Project 1 - CS 767 – Foundations of Network Security

Assigned by: Dr. Murtuza Jadliwala

Due: 21st March 2016, 11:59 pm

Background:

Symmetric or shared-key cryptography is a message confidentiality preserving mechanism that involves a series of message transformations using some secret key known only to the sender and the receiver. The original message can be viewed by a reverse transformation process using the same pre-shared secret. The process of transforming information into an unintelligible form is called Encryption, while the reverse process is called Decryption. We have discussed numerous symmetric block ciphers in the class that are used to encrypt fixed blocks of data. *DES* was one of the most widely used symmetric block ciphers which is based on the Fiestel Cipher Structure (Details about DES can be found in the text book). Another encryption scheme based on the Fiestel cipher structure is the Tiny Encryption Algorithm (TEA). Details of the TEA can be found at the following link (http://143.53.36.235:8080/tea.htm).

Project Tasks (all the tasks are required):

1. **Task 1:** In this task, you will be required to investigate TEA in further detail and write a brief (one-page) write-up on the specific properties of TEA, for example, the maximum size of the key, maximum number of rounds, and the round function. Also list some of the major similarities and differences between DES and TEA in this write-up.

2. **Task 2:** In the class, we studied various additional block cipher modes, such as Cipher Block Chaining (CBC), Counter Mode (CTR) and Output Feedback (OFB), which could be used to extend the standard (Electronic Code Book mode) block cipher operation. In this task of the project, you will be required to implement any ***TWO*** of the modes of operation of your choice from the following - Cipher Block Chaining (CBC), Counter Mode (CTR) and Output Feedback (OFB) for the Tiny Encryption Algorithm (TEA). The implementation must be done using the C programming language. The C source code and details about the Tiny Encryption Algorithm can be found by following the links http://143.53.36.235:8080/source.htm#ansi and http://143.53.36.235:8080/tea.htm respectively. The TEA source code contains two functions, one for encrypt and one for decrypt operation. Your task is to code two additional functions for each block cipher mode

you are implementing, i.e., cbc_encrypt and cbc_decrypt if you are implementing the CBC mode. Each of these new functions should make use of (or call) the original encrypt and decrypt functions of the TEA cipher.

Next, you are required to write C code for encryption and decryption using the corresponding block cipher modes in the standard Data Encryption Standard (DES) implementation (using the openssl or crypto libraries). Note that you do NOT have to implement these block cipher modes in DES from the ground-up. Rather, you can use the appropriate openssl library functions already available for DES. You can find information and details about the openssl library by going to http://www.openssl.org/docs/crypto/crypto.html and the libcrypto library at http://www.gnu.org/software/libgcrypt/. Some of the openssl library functions you can use in your code:

```
void DES_ofb_encrypt((const unsigned char *in, unsigned char
*out, int numbits, long length, DES_key_schedule *schedule,
DES_cblock *ivec);
```

```
void DES_ncbc_encrypt(const unsigned char *input, unsigned
char *output, long length, DES_key_schedule *schedule,
DES_cblock *ivec, int enc);
```

```
Where: input-plaintext, output-ciphertext, ivec-initialization
vector, enc-encrypt or decrypt, schedule-to check for strength
of key, length-output length. You will need to include more
functions in your code that can be found on the links below.
If you used the openssl library, you can compile your code on
the CS servers as follows: gcc program.c –o program –lcrypto
and then execute
```

Detailed information about the functions is available at http://www.openssl.org/docs/crypto/des.html# and information about the details of DES cipher modes can be found at http://www.openssl.org/docs/crypto/des_modes.html#

3. **Task 3:** In this last task of the project, you are required to do performance measurements for both TEA and DES in the two block cipher modes of operation that you implemented earlier. What is needed is to record the time it takes for encryption and decryption for various message sizes. Your C program should be able to read the data to be encrypted from a file and write the output to a file. You should create text files of sizes 64, 512, 4096 and 32768 ASCII characters as input files from which data will be read as input to your program.

Your code should be able to encrypt and decrypt the information as well as measure the time it takes to perform both operations in both modes of operations you have implemented. The "gettimeofday" function in C will be useful for measuring time, by appropriately placing it before and after the encryption (and decryption) function, in order accurately measure the difference in time.

Guidelines and Deliverables:

a. The project can be either done ***individually*** or in groups of maximum ***TWO (2)*** students. If you decide to do a group project, you should identify a group member and email the names of both group members to the GTA Anindya Maiti at axmaiti@wichita.edu no later than *Friday, 26th February 2016, 11:59 pm*. When sending email, please 'cc' all group members on the email. If no information about your group is received on or before that date, you will NOT be allowed to form any group and will be expected to work individually on the project. The maximum number of group members is ***TWO (2)*** and no changes can be made to your group after it is announced. If you are doing the project individually, you do not need to send any notification email to the GTA.

b. ***Deliverable 1:*** A zipped file containing your group's implementation named "project1_code-username.zip". In the zip file, include the source code, input text files used for analysis, any other libraries used and executable files. It is your task and responsibility to make sure that your code compiles and runs on the Wichita State University CS servers (e.g., kirk.cs.wichita.edu or mccoy.cs.wichita.edu) before the final submission. To enable coding directly on the CS servers, the libgcrypt and crypto libraries are already installed on the servers. Also make sure that you submit the correct (and all the) files as no additional submissions will be allowed after the due date. Ensure that you have enough comments to explain each major step in of your code.

c. ***Deliverable 2:*** A typed report giving detailed information about Task 1 and the results obtained in Task 3. The report should be in ".pdf" format and the file should be named "project1_report-username.pdf". All details that you think are relevant to your project should be included in the report. NO printouts or hand written reports will be accepted. **If you used any information in your report from a research paper or web article, it should be appropriately cited or referenced. Any information used in the project that is from an external source and is not cited or referenced will be considered as plagiarism.** If the project is done in a group, every member of the group should contribute equally to both the coding and report preparation efforts. No credit will be given to a particular group member if he/she has not participated adequately in the completion of the project. Each group is required to work independently and any cooperation between groups is NOT allowed and, if detected, will be considered as academic dishonesty. Any academic dishonesty or plagiarism detected during or after the project submission will be dealt with as per the guidelines outlined in the syllabus.

d. ***Deliverable 3:*** A "README.TXT" file providing specific details and instructions on how to execute your code should also be submitted. It should also indicate the tasks and contributions of each group member in case of a group project.

Project Submission:

All necessary files, i.e. zipped source code, project report and README.TXT should be submitted using the "handin" command from the CS servers by ONLY one of the group members. NO email submissions will be accepted by either the professor or the TA and will be automatically deleted. Avoid submitting multiple files from the same group. When submitting your project, navigate to the directory where the files to be submitted are located and use the following command to submit. **~cs767/bin/handin 1 project1_code-username.zip project1_report-username.pdf README.TXT**. If you submit the same file(s) multiple times, only the latest submission will be retained. Instructions on how to use the CS servers and the handin command can be found at: http://www.cs.wichita.edu/~wallis/handin/students.html

Grading:

Task 1: *15 points*

Task 2: *50 points*

Task 3: *25 points*

Readme file and good programming style (includes comments and proper indentation)*: 10 points*

Total possible points: 100