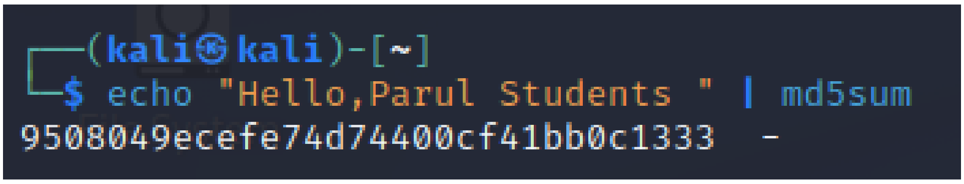
Cryptography

Cryptography is an integral part of cybersecurity, and Kali Linux provides a variety of tools and frameworks for cryptographic operations. Here’s a practical guide to cryptography tasks in Kali Linux, including basic encryption, decryption, hashing, and more advanced techniques:

1. Hashing

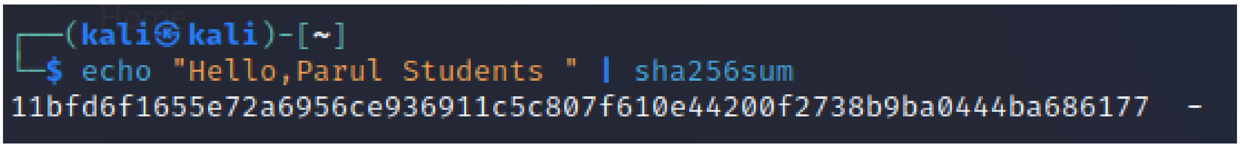
Hashing ensures data integrity by generating a fixed-size string (hash) for any input. Tools like md5sum, sha256sum, or openssl are commonly used.

Generate an MD5 hash



```
┌──(kali㉿kali)-[~]
└─$ echo "Hello,Parul Students " | md5sum
9508049ecefe74d74400cf41bb0c1333  -
```

Generate a SHA-256 hash



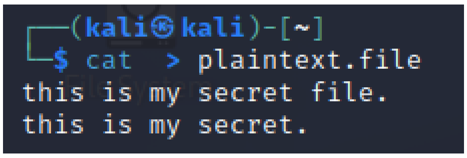
```
┌──(kali㉿kali)-[~]
└─$ echo "Hello,Parul Students " | sha256sum
11bfd6f1655e72a6956ce936911c5c807f610e44200f2738b9ba0444ba686177  -
```

2. File Encryption and Decryption Using OpenSSL

openssl is a command-line tool and library widely used for managing SSL/TLS certificates, encryption,

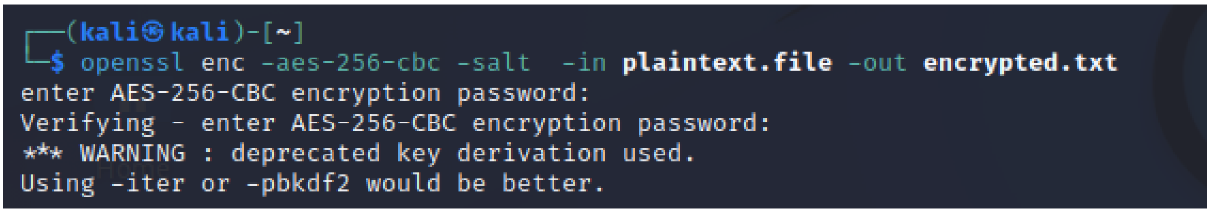
decryption, and cryptographic operations. Below is an overview of the key functionalities of the openssl command, based on its man page.

- ## Step-1 create the plain text file for encryption and decryption



```
┌──(kali㉿kali)-[~]
└─$ cat  > plaintext.file
this is my secret file.
this is my secret.
```

- ## Step-2 Encrypt a file using a symmetric algorithm (e.g., AES-256-CBC)



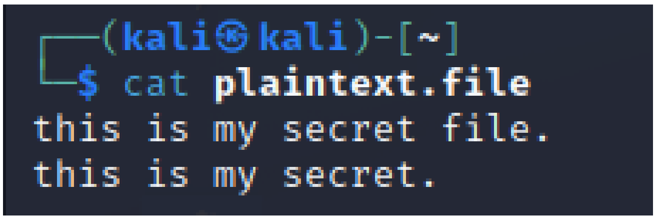
```
┌──(kali㉿kali)-[~]
└─$ openssl enc -aes-256-cbc -salt  -in plaintext.file -out encrypted.txt
enter AES-256-CBC encryption password:
Verifying - enter AES-256-CBC encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

[note: set encryption password:12345]

- `-aes-256-cbc` : Specifies the encryption algorithm.
- `-salt` : Adds random data to strengthen encryption.
- `-in` : Input file.
- `-out` : Output encrypted file.
- You'll be prompted to enter a password.

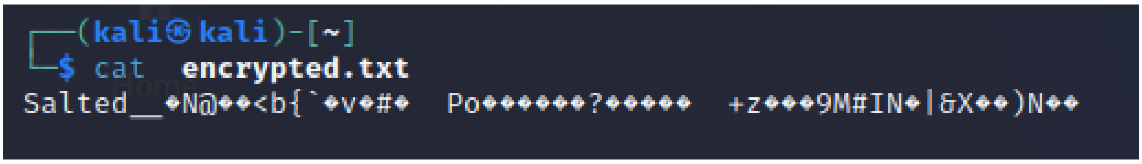
➢ Step-3 See the plain text file and encrypted file:

plain text file:



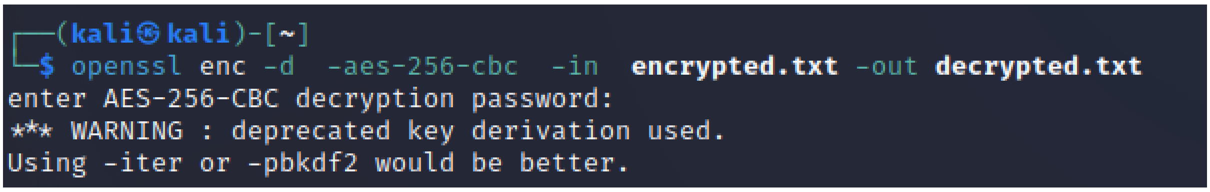
```
┌──(kali㉿kali)-[~]
└─$ cat plaintext.file
this is my secret file.
this is my secret.
```

encrypted file:



```
┌──(kali㉿kali)-[~]
└─$ cat  encrypted.txt
Salted__◆N@◆◆<b{`◆v◆#◆  Po◆◆◆◆◆◆◆?◆◆◆◆◆◆  +z◆◆◆◆9M#IN◆|&X◆◆)N◆◆
```

➢ Step-4 Decrypt a file using OpenSSL

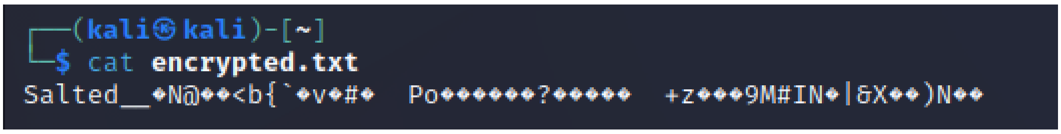


```
┌──(kali㉿kali)-[~]
└─$ openssl enc -d  -aes-256-cbc  -in  encrypted.txt -out decrypted.txt
enter AES-256-CBC decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

[note: set decryption password:12345]

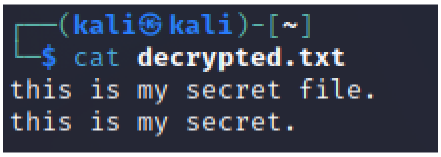
➢ Step-5 See the encrypted file and decrypted file:

encrypted file:



```
┌──(kali㉿kali)-[~]
└─$ cat encrypted.txt
Salted__◆N@◆◆<b{`◆v◆#◆  Po◆◆◆◆◆◆◆?◆◆◆◆◆◆  +z◆◆◆◆9M#IN◆|&X◆◆)N◆◆
```

decrypted file:



```
┌──(kali㉿kali)-[~]
└─$ cat decrypted.txt
this is my secret file.
this is my secret.
```