ECE 404 Homework #2

Due: Thursday 01/30/2020 at 4:29PM

Introduction

The goal in this homework is to help you understand the implementation of the DES (Data Encryption Standard).

Problem 1

Write a Perl or a Python script that implements the full DES. Refer to Lecture 3 on the course website as it outlines the key steps to DES. Your algorithm must produce encryption/decryption results using an encryption key.

Program Requirements

Your script for Problem 1 should have the following command-line syntax:

```
DES_text.py -e message.txt key.txt encrypted.txt
DES_text.py -d encrypted.txt key.txt decrypted.txt
```

An explanation of this syntax is as follows:

For encryption (indicated with `-e`), your script should read the input text from a file called `message.txt` (or whatever the name of the command-line argument after `-e` is). The next argument `key.txt` is the file containing the 64-bit DES encryption key in **text string** format. The encrypted output should be saved in **hexstring** format to a file with the name of the final argument, in this case a file called `encrypted.txt`.

For decryption (indicated with the `-d` argument), the input hexstring file is specified with argument after the `-d` argument, in this case `encrypted.txt`. The next argument specifies the file containing the DES decryption key (keep in mind the key used for both encryption and decryption should be the same). The decrypted output should be saved to a file with the name specified by the last argument, in this case `decrypted.txt`.

You are encouraged to use the starter file that is a part of the gzipped archive for Lecture 3 at your instructor's Lecture Notes website. This starter file includes all the permutation "tables" you need for the homework. In addition to this starter file, the gzipped archive also includes a script for generating the round key, a script for showing the permutation of the encryption key, a script demonstrating the substitution step, and a .txt file with the eight S-box tables.

The plaintext bit size is not necessarily divisible by the DES block size. For the sake of this assignment, your program can pad the last block with zeros if this is the case.

Problem 2

As you will soon learn in lecture 9, block ciphers such as DES should not be used in the manner used in problem 1 - directly encrypting the data in independent blocks (known as electronic code book mode). Because each block of data is encrypted independently, overall patterns in the data may still be obvious if used to encrypt an image, for example. This is not readily apparent when encrypting text like in Problem 1.

Therefore your job in problem 2 is to write a script that takes an image in PPM format, uses DES to encrypt the image data (**NOT** the PPM header) with a key contained in a text file, and then combine the encrypted image with a PPM header (you can use the original image's header) so you can write it as a readable PPM file. The result should be the encrypted image viewable as a PPM file.

Program Requirements

The call syntax for your program is similar to Problem 2:

```
DES_image.py image.ppm key.txt encrypted.txt
```

Here `image.ppm` is the input image you will encrypt, `key.txt` is the key you will use for encryption, and `image_enc.ppm` is the name of the .ppm file you will write the encrypted image to. You may use parts of your script from problem 1 for this, though you probably won't use all of it as text may be read differently than image data. Keep in mind you are not required to implement image decryption for Problem 2.

A ".ppm" image file consists of a header and the actual image data. The header occupies the first 3 lines of the file, and the rest is the image data (pixel values) that you want to encrypt. For those interested in a more general approach, descriptions of the PPM header can be found at:

- http://netpbm.sourceforge.net/doc/ppm.html
- http://paulbourke.net/dataformats/ppm/

Text and Image to be Used for the Submitted Output

The text and image to be used can be found in the Homework section at the course website at https://engineering.purdue.edu/ece404/homework.htm
The text is from a wired.com article discussing the Meltdown and Spectre vulnerabilities (it starts with "Earlier this week..."). The image is a PPM of a helicopter.

For debugging purposes, we have also have posted to the homework section a file named `first_round.txt` containing the left and right halves of the first plaintext block after the first Feistel round for the text file mentioned above.

Key to be Used for the Submitted Output:

purduece

Notes

- If you are having problems implementing DES, try to debug just one Feistel round first. That is, create a single-round version of DES and see if your decryption can recover the plaintext. For the purpose of debugging, you can start with a key that is made of all zeros and assume, during debugging, that all round keys are the same.
- Remember that text editors can add additional bytes to file contents. When reading in the key you should ignore additional bytes such as newline character since the key should be only 64 bytes.
- Please use your personal ECN account for submission (as opposed to an account for a class like ECE 364).
- Please include comments along with your code.
- If using Python, please denote the Python version in your code with either a shebang line (e.g. #!/usr/bin/env python3) or a comment denoting the version.
- If you would like to use your own key for testing, remember that the encryption key should consist of at least 8 printable ASCII characters. The same key should be used for both encryption and decryption.

Submission Instructions:

- Make sure to follow program requirements and submission instructions. **Failure to follow these instructions may result in loss of points!**.
- For Problem 1, your PDF submission should include your script, a brief explanation of your script, and the encrypted and decrypted output for the text mentioned above using the key provided.
- For Problem 2 of the homework, your PDF submission must include the script used for encrypting the PPM image, a brief explanation of the script, and a picture of the encrypted PPM image.
- As in homework 1, you will be electronically submitting your code and your PDF using the turnin command (see the next section for this command). Therefore your electronic submission must include 3 files.

Electronic Turn-In

- turnin -c ece404 -p hw02 hw02.pdf DES_text.pl DES_image.pl (if using Perl)
- turnin -c ece404 -p hw02 hw02.pdf DES_text.py DES_image.py (if using Python)