Project Report On

File Encryption and Decryption with DES

By
Md. Mahedi Hasan
1014312019

ICT-6544: Applied Cryptography
Programming Project 1

Date of Submission: February 20, 2016

Problem Definition:

File Encryption and Decryption with DES

- Implement two programs, File encryptor and File decryptor.
- The program must use DES algorithm for encryption and decryption
- You program must implement ECB or CBC mode for reading and writing files.
- The encryption program will read a text file (input.txt) for data and a key file (key.txt) for key. It will save the output in output.txt
- The decryption program will read a text file (output.txt) for data and a key file (key.txt) for key. It will save the output in gen_input.txt
- The input.txt and gen_input.txt must be same if your programs are correct.

Introduction:

The main goal of Cryptography is to keep sensitive data secure from unauthorized user. It requires two basic element for encryption and decryption i.e. encryption/decryption algorithm and key. On the basis of key two types of encryption is done. If the single key is used for both encryption and decryption then it is symmetric cipher (private key cryptography). On other side, if pair of keys (public and private keys) is used then it is asymmetric cipher (public key cryptography). The DES (Data Encryption Standard) is a symmetric block cipher and most widely used encryption scheme. In this project DES Algorithm is being implemented.

DES Algorithm Description:

DES encryption algorithm take 64-bit block plaintext and 64-bit key. It use the 16 round consisting same function. The encryption scheme is shown in Figure 1. There are two input to algorithm Plaintext and Key.

Steps of encryption

1. Firstly, the plaintext to be encrypted is divided into fixed size 64-bit block.

2. Every eight bit of 64-bit key is ignored and 56-bit key is input to algorithm.

3. On the block of plaintext (64-bit) initial permutation (IP) is performed and divided into two halves L_i, R_i (each of 32-bit). On the key (56-bit) permuted choice1 (PC_1) is performed and it is also divided into two halves C_i, D_i (each of 28-bit).

4. After permutation, Data is processed in 16 rounds of same function f.

5. During each round, permuted key halves shifted left circular by 1 or more depend on implementation and then apply permuted choice2 (PC_2) (48-bit key as output).

6. In each round data is encrypted using 48-bit key and output of first round become input of second round .

7. After 16th round 32-bit halves output is swapped and produced Pre output.

8. At last inverse initial permutation (IP^{-1}) is performed on pre output and finally get the 64-bit cipher text.

Steps of decryption:

Decryption process is much similar to encryption unless the round key are implemented on reverse order than that of encryption.

Modes of DES:
1. Electronic Codebook (ECB) mode
2. Cipher Block Chaining (CBC) mode
3. Cipher Feedback (CFB) mode
4. Output Feedback (OFB) mode.

In this project ECB mode is used. In this mode the complete plaintext is basically divided into 64 bit block. DES algorithm is implemented in all this block separately and then grouped together to get the complete cipher text.

DES Algorithm

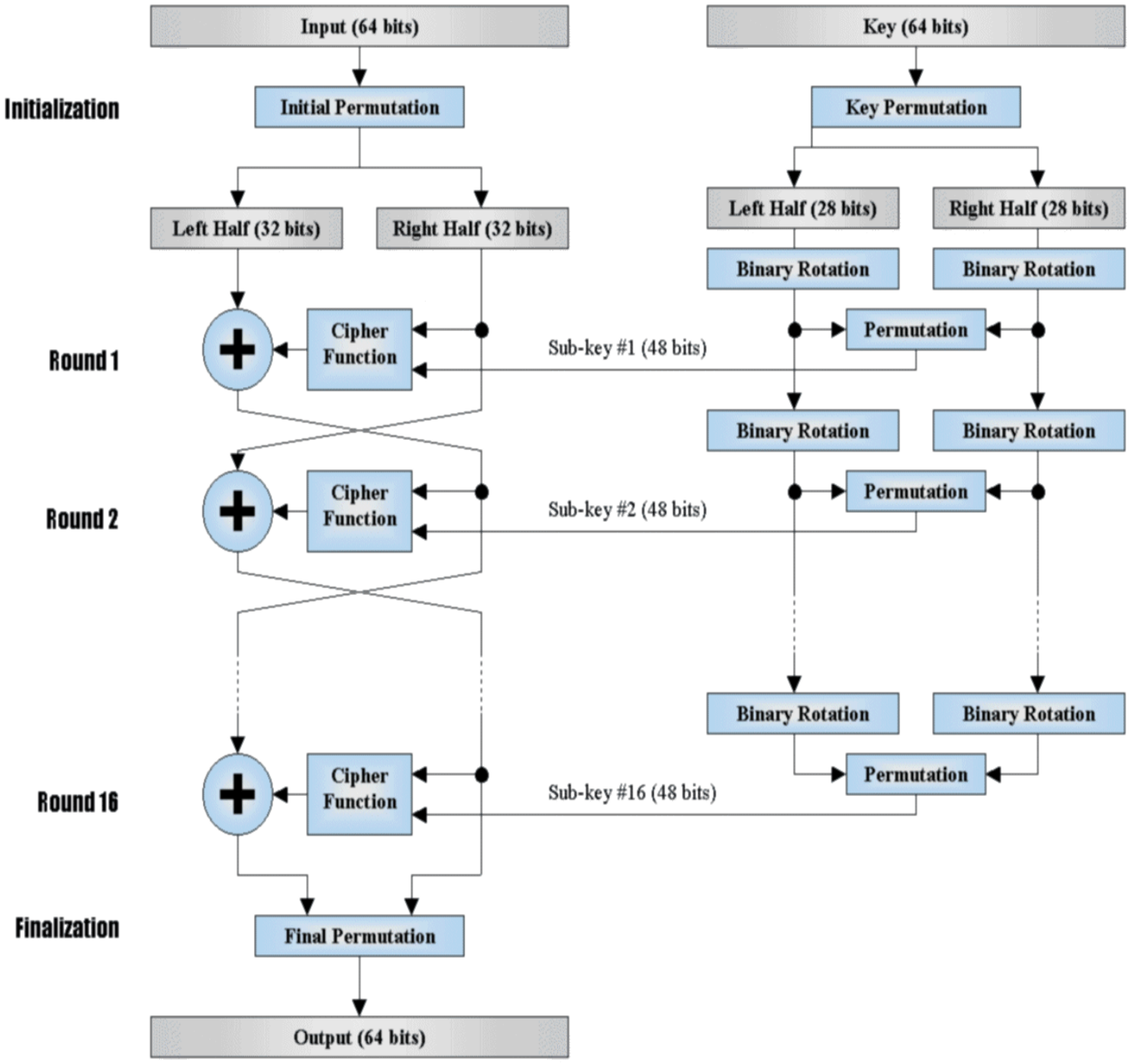

Figure 1: Structure of DES Algorithm Encryption and Key generation.

Code Description

The project is built in java language. Here four classes has been used.

1. AllData.java
2. KeyGenerator.java
3. DESAlgorithm.java
4. DESImplementation.java

AllData.java:

This class contains all necessary tables used in des algorithm. SBoxes, Initial and Final Permutation table, Permuted Choices, Expansion and Compression boxes are written according to the definition of DES Algorithm. Special methods are also written to convert these boxes into one dimensional array.

KeyGenerator.java:

This class generates 16 different 48 bit key required for des algorithm from given key.txt file. If the length of given key is more than 8 characters or 64 bit than first 8 character is considered as key and other characters are omitted.

In this project minimum key length is considered as 1 character or 8 bit. In this case other bits are considered as 0. Round key are generated according to the definition of the des algorithm.

DESAlgorithm.java

This class contains all functions necessary for implementing des encryption and decryption. This class converts every 8 characters plaintext block to 64 bit input block and produce 64 bit output block. Output block is then converted to cipher text. This class uses round key from KeyGenerator class.

For decryption round key generated from keyGenerator class is being implemented in reverse order.

DESImplementation.java

It is the main class of the project which take plaintext from input.txt file and key from key.txt file and do necessary padding in ECB mode and then produce cipher text in output.txt file using DES algorithm.

It also produce plaintext from cipher text written in output.txt file using DES decryption algorithm and put this plaintext in gen_input.txt file.

Rules for running project:

This project is built in NetBeans IDE using java language and hence to run the project JRE is first needed to install.

In the project folder All Four java source files, the jar file, input.txt, key.txt, run.bat and README.txt file is given.

For windows machine simply run the bat file.

After running this project by double clicking bat file the command prompt open. The project take input plaintext from input.txt file and key from key.txt file.

Please enter 1 for encryption. The project then produce output.txt file containing cipher text in the same directory. For decryption please enter 2. It then generates gen_input.txt file which contains original plaintext.

For complete implementation please DO NOT MODIFY the ouput.txt file. Otherwise plaintext from input.txt will not be similar to that gen_input.txt.

Please DO NOT CHANGE the encoding mode of any file while running this project.

For Linux machine simply run the jar in Terminal using Linux command.

Summary:

In this project DES cryptographic algorithm is implemented. The project can encrypt any plaintext given in input.txt file. It can also decrypt any cipher text given in output.txt file. For complete implementation the text of the input.txt should be similar gen_input.txt file. After checking various plaintext this project shows 100% accuracy.