Kevin Huang
CSS 577
Encryption Assignment
04/26/2022

Encryption Assignment

Objective

The goal of this assignment is to create a file encryption/decryption utility program. Users will run the program through the command line by inputting the appropriate parameters for encryption or decryption.

Specification

For encryption, the program will derive a master key from a provided password and a randomly generated salt. The PBKDF2 function from the OpenSSL library is used for the master key derivation. The program currently supports SHA256 and SHA512 hashes. The same hashing algorithm will be used for both KDF and HMAC. The encryption and HMAC keys are then derived from the master key.

The inputted data is encrypted using CBC chaining mode using the OpenSSL EVP interface. The program supports 3DES, AES128, and AES255 encryption algorithms. It will use a randomly generated IV of one block size depending on the encryption algorithm that is selected. An HMAC of the IV and ciphertext will then be created with the same hashing algorithm. The metadata, HMAC, IV, and ciphertext is then written to an output file.

For decryption, the metadata is extracted from the input file to determine the salt, hashing algorithm, and encryption algorithm. The program will derive a master key, encryption key, and HMAC key with the same method as in the encryption process. The HMAC is then calculated and then compared to the HMAC in the input file. The ciphertext will be decrypted with the encryption algorithm in the metadata and written to a specified output file.

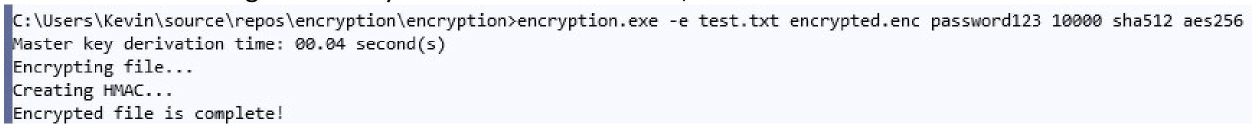
Usage

- Compile encryption.cpp: 'g++ -Wall encryption.cpp -o encryption -lcrypto'
- Encryption: 'progname -e inputfile outputfile password iterations hashalgo encryptalgo'
  Example: 'encryption.exe -e input.txt encrypted.enc password123 1000 sha512 aes256'
- Decryption: 'progname -d inputfile outputfile password'
  Example: 'encryption.exe -d input.enc decrypted.txt password123'

Tests

One aspect of the program that I tested was the amount of iterations needed for the master key derivation. I decided that a derivation time of 0.25 seconds is an acceptable delay for general users. The results of my tests are reported in the table below. An iteration count of 60,000 would be acceptable.

Screenshot 1: Testing master key derivation time with 10,000 iterations



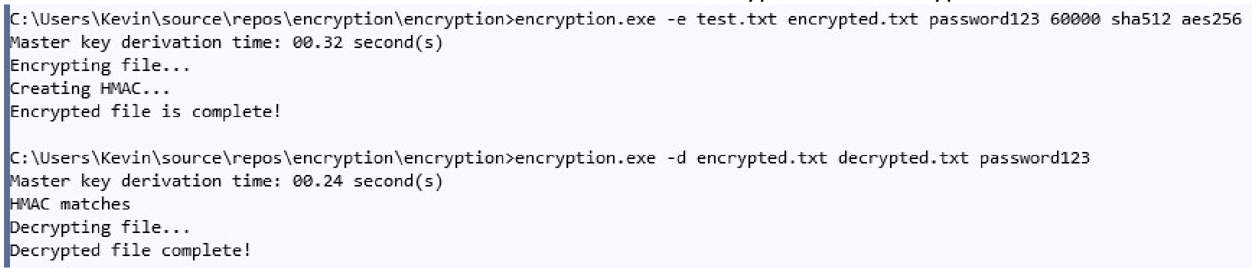
```
C:\Users\Kevin\source\repos\encryption\encryption>encryption.exe -e test.txt encrypted.enc password123 10000 sha512 aes256
Master key derivation time: 00.04 second(s)
Encrypting file...
Creating HMAC...
Encrypted file is complete!
```

Table 1:

Iterations	Trial 1	Trial 2	Trial 3	Average
10,000	0.04 seconds	0.04 seconds	0.04 seconds	0.043 seconds
50,000	0.21 seconds	0.20 seconds	0.20 seconds	0.203 seconds
60,000	0.24 seconds	0.30 seconds	0.24 seconds	0.260 seconds
100,000	0.41 seconds	0.41 seconds	0.40 seconds	0.407 seconds

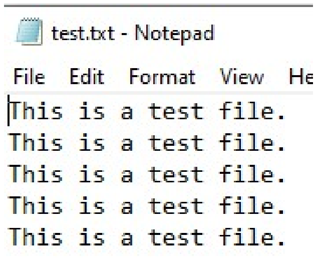
Another aspect of the program that I tested was making sure encryption and decryption was successful for the following hashing algorithm/encryption algorithm pairs: SHA256 – 3DES, SHA256 – AES128, and SHA512 – AES256. The successful runs are documented in the screenshots below.

Screenshot 2: SHA512 – AES256 command line successful encryption and decryption



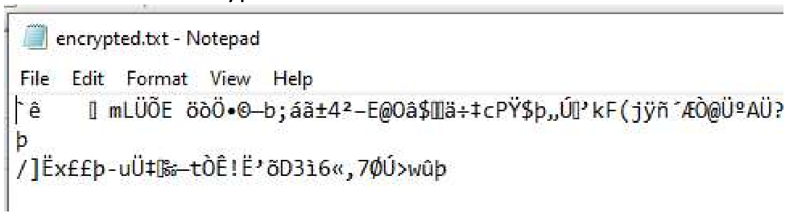
```
C:\Users\Kevin\source\repos\encryption\encryption>encryption.exe -e test.txt encrypted.txt password123 60000 sha512 aes256
Master key derivation time: 00.32 second(s)
Encrypting file...
Creating HMAC...
Encrypted file is complete!

C:\Users\Kevin\source\repos\encryption\encryption>encryption.exe -d encrypted.txt decrypted.txt password123
Master key derivation time: 00.24 second(s)
HMAC matches
Decrypting file...
Decrypted file complete!
```

Screenshot 3: test.txt



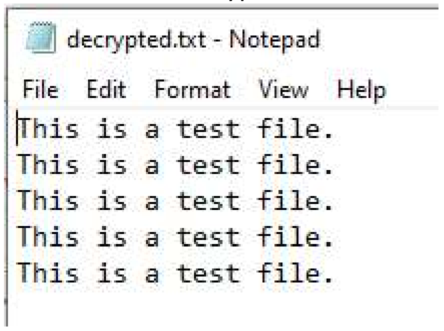
```
This is a test file.
This is a test file.
This is a test file.
This is a test file.
This is a test file.
```

Screenshot 4: encrypted.txt



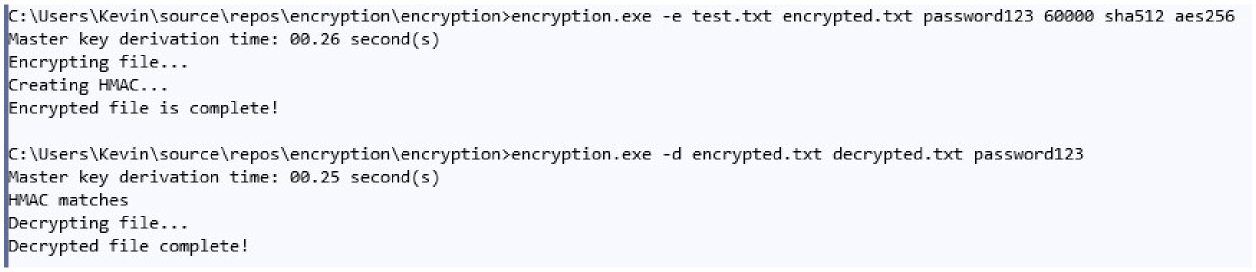
```
ˆê     mLÜÕE öòÖ•©–b;áã±4²–E@Oâ$ä÷‡cPŸ$þ„Ú’kF(jÿñ´ÆÒ@ÜºAÜ?
þ
/]Ëx££þ-uÜ‡‰–tÒÊ!Ë’õD3ì6«,7ØÚ>wûþ
```

Screenshot 5: decrypted.txt



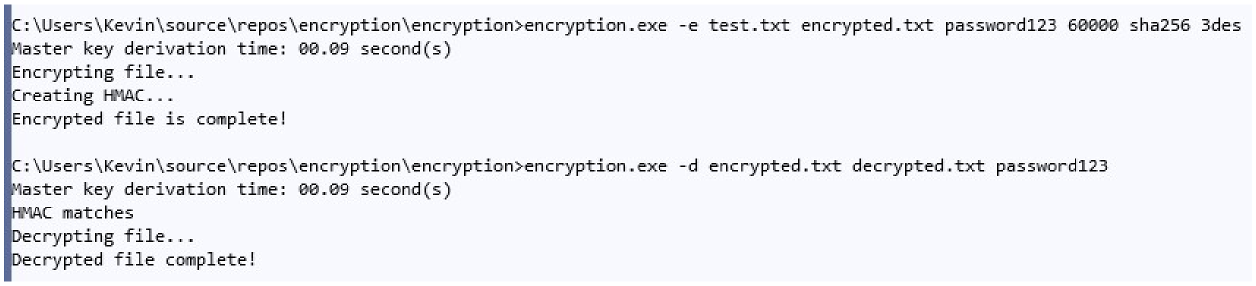
```
This is a test file.
This is a test file.
This is a test file.
This is a test file.
This is a test file.
```

Screenshot 6: SHA256 – AES128



```
C:\Users\Kevin\source\repos\encryption\encryption>encryption.exe -e test.txt encrypted.txt password123 60000 sha512 aes256
Master key derivation time: 00.26 second(s)
Encrypting file...
Creating HMAC...
Encrypted file is complete!

C:\Users\Kevin\source\repos\encryption\encryption>encryption.exe -d encrypted.txt decrypted.txt password123
Master key derivation time: 00.25 second(s)
HMAC matches
Decrypting file...
Decrypted file complete!
```

Screenshot 7: SHA256 – 3DES



```
C:\Users\Kevin\source\repos\encryption\encryption>encryption.exe -e test.txt encrypted.txt password123 60000 sha256 3des
Master key derivation time: 00.09 second(s)
Encrypting file...
Creating HMAC...
Encrypted file is complete!

C:\Users\Kevin\source\repos\encryption\encryption>encryption.exe -d encrypted.txt decrypted.txt password123
Master key derivation time: 00.09 second(s)
HMAC matches
Decrypting file...
Decrypted file complete!
```