Crypto Lab – Symmetric Key Ciphers

The development of this document is/was funded by three grants from the US National Science Foundation: Awards No. 0231122 and 0618680 from TUES/CCLI and Award No. 1017771 from Trustworthy Computing.

This document has been slightly modified by Mirela Damian, Villanova University. The original document can be found at `http://www.cis.syr.edu/~wedu/seed/`.

1 Overview

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

2 Lab Environment

Cryptography library `OpenSSL`. In this lab, we will use `openssl` commands and libraries. Make sure you have `openssl` installed in your VM. It should be noted that if you want to use `openssl` libraries in your programs, you also need to install `libssl-dev`, the development version of ssl, using the following command:

```
sudo apt-get install libssl-dev
```

Binary editor `GHex`. In this lab, we need to be able to view and modify files of binary format. Make sure you have `GHex` installed in your VM. GHex is a hex editor for GNOME that allows the user to load data from any file, view and edit it in either hex or ascii.

3 Lab Tasks

3.1 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 the `ciphertype` with a specific cipher type, such as `-aes-128-cbc`, `-aes-128-cfb`, `-des-cbc`, etc. In this task, you should try at least 3 different ciphers and three different modes. 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:

```
-in <file>      input file
-out <file>     output file
-e              encrypt
-d              decrypt
-K/-iv          key/iv in hex is the next argument
-[pP]           print the iv/key (then exit if -P)
```

3.2 Task 2: Encryption Mode – ECB vs. CBC

The file `pic_original.bmp` contains a simple picture. We would like to encrypt this picture, so people without the encryption keys cannot know what is in the picture. 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, the first 54 bytes of a `.bmp` file contain the header information about the picture. We have to set these bytes correctly, so that 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 the `ghex` tool to directly modify binary files.

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.

3.3 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 `ghex`.

4. Decrypt the corrupted file (encrypted) using the correct key and IV.

Please answer the following questions: (1) 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. (2) Explain why. (3) What are the implication of these differences?

3.4 Task4 : Padding

For block ciphers, when the size of the plaintex 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 (try `man enc`) says that `openssl` uses PKCS5 (Public Key Cryptography Standard) for its padding. In this standard the value of each byte is the number of bytes that are added – for example, if padding is required for 4 bytes, the hexadecimal padding string `04 04 04 04` is used. If the input data is a multiple of the block size, then an extra block of bytes with the block size value is added – for example, if the input data is 80 bytes and the block size is 16 bytes, an extra padding block of 16 bytes, with each byte `10` in hexadecimal, is added.

   Design an experiment to verify the padding scheme. 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. Use ECB, CBC, CFB, and OFB modes to encrypt a file (you can pick any cipher). Report which modes have paddings and which ones do not. For those that do not need paddings, explain why.

3.5 Task 5: Programming using the Crypto Library

So far, we have learned how to use the tools provided by `openssl` to encrypt and decrypt messages. In this task, we will learn how to use `openssl`'s crypto library to encrypt/descrypt messages in programs.

OpenSSL provides an API called EVP, which is a high-level interface to cryptographic functions. Although OpenSSL also has direct interfaces for each individual encryption algorithm, the EVP library provides a common interface for various encryption algorithms. To ask EVP to use a specific algorithm, we simply need to pass our choice to the EVP interface. A sample code is given in `http://www.openssl.org/docs/crypto/EVP_EncryptInit.html`. Please get yourself familiar with this program, and then do the following exercise.

You are given a plaintext and a ciphertext, and you know that `aes-128-cbc` is used to generate the ciphertext from the plaintext, and you also know that the numbers in the IV are all zeros (not the ASCII character '0'). Another clue that you have learned is that the key used to encrypt this plaintext is an English word shorter than 16 characters; the word that can be found from a typical English dictionary. Since the word has less than 16 characters (i.e. 128 bits), space characters (hexadecimal value 0x20) are appended to the end of the word to form a key of 128 bits. Your goal is to write a program to find out this key. You can download a English word list from the Internet. We have also linked one on the web page of this lab. The plaintext and ciphertext are as follows:

```
Plaintext (total 21 characters): This is a top secret.
Ciphertext (in hex format): 8d20e5056a8d24d0462ce74e4904c1b5
                            13e10d1df4a2ef2ad4540fae1ca0aaf9
```

Note 1: If you choose to store the plaintex message in a file, and feed the file to your program, you need to check whether the file length is 21. Some editors may add a special character to the end of the file. If that happens, you can use the `ghex` tool to remove the special character.

Note 2: In this task, you are supposed to write your own program to invoke the crypto library. No credit will be given if you simply use the `openssl` commands to do this task.

Note 3: To compile your code, you may need to include the header files in `openssl`, and link to `openssl` libraries. To do that, you need to tell your compiler where those files are. In your `Makefile`, you may want to specify the following:

```
IDIR=/usr/include/
LDIR=/usr/lib/x86_64-linux-gnu

all:
	gcc -I$(IDIR) -L$(LDIR) -o file file.c -lcrypto -ldl
```

4 Submission

You need to submit a detailed lab report to describe what you have done and what you have observed; you also need to provide explanation to the observations that are interesting or surprising. In your report, you need to answer all the questions listed in this lab.