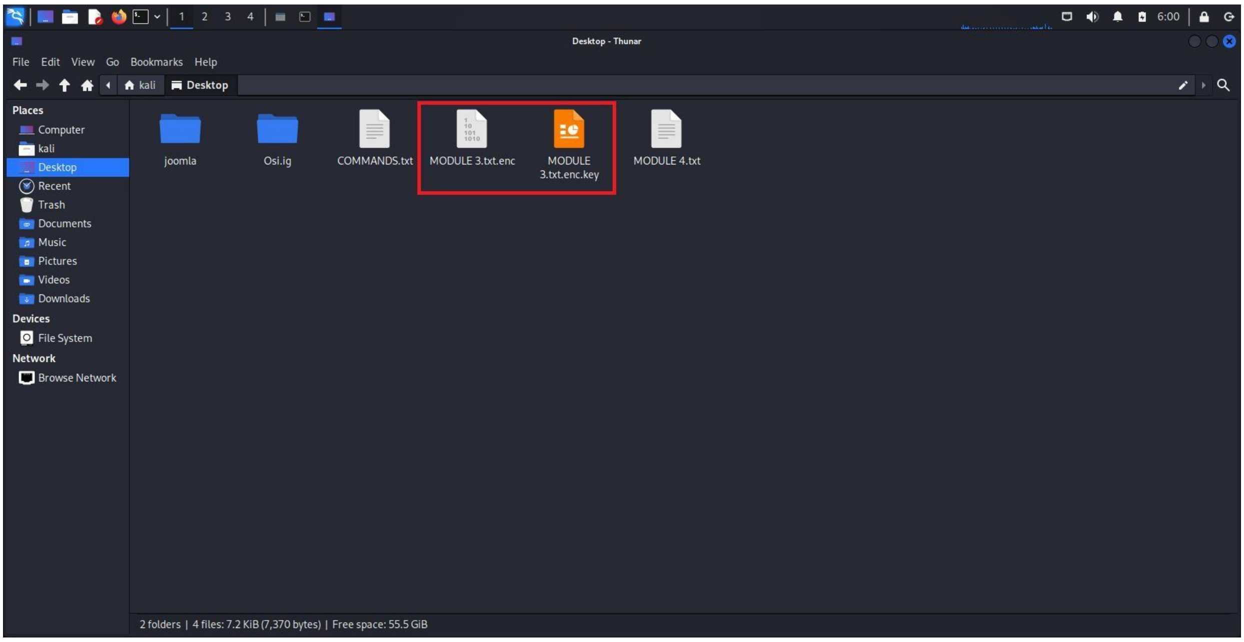


File/Folder Path:

Encrypting...

Success: Encrypting...

Step : As you can see, the file is encrypted and the key file is generated. With the, **.enc**

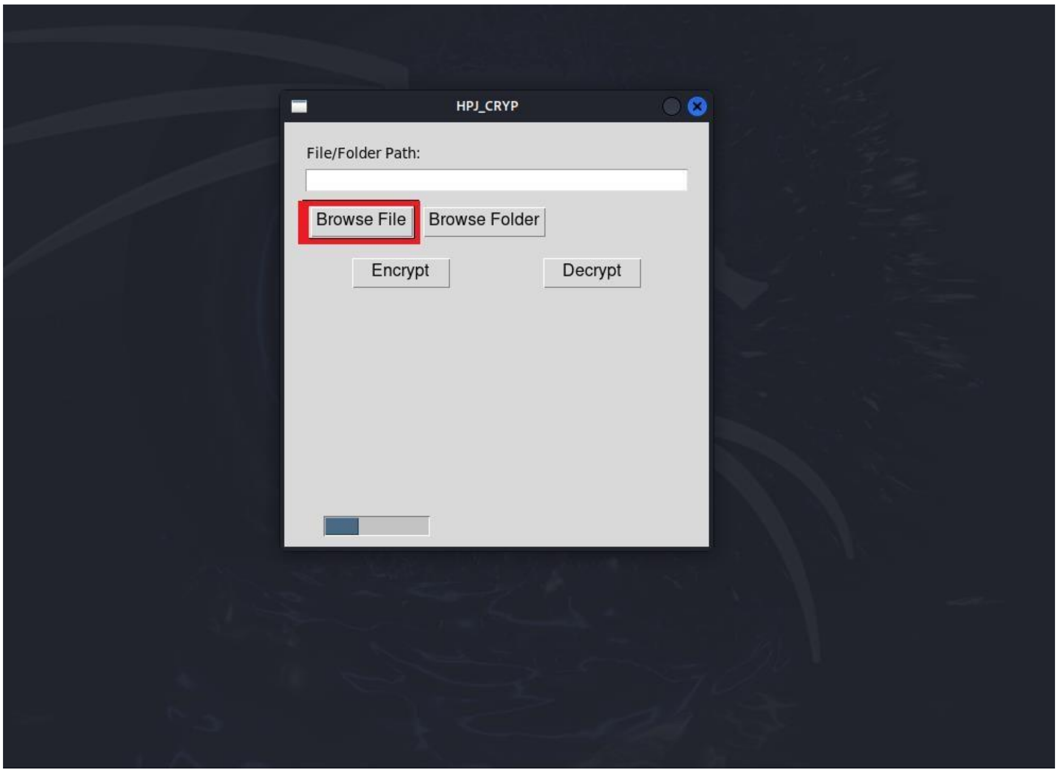


2 folders | 4 files: 7.2 KiB (7,370 bytes) | Free space: 55.5 GiB

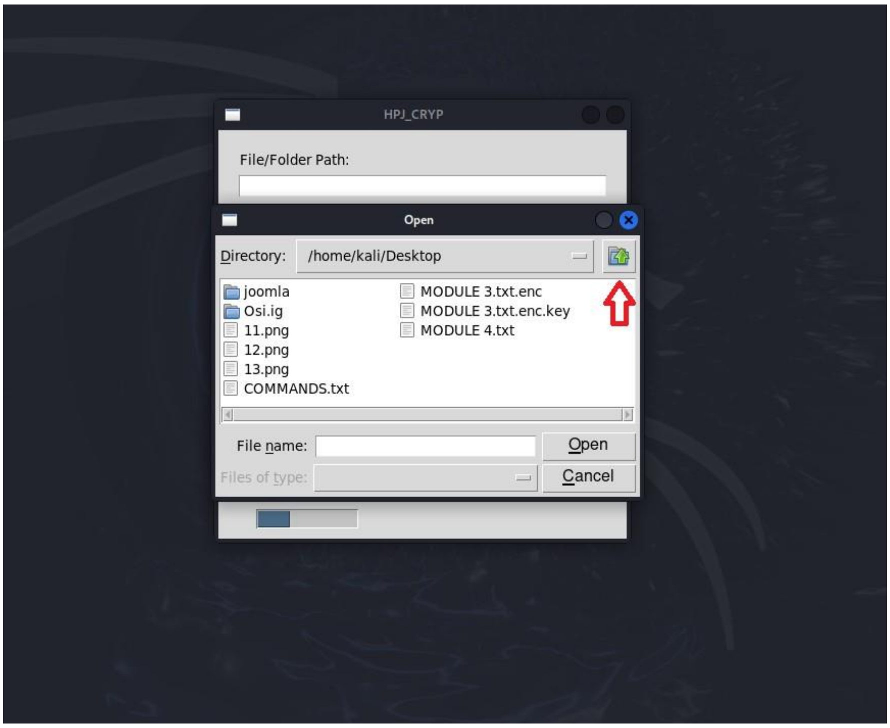
As we have successfully encrypted the file, now if you want to retrieve back the file, we must do the decryption process for the file. Here is the decryption method.

➢ File Decryption process :

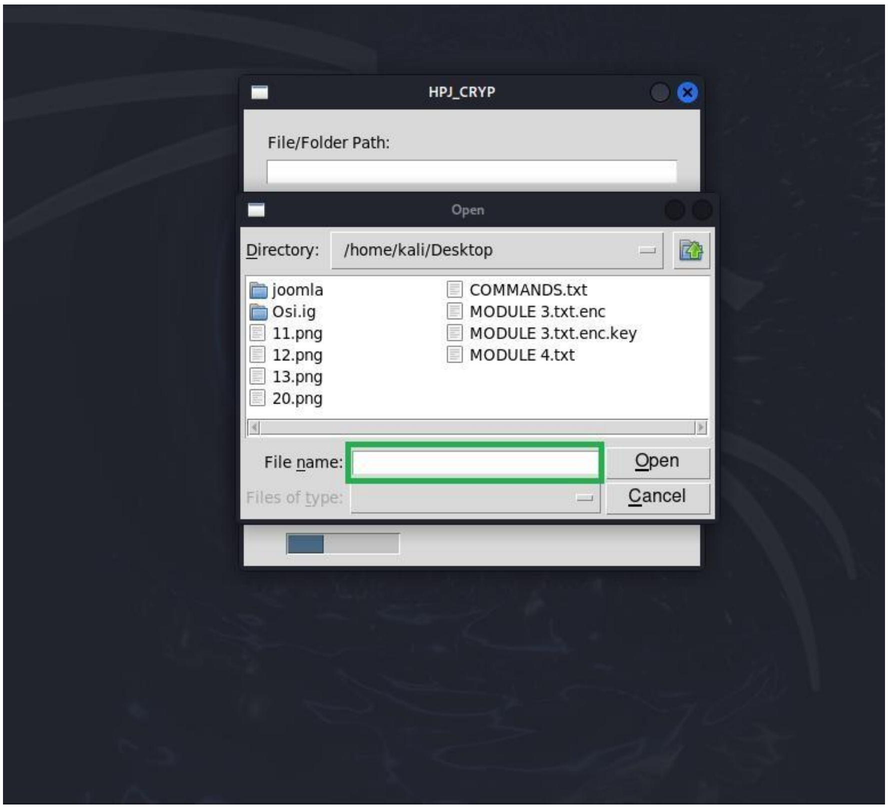
Step : After successfully encrypting, execute the same python file for the decryption process. In the HPJ_CRYP dialog box, select the Browse file option to decrypt the file.



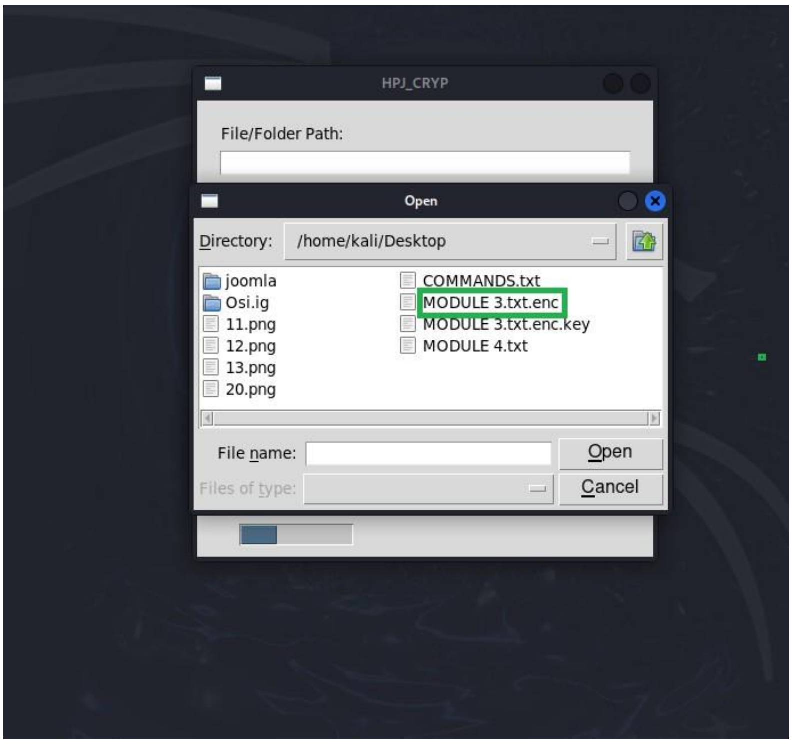
Step : To choose the encrypted file, click the Folder option to acquire your necessary file.



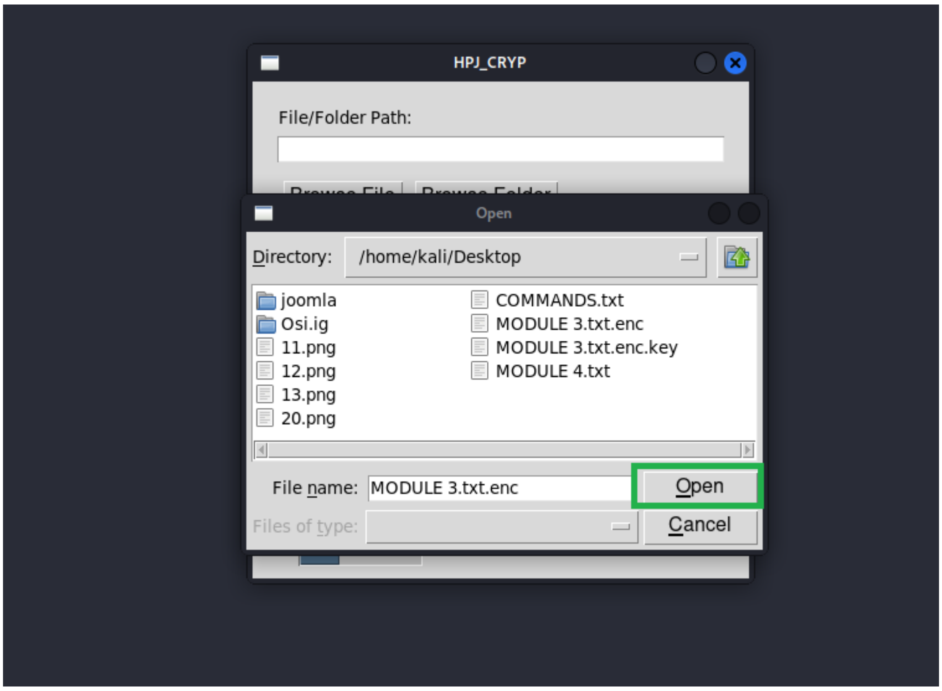
Step : Enter the file name to be decrypted. Instead of selecting the file, type it into the File name: area. Dialog window



Step : Select the **Encrypted** file that should be **Decrypted**



Step : When the file is selected, click the open button.

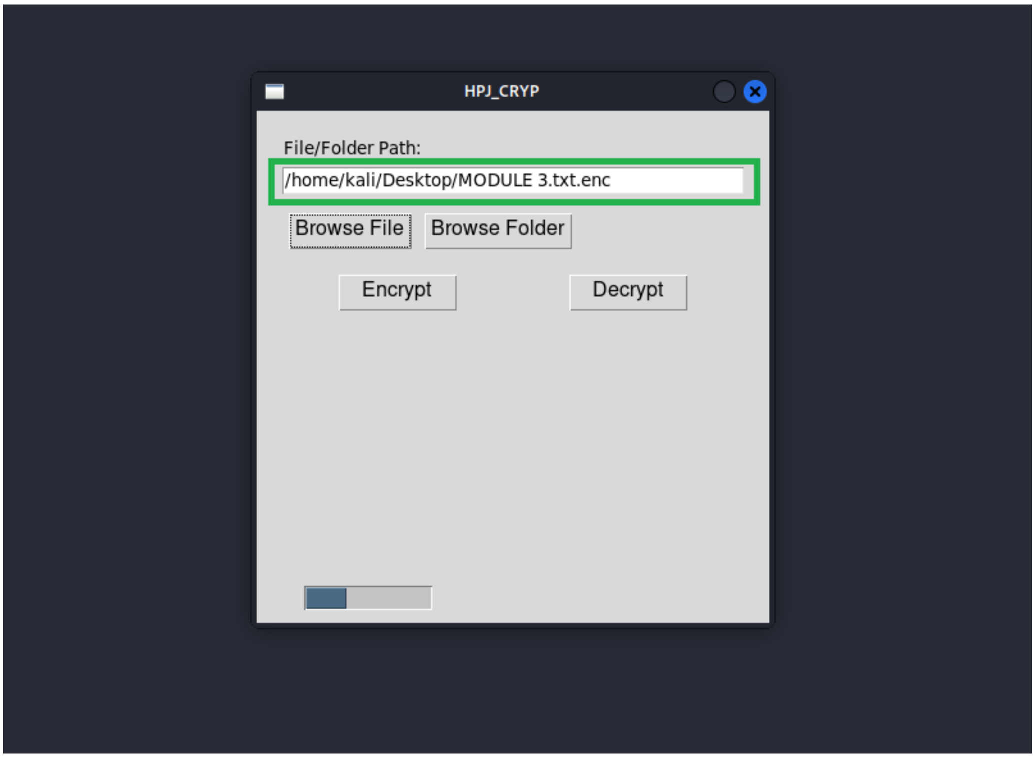


Directory: /home/kali/Desktop

joomla
Osi.ig
11.png
12.png
13.png
20.png
COMMANDS.txt
MODULE 3.txt.enc
MODULE 3.txt.enc.key
MODULE 4.txt

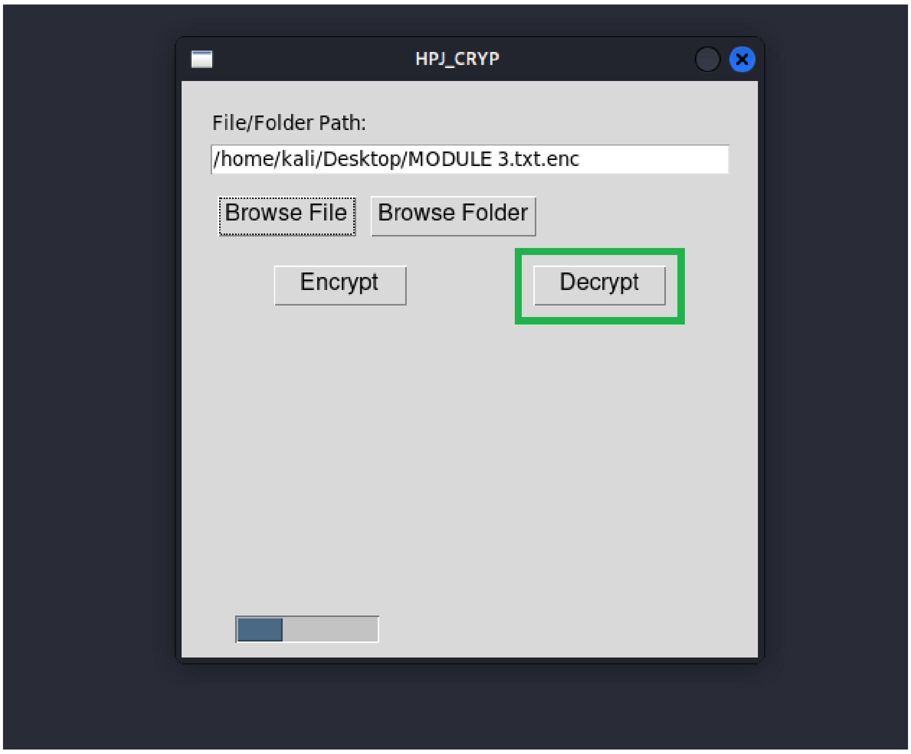
File name: MODULE 3.txt.enc

Step : After clicking the open button, you will see the encrypted file path.



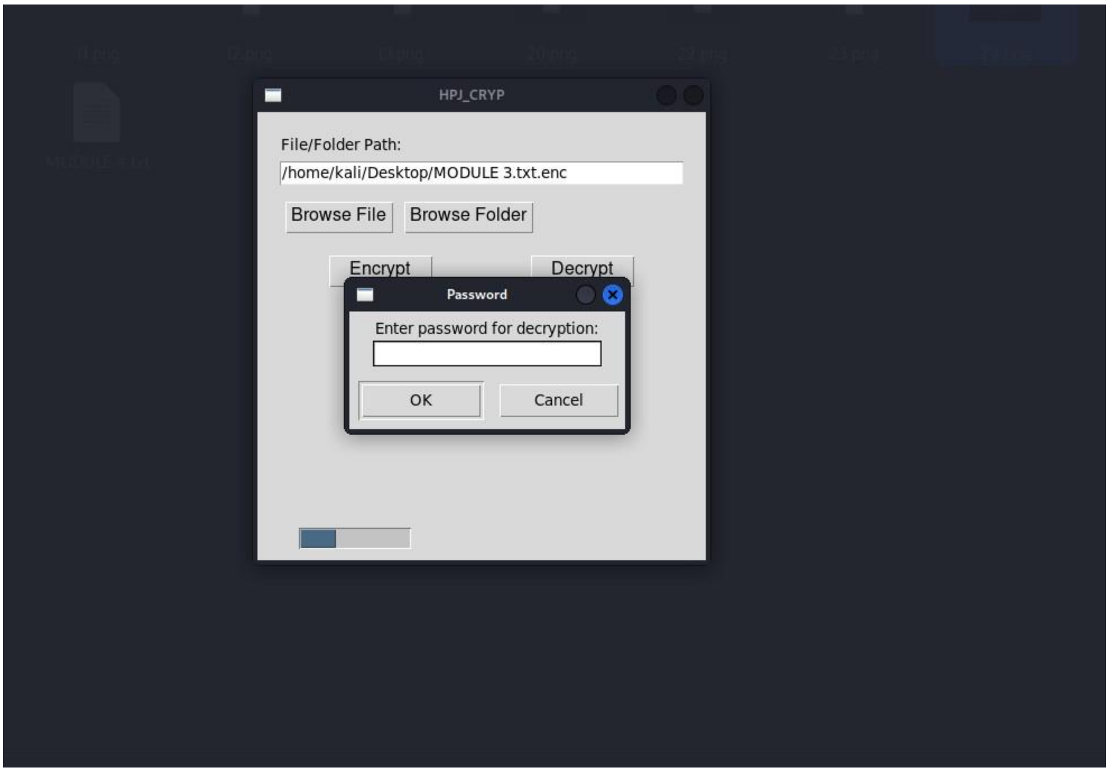
File/Folder Path: /home/kali/Desktop/MODULE 3.txt.enc

Step : Now we are doing the file Decryption process, so click on **Decrypt**.



File/Folder Path:
/home/kali/Desktop/MODULE 3.txt.enc

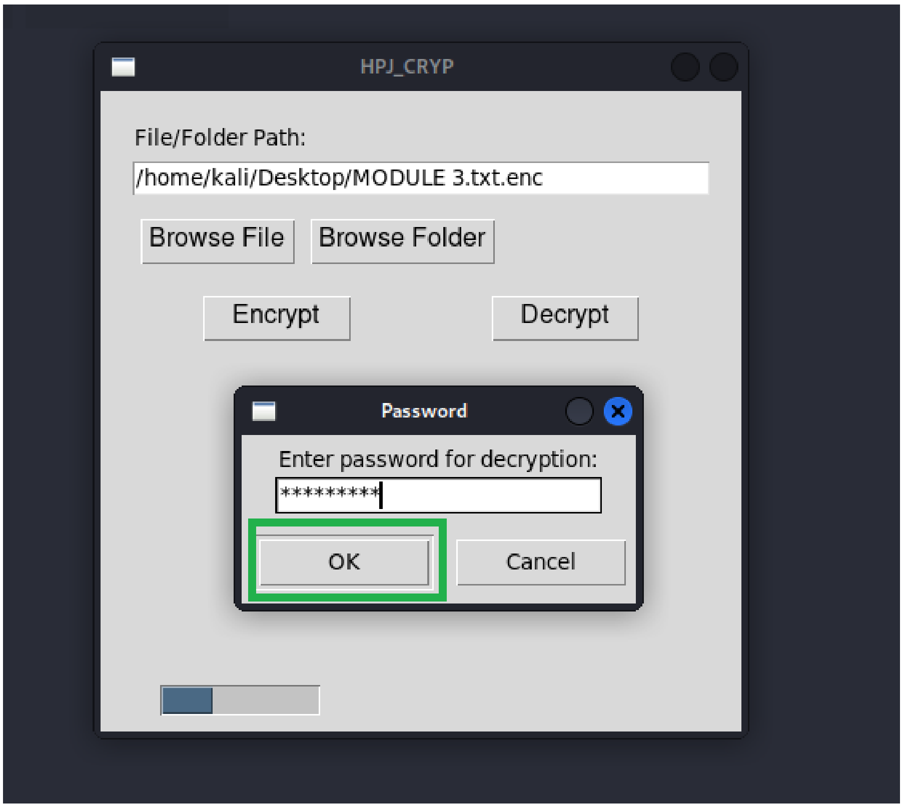
Step : After clicking on the **Decrypt** you will get a Dialog box for entering the password



File/Folder Path:
/home/kali/Desktop/MODULE 3.txt.enc

Enter password for decryption:

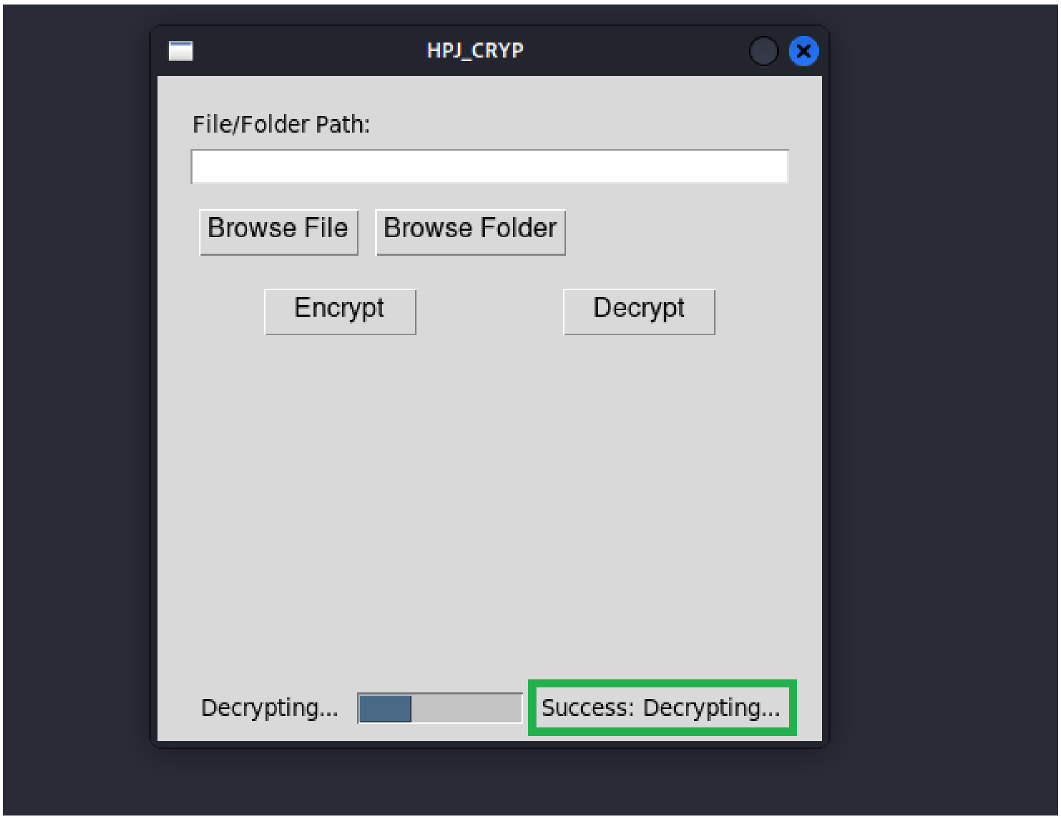
Step : Enter the password you entered to **Encrypt** the file and click OK.



File/Folder Path: /home/kali/Desktop/MODULE 3.txt.enc

Enter password for decryption: *********

Step : After clicking the ok you can see the success Decrypting of the File

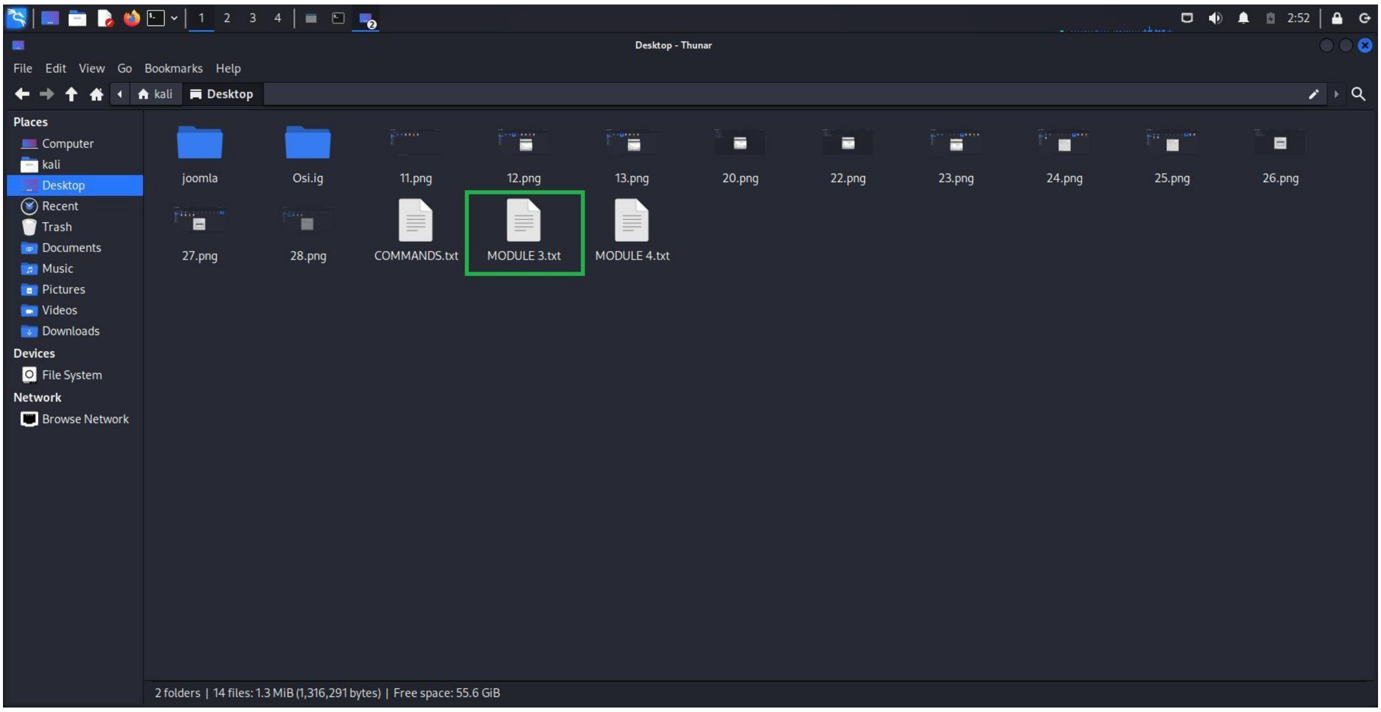


File/Folder Path:

Decrypting...

Success: Decrypting...

Step : You can observe the retrieved file by doing the Decryption process

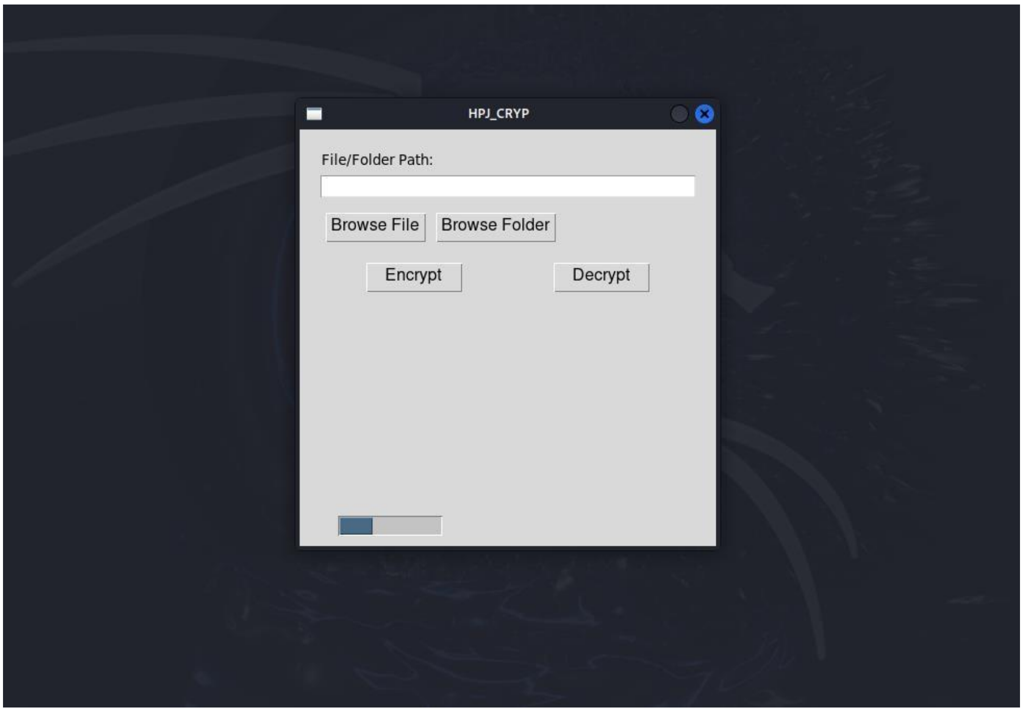


- ➢ Here is the completion of the Encryption and Decryption process of the Files

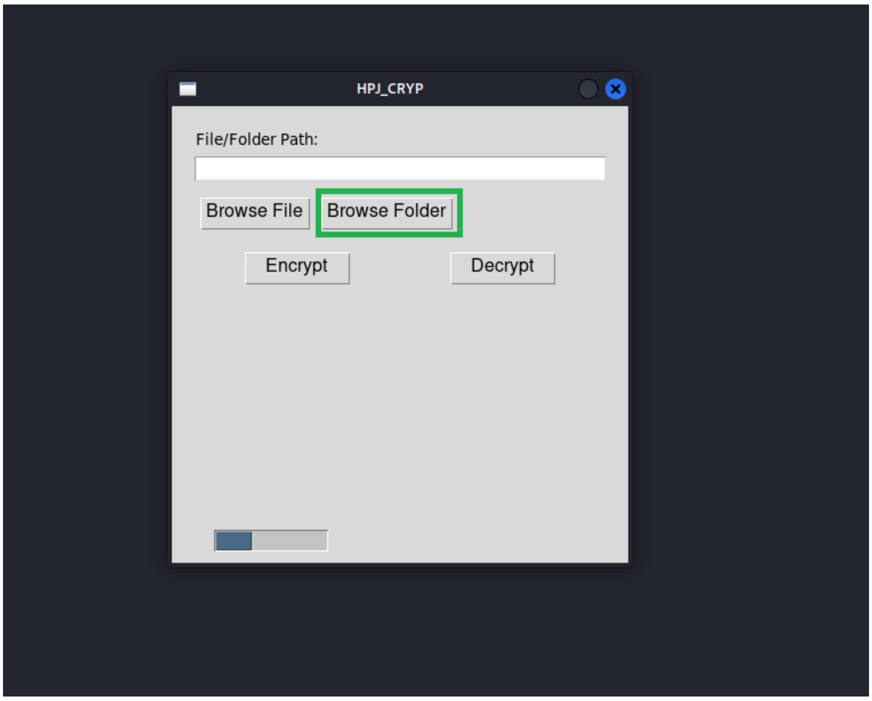
- ➢ We have witnessed the File **Encryption** and **Decryption** process. We can start the **Encryption** and **Decryption** procedure for the folders.

➢ Folder Encryption process:

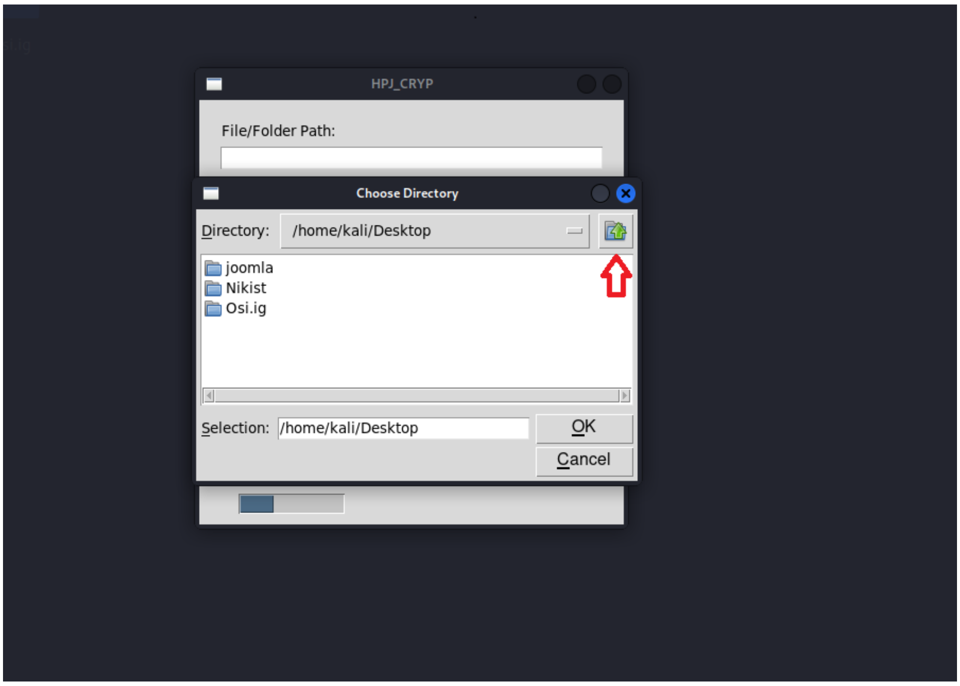
Step : Upon successful implementation of the Python program, the **HPJ_CRYP** Dialog box will display on the screen.



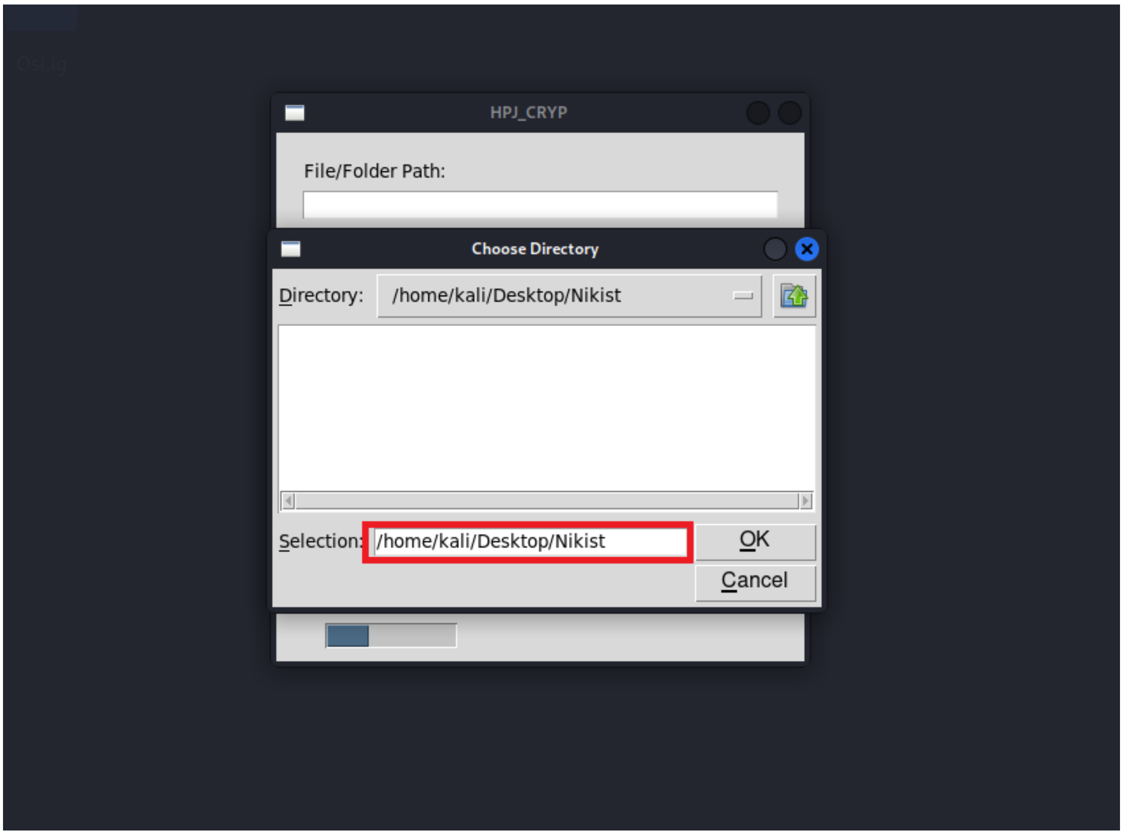
Step : Choose the Browse Folder option because we are doing the encryption for the folders.



Step : After choosing Browse Folder, we need to pick the Folder. This option allows us to perform a folder search.

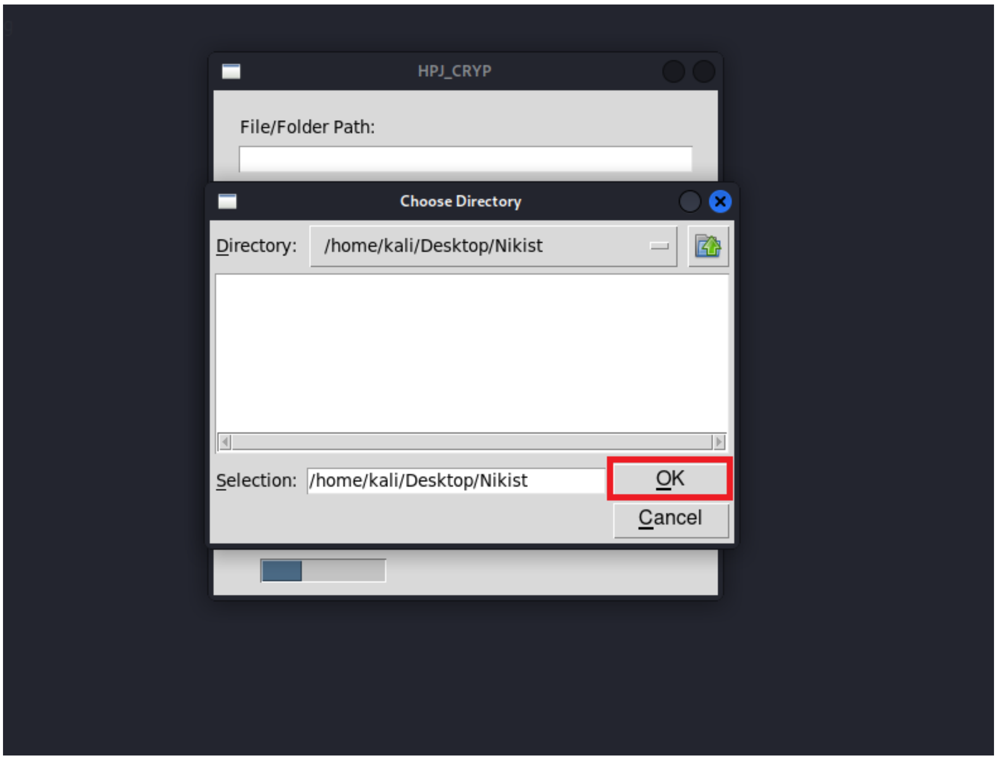


Step : Rather than choosing the folder, enter the Folder path in the selection



HPJ_CRYP
File/Folder Path:
Choose Directory
Directory: /home/kali/Desktop/Nikist
Selection: /home/kali/Desktop/Nikist
OK
Cancel

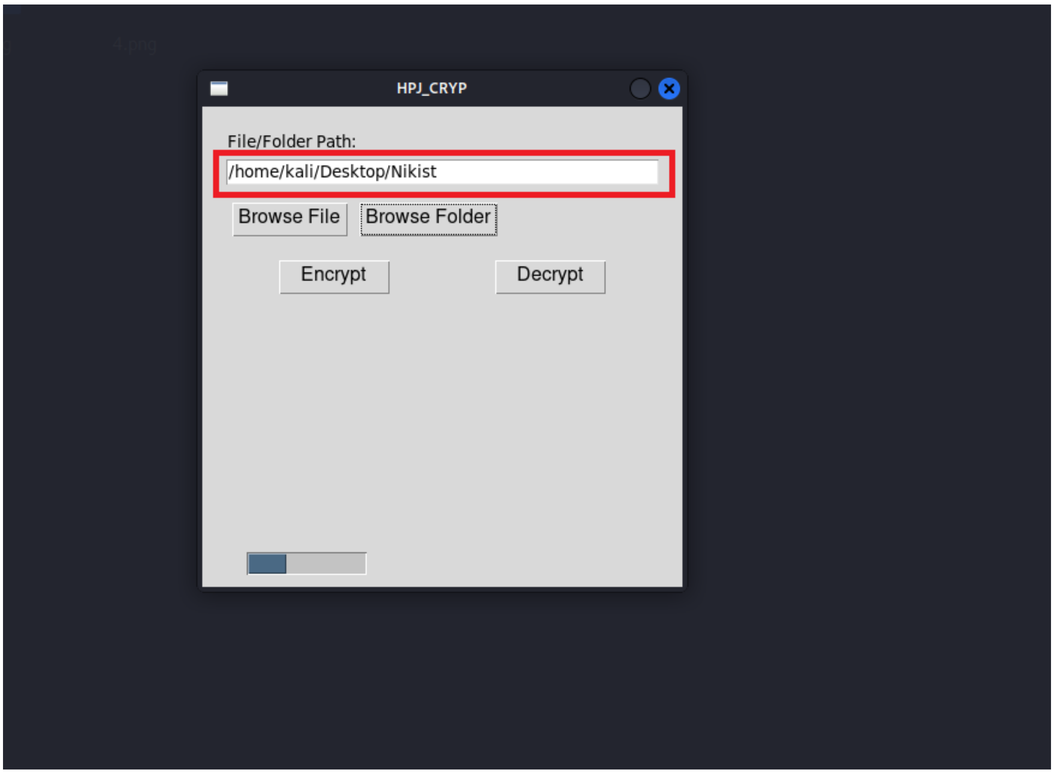
Step : After specifying the Folder path, press the OK button.



Directory: /home/kali/Desktop/Nikist

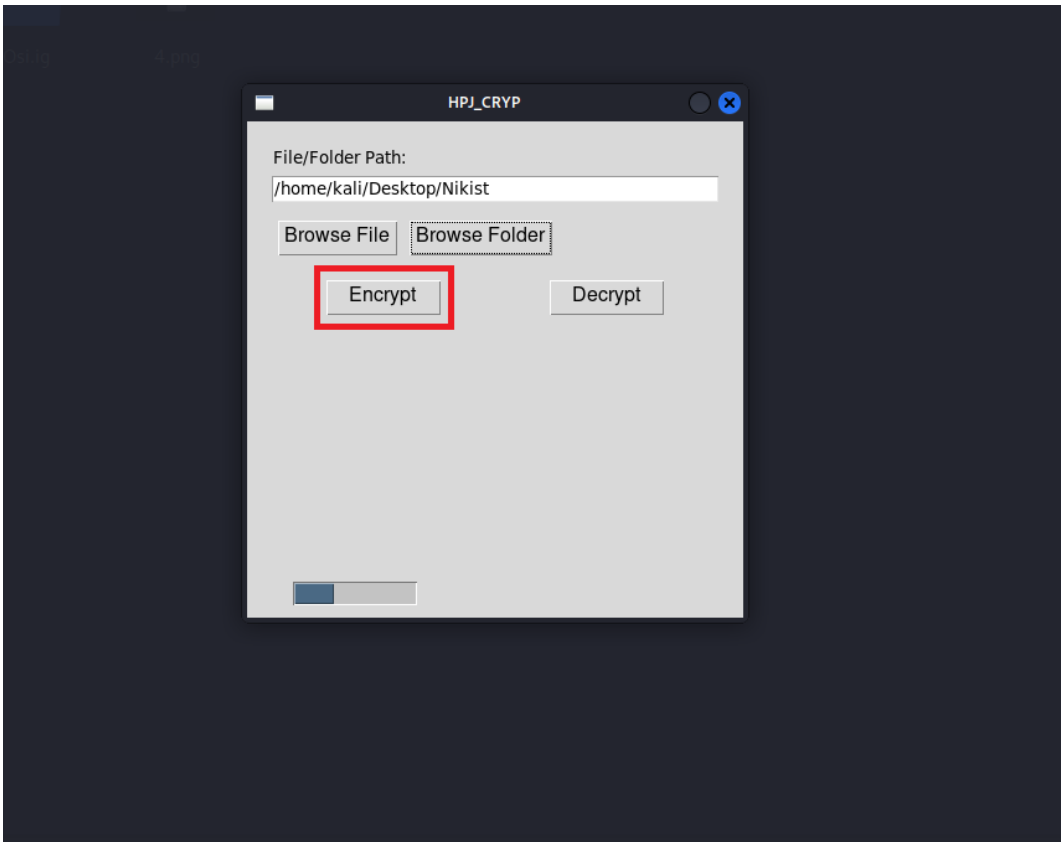
Selection: /home/kali/Desktop/Nikist

Step : After selecting the **OK** button, the Folder path appears in the **File/Folder path:**



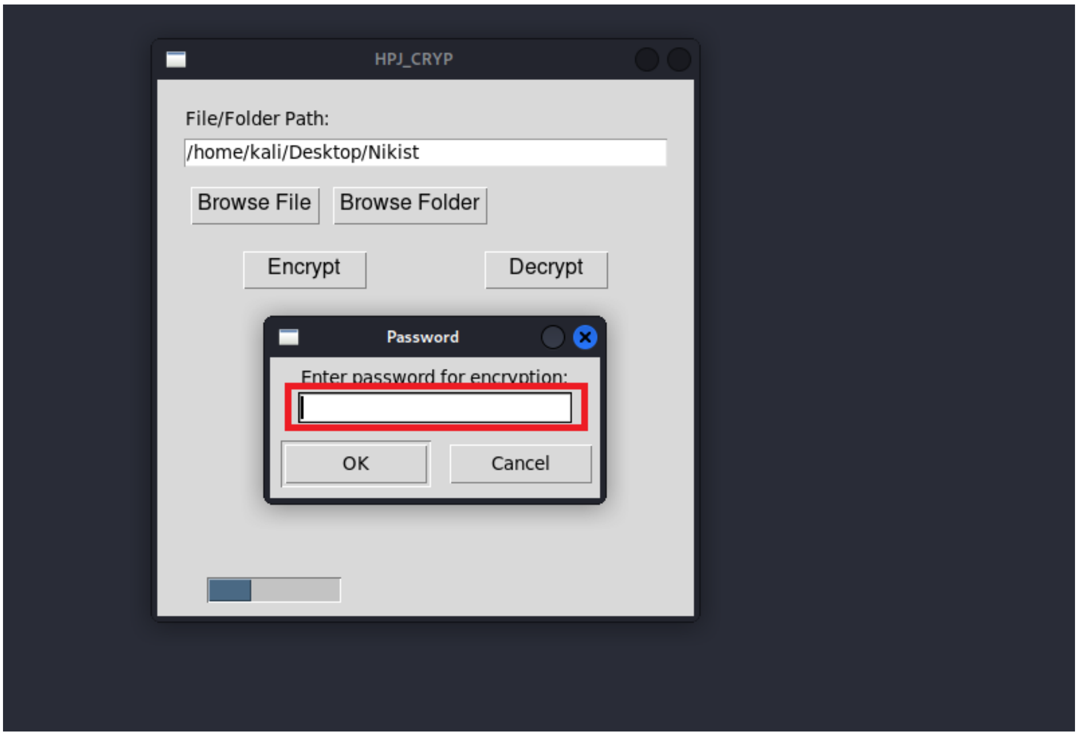
File/Folder Path: /home/kali/Desktop/Nikist

Step : To begin the folder's encryption procedure, we must first select the Encrypt option.



File/Folder Path: /home/kali/Desktop/Nikist

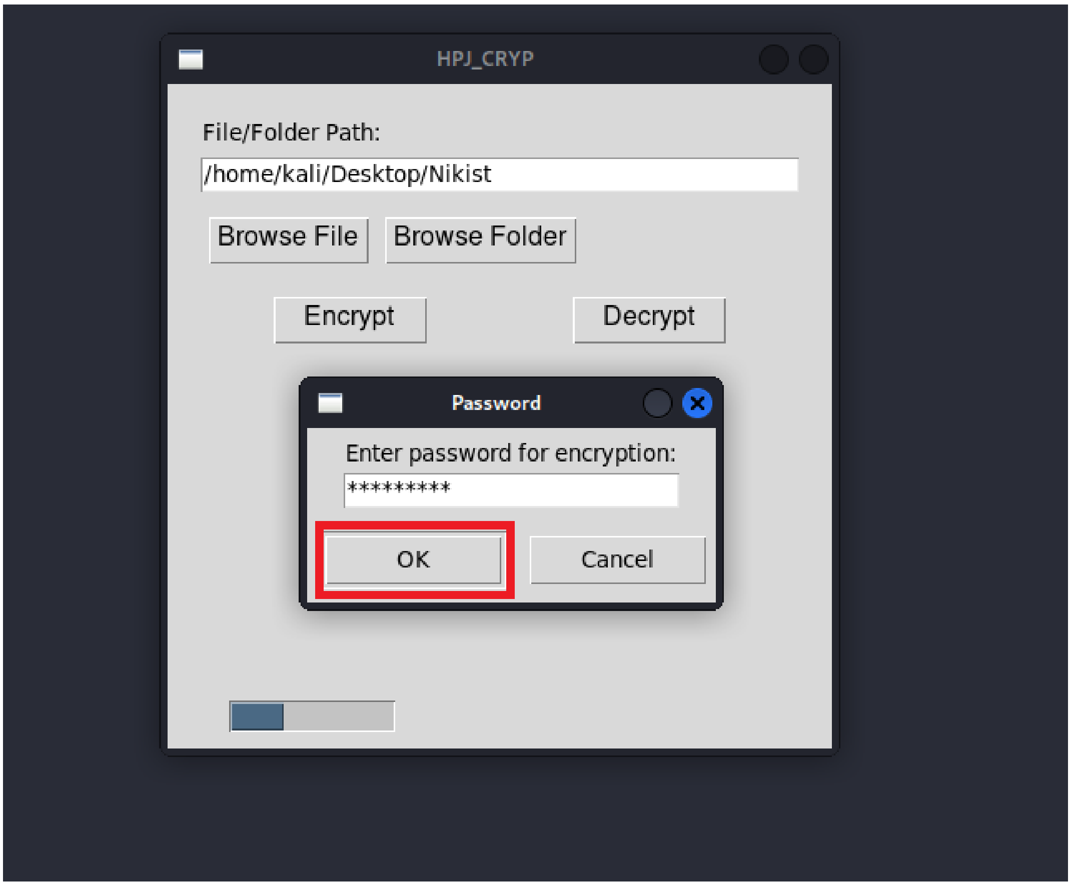
Step : After pressing the Encrypt button, it will ask for the password to enter.



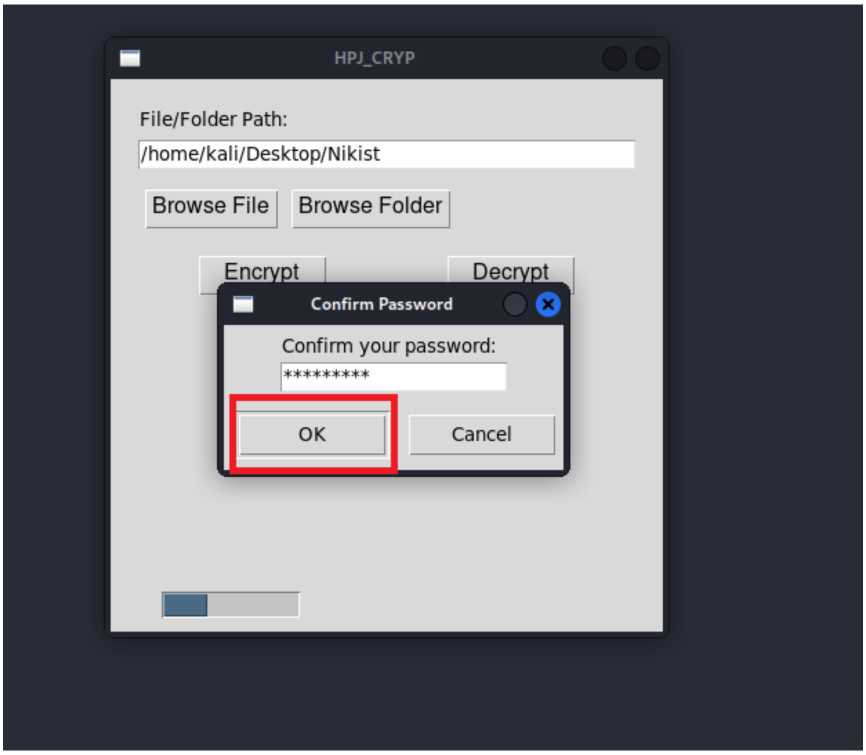
File/Folder Path: /home/kali/Desktop/Nikist

Enter password for encryption:

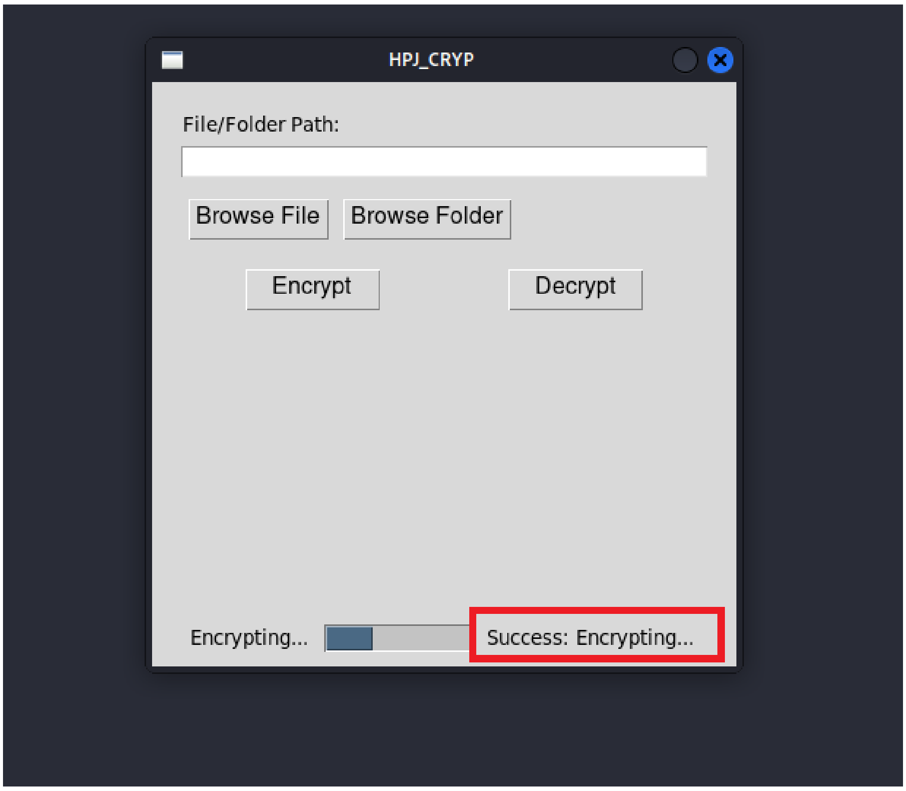
Step : After entering the password, press the OK button.



Step : After clicking the ok button, it will ask for the confirm password. Re-enter the password and click ok.



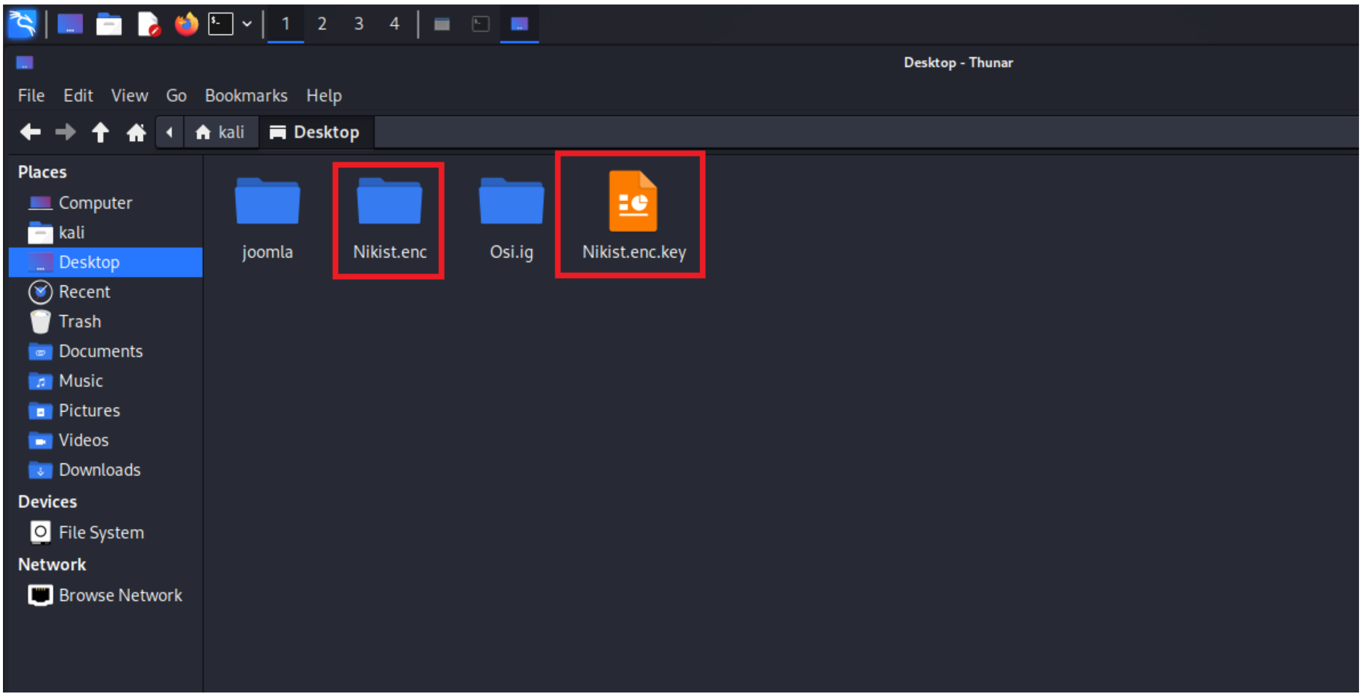
Step : After selecting OK, we can see the successful encryption of the folder.



Encrypting...

Success: Encrypting...

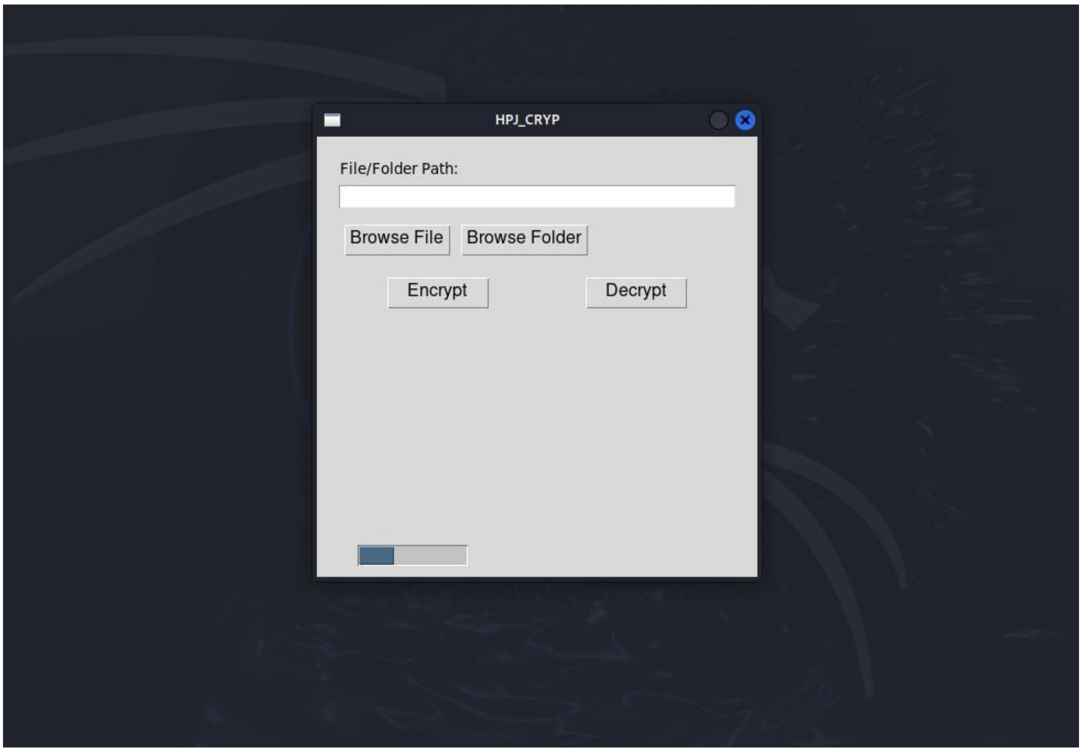
Step : We can see the encrypted folder and the key created for that file.



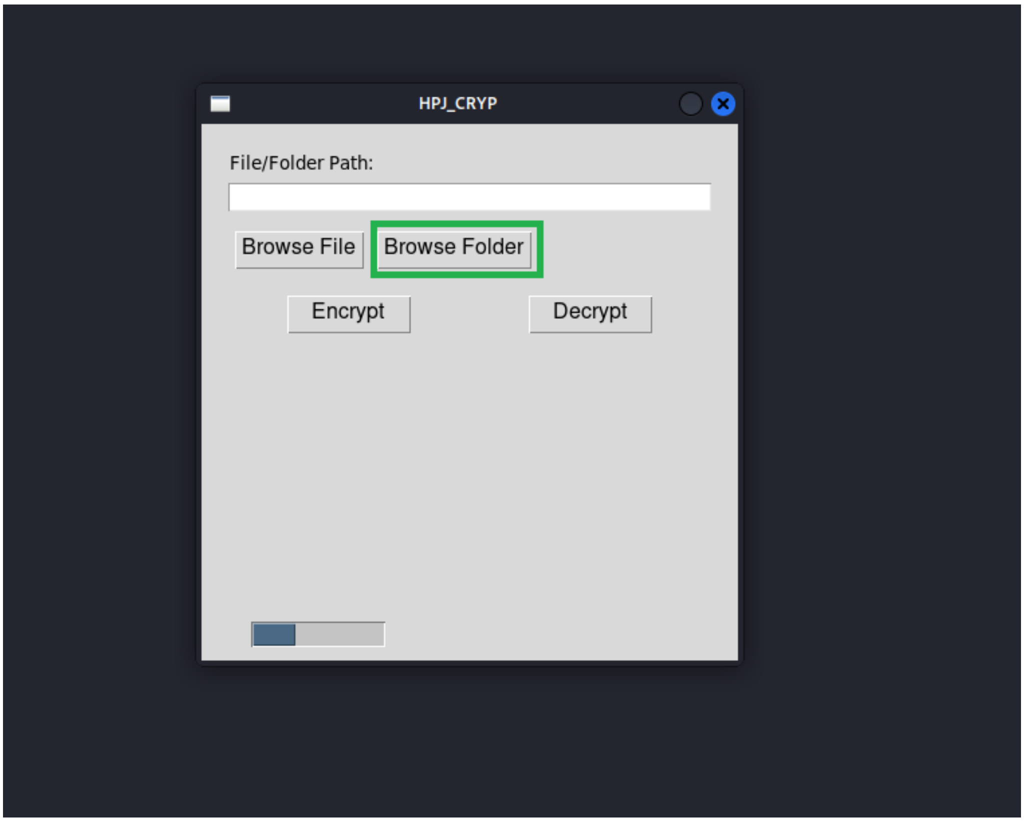
- As we have successfully encrypted the Folder, now if you want to retrieve back the Folder, we must do the decryption process for the Folder. Here is the decryption method

- **Folder Decryption Process :**

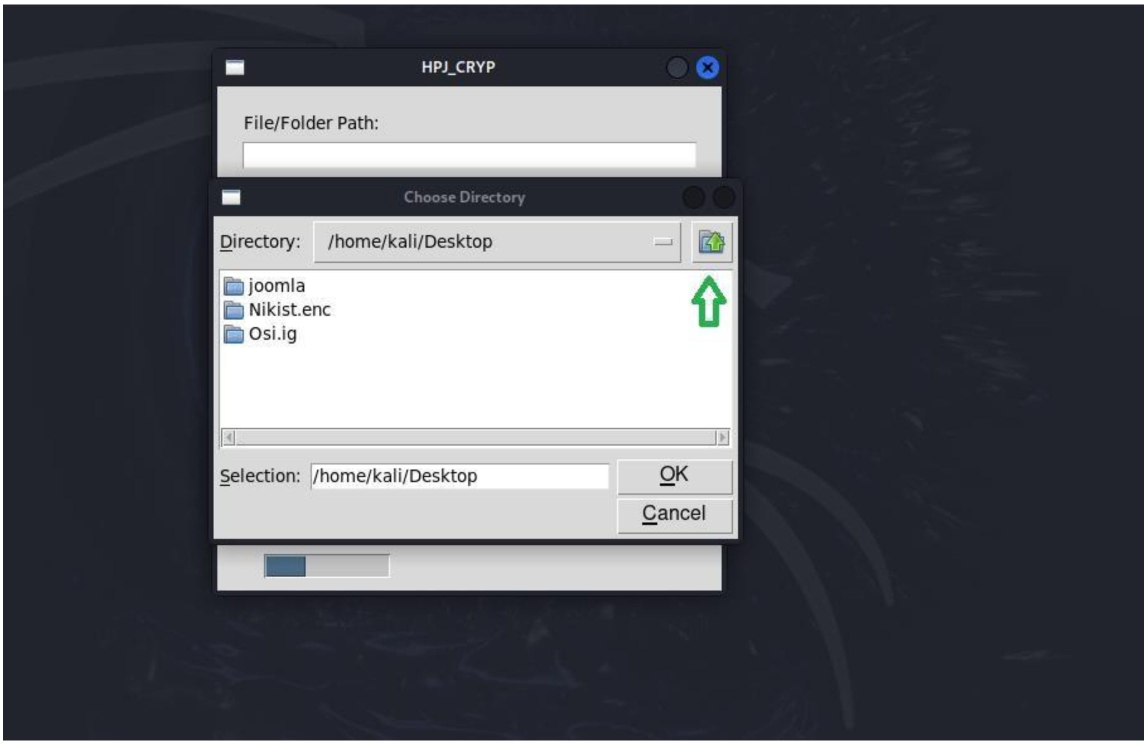
Step : Upon successful implementation of the Python program, the **HPJ_CRYP** Dialog box will display on the screen.



Step : Choose the Browse Folder option because we are doing the Decryption for the folders.



Step : After choosing Browse Folder, we need to pick the Folder which is encrypted. This option allows us to perform a folder search.

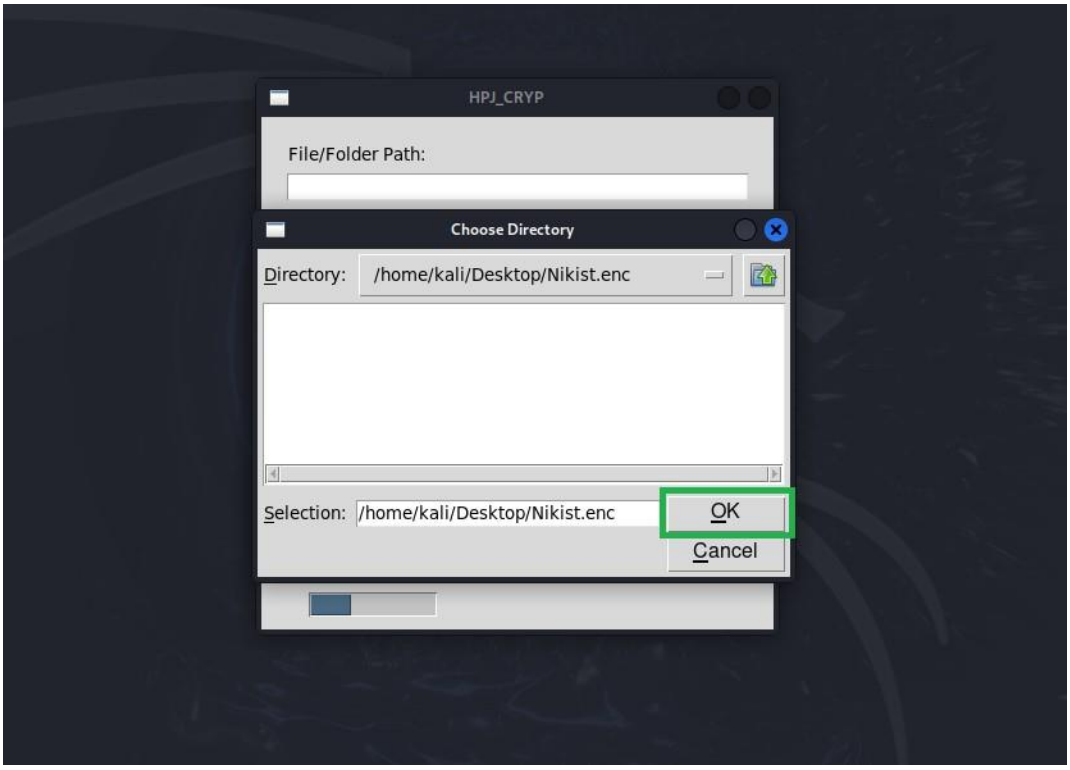


Directory: /home/kali/Desktop

joomla
Nikist.enc
Osi.ig

Selection: /home/kali/Desktop

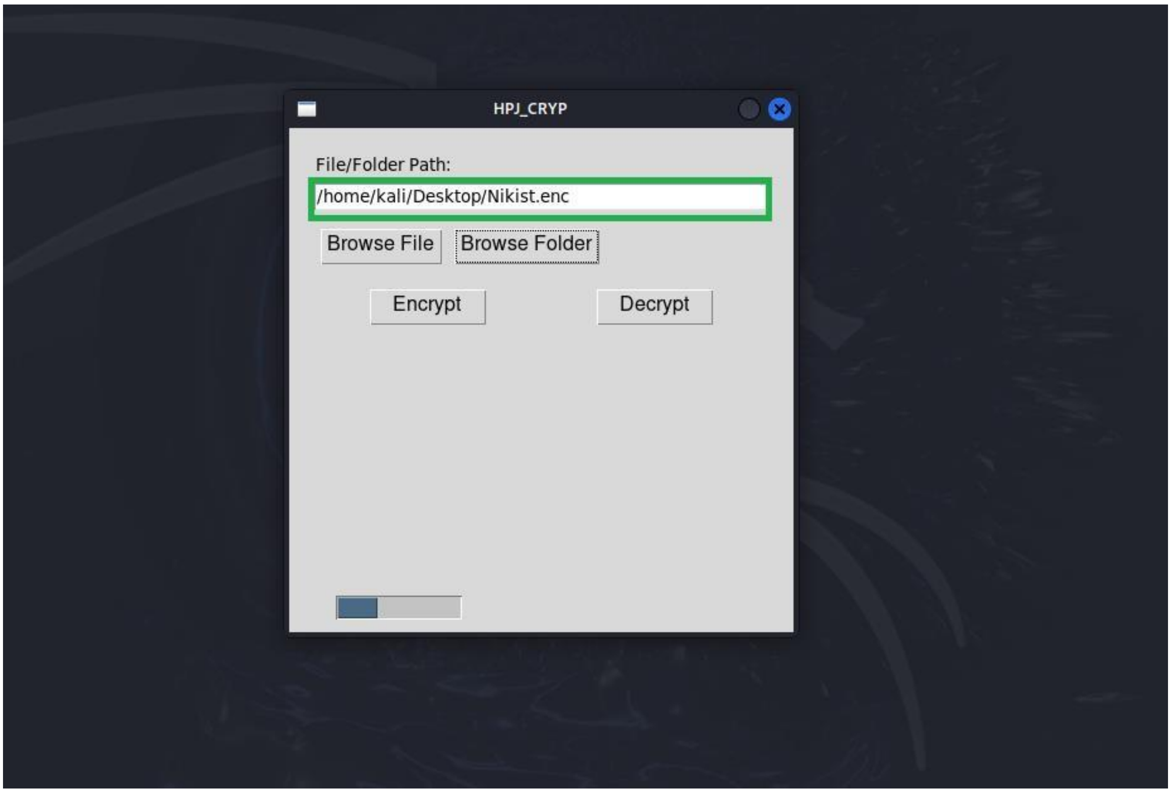
Step : Select the folder that should be Decrypted and press Ok



Directory: /home/kali/Desktop/Nikist.enc

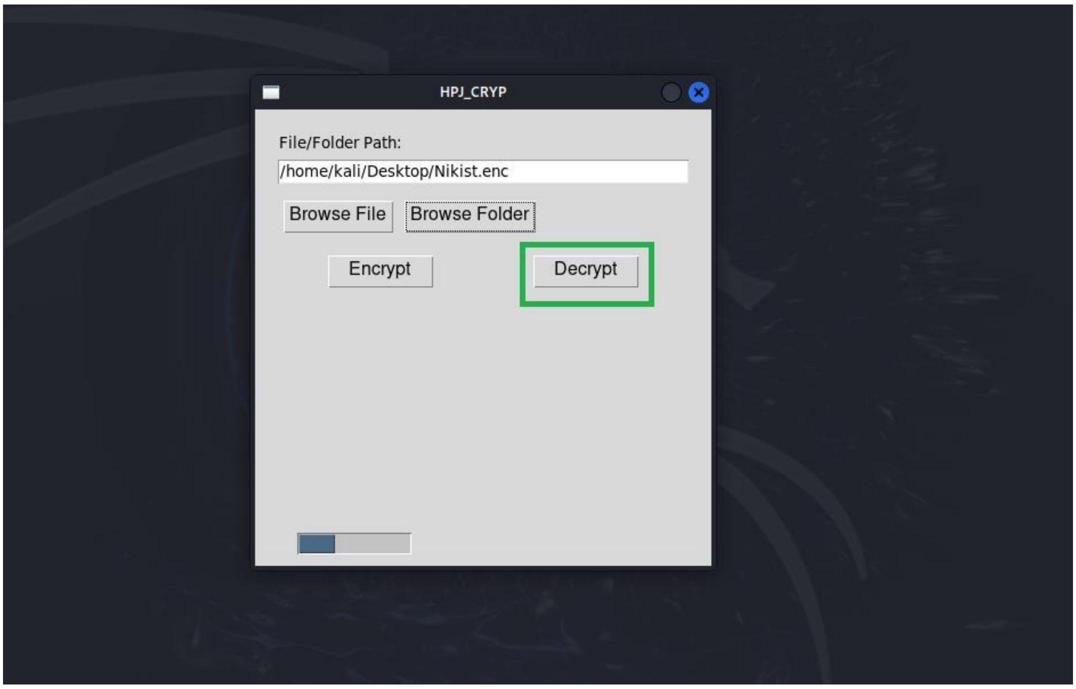
Selection: /home/kali/Desktop/Nikist.enc

Step : After selecting the OK button, you'll see the encrypted folder path.



File/Folder Path: /home/kali/Desktop/Nikist.enc

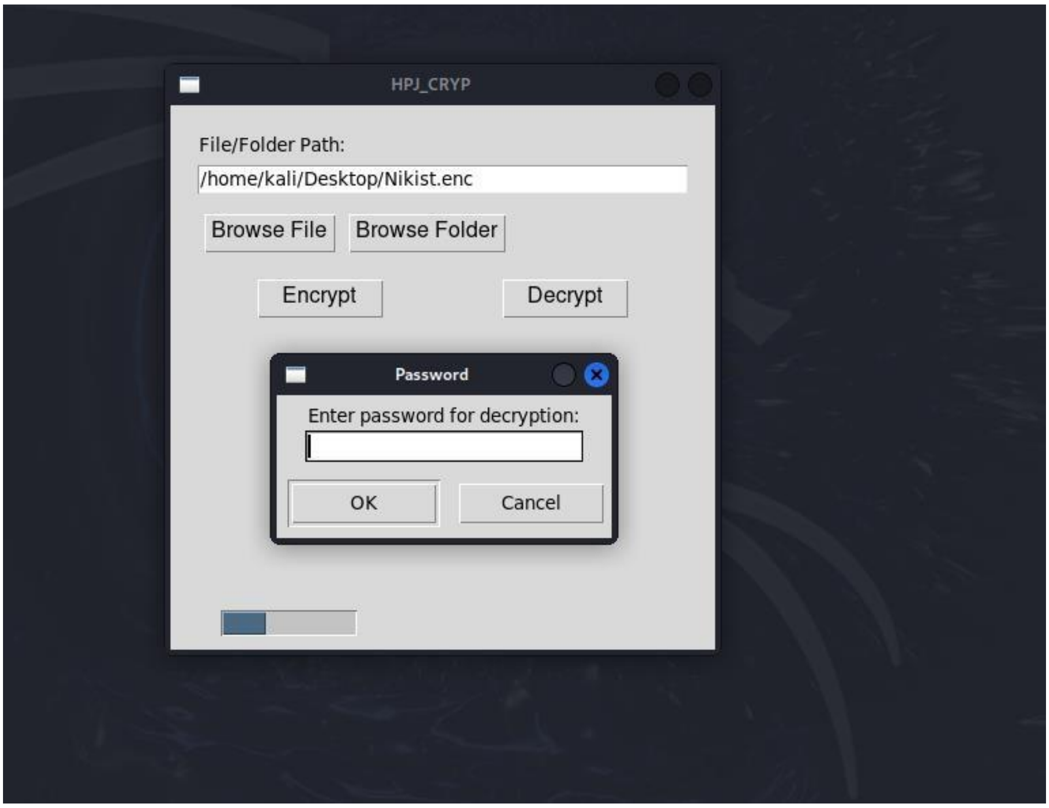
Step : Select the Decrypt option because we are **Decrypting** the folder.



HPJ_CRYP

File/Folder Path: /home/kali/Desktop/Nikist.enc

Step : After selecting Decrypt, it will ask for the password.



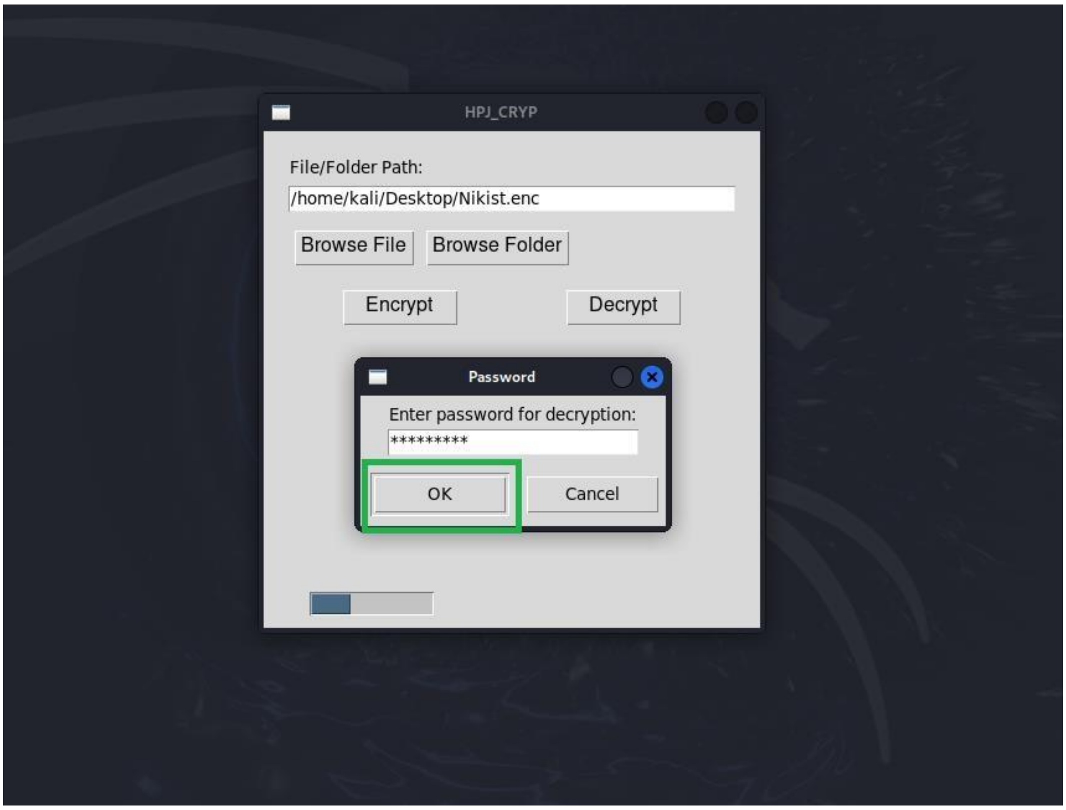
HPJ_CRYP

File/Folder Path: /home/kali/Desktop/Nikist.enc

Password

Enter password for decryption:

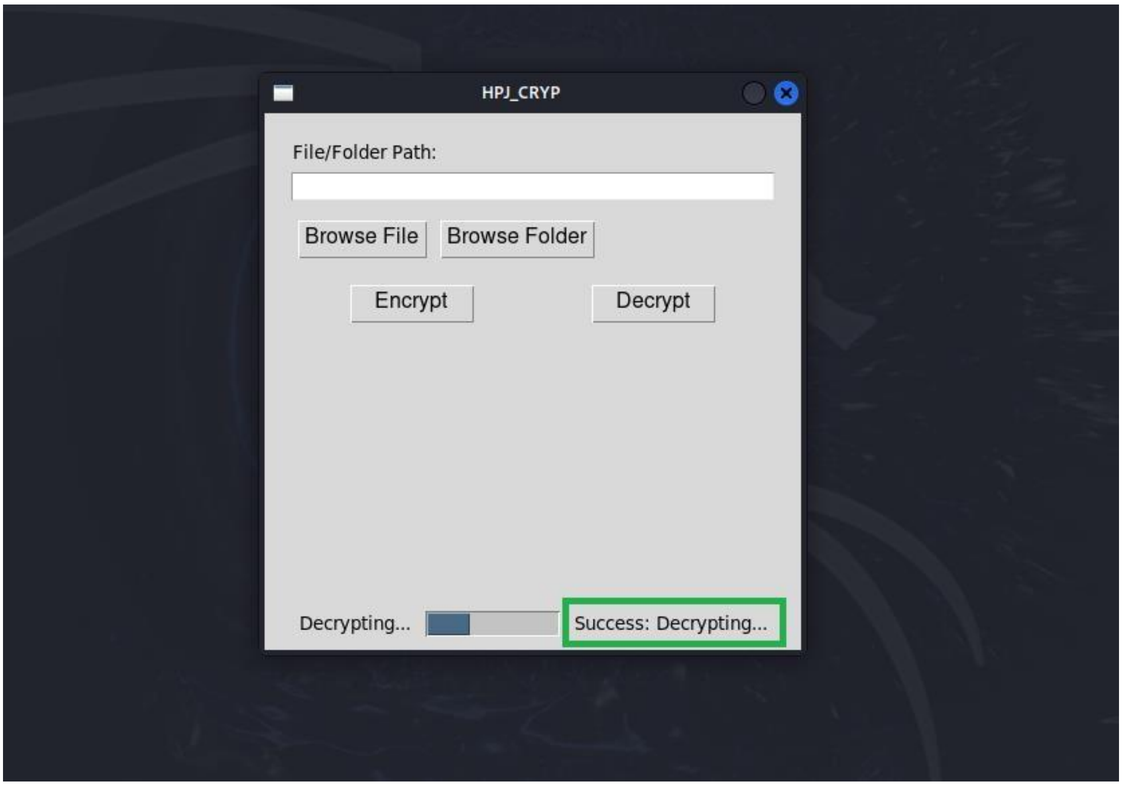
Step : Enter the password you entered during the encryption process and press OK.



File/Folder Path:
/home/kali/Desktop/Nikist.enc

Enter password for decryption:
****

Step : After selecting OK, you will see the successful decryption.

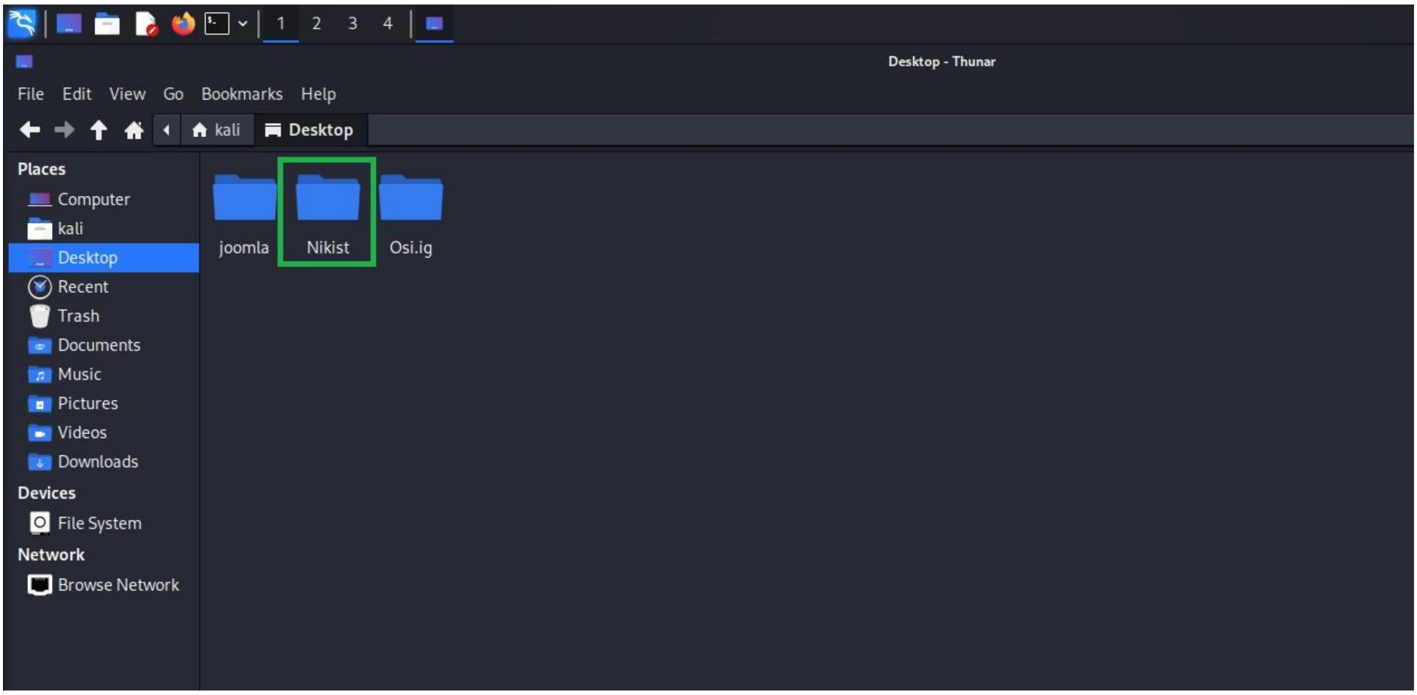


File/Folder Path:

Decrypting...

Success: Decrypting...

Step : We can see the Decrypted Folder after the decryption procedure.



- ➢ The folder decryption operation has been completed successfully.

- ➢ We have now successfully completed the encryption and decryption for the file and folder for the Protecting User Password Keys at Rest (on the disk).