CS-G513 Homework #2
(4 marks

Due: 20/2/2018 before 11:59 PM

Introduction

The goal in this homework is to help you understand the implementation of the DES (Data Encryption Standard) algorithm and to observe for yourself the effects of Diffusion and Confusion (also known as the Avalanche effect)

Problem 1

Write a Perl or a Python script that implements the full DES. Please make sure that you implement all of the key generation steps. Your algorithm must produce encryption/decryption results using an encryption key. Your script should read the text from a file called 'message.txt' and the key from a file called 'key.txt'. It should place the encrypted output into 'encrypted.txt' and the decrypted output into 'decrypted.txt'. You should use 8 characters for the key. Please use the first seven bits of each character byte for the 56-bit key you need for DES. (For the submission, you must use the key and text that is shown in the last part of this document.)

You are encouraged to use the starter file that is part of the zipped archive homework2startercode.zip available in the same directory. This starter file includes all the permutation "tables" you need for the homework. In addition to this starter file, the zipped 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.

Problem 2

Once you have implemented DES, now estimate the extent of Diffusion and Confusion.

1. In order to observe the effects of Diffusion, you need to change one bit in your plaintext and determine the number of bits changed in the ciphertext. Find an average of the number of ciphertext bits changed for several plaintext blocks.

2. To understand the effect of the S-boxes on diffusion, populate the 4x16 tables for the S-boxes with randomly generated integers. Make sure that each of the randomly generated entries are between 0 and 15, both ends inclusive. Perform the same experiment as above to measure diffusion. The only difference between the previous task and this task will be the S-boxes. Repeat the experiment for 2 sets of randomly generated S-boxes. Your choice of different plaintext blocks and keys would stay the same.

3. For Confusion, change one bit in your encryption key and determine the number of bits changed in the ciphertext. Find an average of the number of ciphertext bits changed for several choices of the keys.

Text to be used for encryption

We prove that the set of DES permutations (encryption and decryption for each DES key) is not closed under functional composition. This implies that, in general, multiple DES-encryption is not equivalent to single DES-encryption, and that DES is not susceptible to a particular known-plaintext attack which requires, on average, 2 ∧ 28 steps. We also show that the size of the subgroup generated by the set of DES permutations is greater than 10 ∧ 2499 , which is too large for potential attacks on DES which would exploit a small subgroup.

Key to be Used for encryption:

bitscsis

Submissions

- For Problem 1, show your encryption and decryption output on the text above using the specified key in your submission. If you 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 would be used for both encryption and decryption. You should also include your script in your submission.
- For Problem 2 of the homework, your submission must include the scripts for computing the averages mentioned above and the values for the averages.

Although you are free to write your own code from scratch, here are some recommendations:

1. If using Python, you might want to start with BitVector class written by Prof. Avinash Kak from Purdue university that is available as python package.
2. If using Perl, use the "Bit::Vector" module from www.cpan.org. It is a popular Perl module for manipulating bit arrays. It is also well documented.

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. Also, make sure you use the hexdump utility to see the output of each function. Remember that text editors can add additional bytes to file contents.

- Your electronic submission must include 3 files, as indicated below.
  1. DES.py (to perform encrryption/decryption using DES, problem#1)
  2. Average.py (to calculate the average for problem #2)
  3. report.pdf (The code and output must also be submitted as a seperate pdf file.)
- Kindly include comments along with your code.
- Finally, submit your assignment in form of a single compressed file (<your-bits-id_Name.zip>. into the shared directory **Submissions_HW#2**, after encrypting with my public_key.