Computer and Network Security: Self Study Lab

Lab #2-2 Symmetric Encryption (II)

1. Introduction

The learning objective of this self-study lab is for students to get familiar with the concepts and practical usage in the symmetric secret-key encryption using `openssl`. After finishing the lab, students should be able to gain a first-hand experience on encryption algorithms, encryption modes, and padding.

2. Lab1: Encoding with miscellaneous encryption algorithm

The following Lab are designed for students to practice various encryption algorithm. Investigate the `openssl` manual to complete the assigned Labs.

Lab 1.1: What is a single command line to encrypt a file with **blowfish** in CBC using salt and output the file in base64 format?

```
openssl enc -bf-cbc -salt -in [file to encrypt] | base64 >
[output file]
```

Lab 1.2: What is a command line to decrypt a file from 1.1?

```
base64 --decode [file to decrypt] | openssl enc -bf-cbc -d
-salt -out [output file]
```

Lab 1.3: What is a single command line to encrypt a file with **3des** in CBC using salt and output the file in base64 format?

```
openssl enc -des-ede3-cbc -salt -in [file to encrypt] | base64
> [output file]
```

Lab 1.4: What is a command line to decrypt a file from 1.3?

```
base64 --decode [file to decrypt] | openssl enc -des-ede3-cbc
-d -salt -out [output file]
```

Lab 1.5: What is a single command line to encrypt a file with **cast5** in CBC using salt and output the file in base64 format?

```
openssl enc -cast5-cbc -salt -in [file to encrypt] | base64 >
[output file]
```

Lab 1.6: What is a command line to decrypt a file from 1.5?

```
base64 --decode [file to decrypt] | openssl enc -cast5-cbc -d
-salt -out [output file]
```

3. Lab 2: padding

Lab 2.1: Create a file `plaintext20B.txt` with size 20 bytes. Show the file size information with `ls -l plaintext20B.txt`

-rw-rw-r--. 1 lab7 lab7 12 Mar 4 13:45 plaintext20B.txt
plaintext20B.txt

Lab 2.2: Given a password= 00112233445566778899aabbccddeeff and IV = 0102030405060708), what is a command line to encrypt the file `plaintext20B.txt` and output to file `cipher20B-aes-128-cbc.bin` with 128 bit AES and CBC mode?

openssl enc -aes-128-cbc -K 00112233445566778899aabbccddeeff -iv 0102030405060708 -in plaintext20B.txt -out cipher20B-aes-128-cbc.bin

Lab 2.3: Show the file size of `plaintext20B.tx`***t*** and *cipher20B-aes-128-cbc.bin.* Explain why `cipher20B-aes-128-cbc.bin` has such a file size?

```
-rw-rw-r--. 1 lab7 lab7     20 Mar  4 23:11
plaintext20B.txt

-rw-rw-r--. 1 lab7 lab7     32 Mar  4 23:11 cipher20B-
aes-128-cbc.bin
```

encrypted file size = input + 16 - (input % 16)
= 20 + 16 - (20 % 16)
= 20 + 16 - 4
= 32 bytes

Lab 2.4: Decrypt the file *`cipher-aes-128-cbc-20B.bin`* and output the result to the file *`plain20B-decrypt.txt`*. Show the command line.

openssl enc -d -aes-128-cbc -K 00112233445566778899aabbccddeeff -iv 0102030405060708 -in cipher20B-aes-128-cbc.bin -out plain20B-decrypt.txt

Lab 2.5: Decrypt the `cipher-aes-128-cbc-20B.bin` with option **`–nopad`** and output to file `plain20B-decrypt-nopad.txt`. Show the command line.

openssl enc -d -nopad -aes-128-cbc -K 00112233445566778899aabbccddeeff -iv 0102030405060708 -in cipher20B-aes-128-cbc.bin -out plain20B-decrypt-nopad.txt

Lab 2.6: Compare file size `plain20B-decrypt.txt` and `plain20B-decrypt-nopad.txt`. Explain the different of file size. What are the extra contents in the bigger file?

-rw-rw-r--. 1 lab7 lab7 20 Mar 4 23:12 plain20B-decrypt.txt

-rw-rw-r--. 1 lab7 lab7 32 Mar 4 23:13 plain20B-decrypt-nopad.txt

-nopad don't remove padding.

Lab 2.7: If an original file size is equal to 32 bytes. What are the file size of three files generated by the same procedures in 2.4 and 2.5?

```
-rw-rw-r--. 1 lab7 lab7    32 Mar  4 23:11 cipher20B-aes-128-cbc.bin
-rw-rw-r--. 1 lab7 lab7    20 Mar  4 23:12 plain20B-decrypt.txt
-rw-rw-r--. 1 lab7 lab7    32 Mar  4 23:13 plain20B-decrypt-nopad.txt
```

Lab 2.8: Encrypt the file plaintext20B.txt with AES 128 bit in 4 different modes: ECB, CBC, CFB and OFB using the same password and IV. Output the encryption to the file

`cipher20B-aes-128-ecb.bin`,
`cipher20B-aes-128-cbc.bin`,
`cipher20B-aes-128-cfb.bin`, and
`cipher20B-aes-128-ofb.bin`.

Compare all output file size with the plaintext and explain the results.

```
-rw-r--r-- 1 lab7 lab7 32 Mar 5 00:09 cipher20B-aes-128-cbc.bin
-rw-r--r-- 1 lab7 lab7 20 Mar 5 00:09 cipher20B-aes-128-cfb.bin
-rw-r--r-- 1 lab7 lab7 32 Mar 5 00:09 cipher20B-aes-128-ecb.bin
-rw-r--r-- 1 lab7 lab7 20 Mar 5 00:09cipher20B-aes-128-ofb.bin
```

4. Lab 3: Mode of Operations

This lab is to investigate the effect of various mode of operations when the encrypted file is corrupted. We will prepare a simple file and encrypt it, after that the encrypted file was manually modified to create corrupted effect. You will learn the different phenomena of mode of operations.

Preparation:
(1) Create a file `corrupted.txt` which contain 3 lines of phrases as follows:

Let's rock the Linux
Let's rock the Linux
Let's rock the Linux

Make sure that the total file size of `corrupted.txt` is equal to 69 bytes!!!!

(2) Encrypt `corrupted.txt` with AES 128 bit with ECB, CBC, CFB and OFB and named the output files as
`cipher-aes-128-ecb.bin`,
`cipher-aes-128-cbc.bin`,
`cipher-aes-128-cfb.bin`, *and*
`cipher-aes-128-ofb.bin` respectively.

Lab 3.1 Using hex editor (pick any editor from the Internet) to change the 30th bytes (change only a single bit) of `cipher-aes-128-ecb.bin` and save the corrupted file using the same name.

Decrypt the file `cipher_aes_128_ecb.bin` which is became a corrupted file and output the result to `decrypt-corrupted-aes-128-ecb.txt`. Show the output of `decrypt-corrupted-aes-128-ecb.txt`.

```
Let's rock the Linux
Let's rock the Linux
Let's rock the Linux
```

Lab 3.2 Do the same procedure as 3.1 for the file `cipher-aes-128-cbc.bin`.

```
Let's rock the Linux
Let's rock the Linux
Let's rock the Linux
```

Lab 3.3 Do the same procedure as 3.1 for the file `cipher-aes-128-cfb.bin`.

Lab 3.4 Do the same procedure as 3.1 for the file `cipher-aes-128-ofb.bin`.