41900 – Security Fundamentals
Secret-Key Encryption

OVERVIEW

The learning objective of this lab is for students to get familiar with the concepts in the secret-key encryption. The lab focuses on encryption algorithms, encryption modes, paddings, and initialization vector.

Install Bless

We will be using a tool called **`bless`** to view encrypted files in hexadecimal. To install this tool, you can use:

- **`Sudo apt-get install bless`**

When asked, select 'yes' to download and install the tool.

Just a recap: Encryption

You can use the following command to encrypt/decrypt a file.

- **`openssl enc <cipher-type> -e/-d -in <> -out <> -K <> -iv <>`**
    - **`-e`** encrypt
    - **`-d`** decrypt
    - **`-in`** input file
    - **`-out`** output file
    - **`-K`** secret key (in hexadecimal)
    - **`-iv`** initialization vector (in hexadecimal)

  `<cipher-type>` some types include:

  - Cipher Block Chaining (**`CBC`**)
  - Cipher Feedback (**`CFB`**)
  - Electronic Codebook (**`ECB`**)
  - Output Feedback (**`OFB`**)

TASKS

Task 1: Influence of IV in the encryption process:

Most of the encryption modes require an initial vector (IV). Properties of an IV depend on the cryptographic scheme used. If we are not careful in selecting IVs, the data encrypted by us may not be secure at all, even though we are using a secure encryption algorithm and mode. To understand the properties of IV we will do the following exercise:

41900 – Security Fundamentals
Secret-Key Encryption

1. Create a text file that is at least 64 bytes long (a single ASCII character is 1 byte of data).
2. Encrypt the file using the AES-128 cipher, setting the IV to **0000**
3. Decrypt the file using the correct key and IV.
4. Again, decrypt the file using the correct key but an incorrect IV, say **ffff**

Observe how many bytes of data were lost when the wrong IV was provided during decryption.

Task 2: Padding in block encryption

For block ciphers, when the size of the plaintext is not the multiple of the block size, padding may be required. The **openssl** manual says that **openssl** uses **PKCS5** standard for its padding. In this task, we will study the padding schemes. Please do the following exercises:

1. Create 2 files, one with 10 bytes of data and another with 16 bytes (a single ASCII character is 1 byte of data).
2. Use AES-128 along with ECB, CBC, CFB, and OFB modes to encrypt the files (separately).

Find out which modes have paddings and which ones do not.

Task 3: 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. Use AES-128 along with ECB, CBC, CFB, and OFB modes to encrypt the files (separately).
3. Use bless and change a single byte of the encrypted file to corrupt it.
4. Decrypt the corrupted file using the correct key and IV.

How much information can you recover by decrypting the corrupted file, if the encryption mode is ECB, CBC, CFB, or OFB, respectively.

Task 4: Encryption Mode – ECB vs. CBC

The file **image.bmp** is a simple image with a two-tone color. We will encrypt this picture, so people without the encryption keys cannot know what is in the picture. We will then try and extract information from the encrypted file. To do so, follow the steps below:

1. Encrypt the image using the ECB (Electronic Code Book) and CBC (Cipher Block Chaining) modes, save them separately (you can name the encrypted files **ecb.bmp** and **cbc.bmp**).

2. We will now open the original **`image.bmp`** file using the bless editor and copy the first 54 bytes of data. This is because in a **`bmp`** file, the first 54 bytes contain the header information about the picture. Save this information to a notepad as we will need it later.
3. We will now add the header information to our encrypted picture using the hex editor bless so that the system can convert the data to image bits.

Try to open the encrypted picture using any picture viewing software. See if you can derive any useful information about the original picture from the encrypted picture.