Lab 4 - Symmetric Key Crypto

Deadline: Friday, 10 April 2020. 11:55 pm

This lab is to be done individually.

Overview

The learning objective of this lab is for students to get familiar with the concepts in the symmetric-key encryption. After finishing the lab, students should be able to gain first-hand experience of encryption algorithms, encryption modes, paddings, and initial vector (IV). Moreover, students will be able to use tools and write programs to encrypt/decrypt messages.

Reading Material

This lab requires you to be familiar with block cipher modes. Please go through the supplementary material (Lab4_Reading_BlockCipherModes.pdf) before starting the lab.

Lab Setup

This lab requires the following tools:

1. Openssl
2. A hex editor (Ghex or Bless. You can also use an online hex editor)

You can either install these tools on your own or download the prepared VM from here or here or here.
(If you have trouble setting up the VM please contact Mannan or your primary TA).

Note. The VM file you download is a virtual hard disk.
Note. VM username: seed , password: dees

Note. For those that cannot set up their VMs, ubuntu and macOS has openssl installed by default. You would only need to install a hex editor for the lab.

Lab Tasks

Task 1: Encryption using different ciphers and modes

In this task, we will play with various encryption algorithms and modes. You can use the following “openssl enc” command to encrypt/decrypt a file. To see the manuals, you can type “man openssl” and “man enc”.

```
“openssl enc ciphertype -e -in plain.txt -out cipher.bin -K
00112233445566778889aabbccddeeff -iv 0102030405060708”
```

Please replace “cipher type” in the above command with a specific cipher type, such as -aes-128-cbc, -aes-128-cfb, -bf-cbc, etc. In this task, you should try at least 3 different ciphers and three different modes. You have to create and fill a plain text file yourself. You can find the meaning of the command-line options and all the supported cipher types by typing"man enc". We include some common options for the openssl enc command in the following:

1. -in <file> input file
2. -out <file> output file
3. -e encrypt
4. -d decrypt
5. -K/-iv <input> key/iv in hex is the next argument
6. -[pP] print the iv/key (then exit if -P)

Task 2: Encryption Mode – ECB vs. CBC

The file ***pic_original.bmp*** (uploaded on lms: assignment tab) contains a simple picture. We would like to encrypt this picture, so people without the encryption keys cannot know what is in the picture. Please encrypt the file using the ECB (Electronic Code Book) and CBC (Cipher Block Chaining) modes, and then do the following:

1. Let us treat the encrypted picture as a picture, and use a picture viewing software to display it. However, for the .bmp file, the first 54 bytes contain the header information about the picture, we have to set it correctly, so the encrypted file can be treated as a legitimate .bmp file. We will replace the header of the encrypted picture with that of the original picture. You can use a hex editor tool (e.g. ghex or Bless) to directly modify binary files. *Note. Bless is the better tool for this job.*
2. Display the encrypted picture using any picture viewing software. Can you derive any useful information about the original picture from the encrypted picture? Please explain your observations.

Template for submission

CipherType	Observation	Explanation
ECB	…	…

Task 3: Encryption Mode – Corrupted Cipher Text

To understand the properties of various encryption modes, we would like to do the following exercise:

1. Create a text file that is at least 64 bytes long.
2. Encrypt the file using the AES-128 cipher.
3. Unfortunately, a single bit of the 30th byte in the encrypted file got corrupted. You can achieve this corruption using a hex editor.
4. Decrypt the corrupted file (encrypted) using the correct key and IV. Please answer the following questions:
    a. How much information can you recover by decrypting the corrupted file, if the encryption mode is ECB, CBC, CFB, or OFB, respectively? Please answer this question before you conduct this task, and then find out whether your answer is correct or wrong after you finish this task. *Note. Go through the readings provided on LMS before answering this question.*
    b. Please explain why.
    c. What are the implications of these differences?

Template for submission

CipherType	Information Recovery	Observations after the task	Implications
ECB	…	…	…

Task 4 : Padding

For block ciphers, when the size of the plain text is not the multiple of the block size, padding may be required. In this task, we will study the padding schemes. Please do the following exercises:

1. The openssl manual says that openssl uses PKCS5 standard for its padding. Please design an experiment to verify this. In particular, use your experiment to figure out the

paddings in the AES encryption when the length of the plaintext is 20 bytes and 32 bytes.

2. Please use ECB, CBC, CFB, and OFB modes to encrypt a file (you can pick any cipher). Please report which modes have paddings and which ones do not. For those that do not need paddings, please explain why.

Template for submission.

4.1

Experiment

Hypothesis	
Method	
Findings for 20 bytes file	
Findings for 32 bytes file	

4.2

Ciphermode	Observations
ECB	…

Submission

Follow the template provided below each question to report your findings.
You are required to submit a .pdf file with all of your findings by Friday 11:55pm