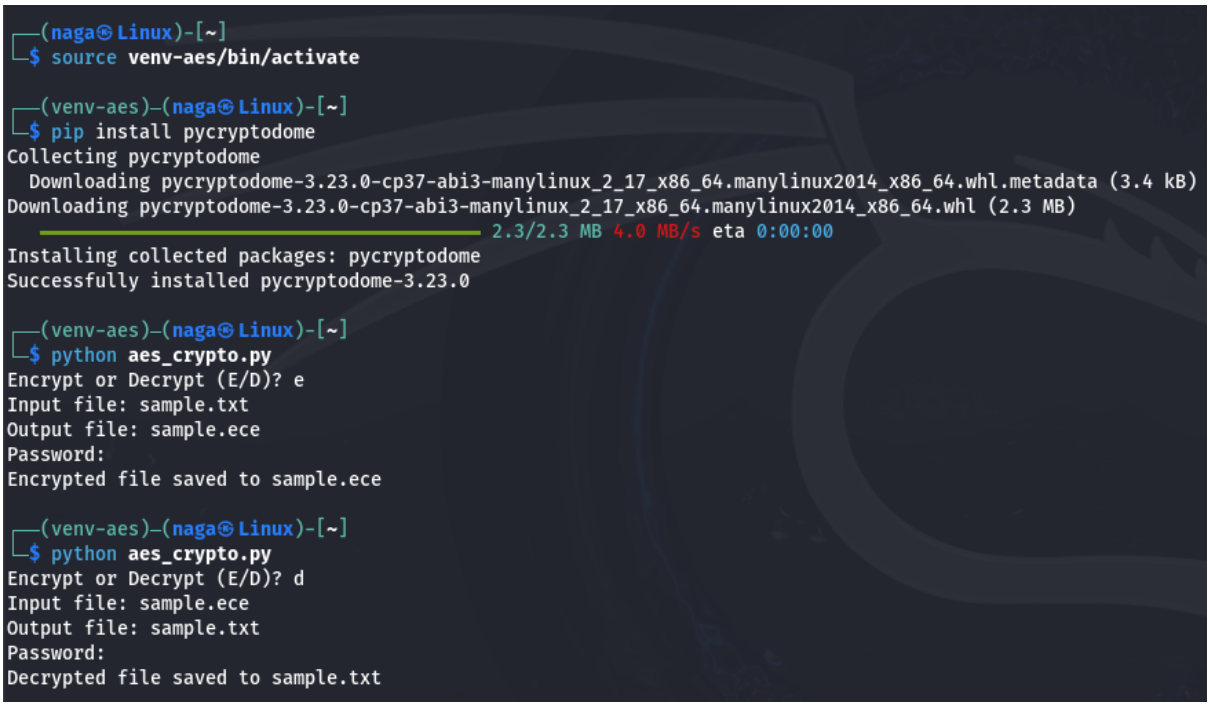
ADVANCED_ENCRYPTION_TOOL

```
import os
import base64
from Crypto.Cipher import AES
from Crypto.Protocol.KDF import PBKDF2
from Crypto.Random import get_random_bytes
from getpass import getpass
BLOCK_SIZE = 16
KEY_SIZE = 32
SALT_SIZE = 16
IV_SIZE = 16
ITERATIONS = 100_000
def pad(data):
    padding_len = BLOCK_SIZE - len(data) % BLOCK_SIZE
   return data + bytes([padding_len] * padding_len)
def unpad(data):
    return data[:-data[-1]]
def derive_key(password, salt):
    return PBKDF2(password, salt, dkLen=KEY_SIZE, count=ITERATIONS)
def encrypt_file(input_file, output_file, password):
    salt = get_random_bytes(SALT_SIZE)
    iv = get_random_bytes(IV_SIZE)
    key = derive_key(password.encode(), salt)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    with open(input_file, 'rb') as f:
        plaintext = f.read()
    ciphertext = cipher.encrypt(pad(plaintext))
    with open(output_file, 'wb') as f:
        f.write(salt + iv + ciphertext)
def decrypt_file(input_file, output_file, password):
    with open(input_file, 'rb') as f:
        salt = f.read(SALT_SIZE)
        iv = f.read(IV_SIZE)
        ciphertext = f.read()
    key = derive_key(password.encode(), salt)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    plaintext = unpad(cipher.decrypt(ciphertext))
    with open(output_file, 'wb') as f:
        f.write(plaintext)
def main():
    choice = input("Encrypt or Decrypt (E/D)? ").strip().upper()
    input_file = input("Input file: ")
    output_file = input("Output file: ")
    password = getpass("Password: ")
    if choice == 'E':
        encrypt_file(input_file, output_file, password)
        print(f"Encrypted file saved to {output_file}")
    elif choice == 'D':
        decrypt_file(input_file, output_file, password)
        print(f"Decrypted file saved to {output_file}")
    else:
        print("Invalid choice.")
if __name__ == "__main__":
    main()
```

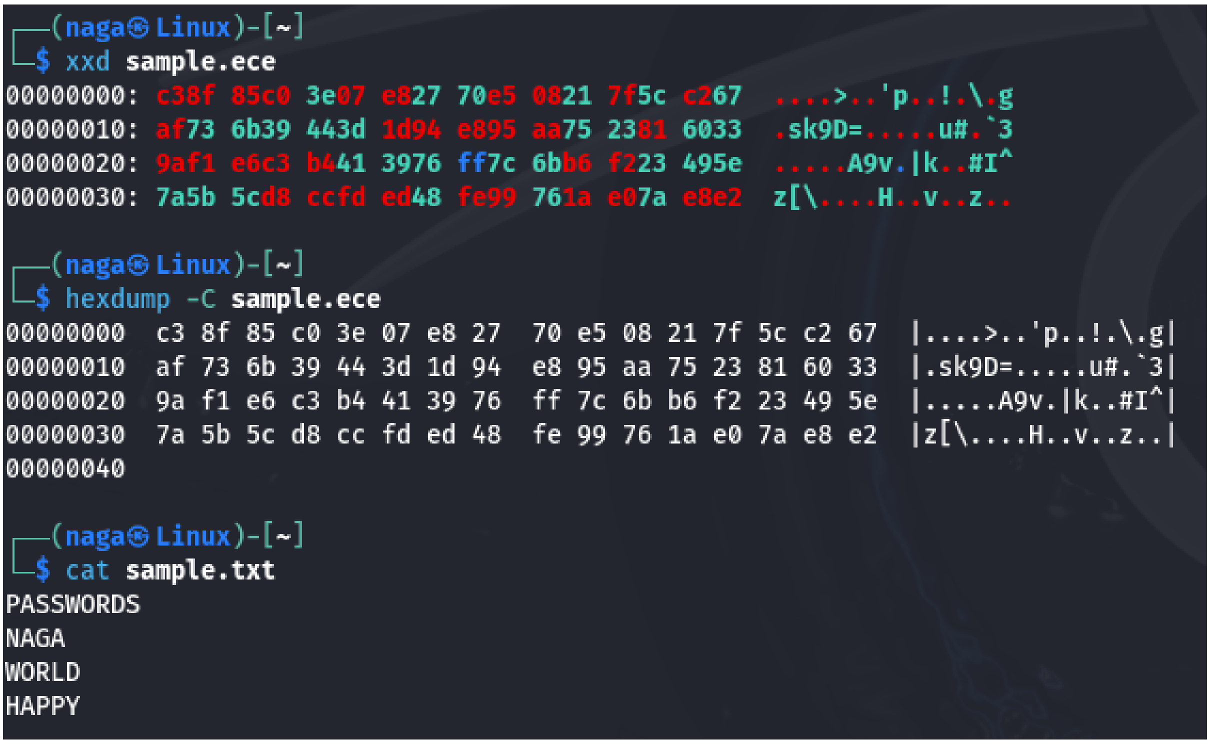


```
┌──(naga㉿Linux)-[~]
└─$ source venv-aes/bin/activate

┌──(venv-aes)─(naga㉿Linux)-[~]
└─$ pip install pycryptodome
Collecting pycryptodome
  Downloading pycryptodome-3.23.0-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.4 kB)
Downloading pycryptodome-3.23.0-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.3/2.3 MB 4.0 MB/s eta 0:00:00
Installing collected packages: pycryptodome
Successfully installed pycryptodome-3.23.0

┌──(venv-aes)─(naga㉿Linux)-[~]
└─$ python aes_crypto.py
Encrypt or Decrypt (E/D)? e
Input file: sample.txt
Output file: sample.ece
Password:
Encrypted file saved to sample.ece

┌──(venv-aes)─(naga㉿Linux)-[~]
└─$ python aes_crypto.py
Encrypt or Decrypt (E/D)? d
Input file: sample.ece
Output file: sample.txt
Password:
Decrypted file saved to sample.txt
```



```
┌──(naga㉿Linux)-[~]
└─$ xxd sample.ece
00000000: c38f 85c0 3e07 e827 70e5 0821 7f5c c267  ....>..'p..!.\.g
00000010: af73 6b39 443d 1d94 e895 aa75 2381 6033  .sk9D=.....u#.`3
00000020: 9af1 e6c3 b441 3976 ff7c 6bb6 f223 495e  .....A9v.|k..#I^
00000030: 7a5b 5cd8 ccfd ed48 fe99 761a e07a e8e2  z[\....H..v..z..

┌──(naga㉿Linux)-[~]
└─$ hexdump -C sample.ece
00000000  c3 8f 85 c0 3e 07 e8 27  70 e5 08 21 7f 5c c2 67  |....>..'p..!.\.g|
00000010  af 73 6b 39 44 3d 1d 94  e8 95 aa 75 23 81 60 33  |.sk9D=.....u#.`3|
00000020  9a f1 e6 c3 b4 41 39 76  ff 7c 6b b6 f2 23 49 5e  |.....A9v.|k..#I^|
00000030  7a 5b 5c d8 cc fd ed 48  fe 99 76 1a e0 7a e8 e2  |z[\....H..v..z..|
00000040

┌──(naga㉿Linux)-[~]
└─$ cat sample.txt
PASSWORDS
NAGA
WORLD
HAPPY
```