Eastern Mediterranean University

Faculty of Engineering

Department of Software Engineering

****

*Course Instructor: Assoc. Prof. Alexander Chefranov*

Submitted to

Dr.Anas Ibrahim

By

1. FARAJ KHALID KHAN (143289)

2.SUNDUS HAROON (15701065)

The Term Project Topic is

*Task “Secure Software System”*

In partial fulfillment of the requirements for the course

CMSE353

Famagusta, Turkish Republic of North Cyprus

25th December 2017

**Table of Content**

[Problem Definition](#_Toc465205989) 1

1. [Algorithm Description](#_Toc465205989) 2

Figure.1.1.General Deception:…………………………………………………….………………….……..2

1.2.Test ………………………………………………………………………………………………………………….……..2

1.2.1. Initial permutation: ………………………………………………………………………….……….2

Table.1.2.1.a. Simple 64 bits table:……………………………………………………..…….3

Table.1.2.1.b.Initial Permutation table:……………………………………………….…...3

1.2.2. Inverse of the initial permutation:……………………………………………………………..3

Table. 1.2.2.a.The Inverse Initial Permutation table:………………………………….…..3

Figure.1.2.2.b.Single Round Des Algorithm:…………………………………………………...4

[1.2.3.Expansion/Permutation…………………………………………………………………………………..…….4](#_Toc465205989)

Table.1.2.3.a.Expansion/Permutation (E table):………………………………………………....4

[1.2.4. Round key generation 5-8](#_Toc465205989)

Table.1.2.4.a.8th bit Permuted Choice 1……………………………………………………………..5

Table. 1.2.4.b.Permuted Choice 1 without 8th bit………………………….……………………5

[1.2.5.Permuted Choice 1. 6](#_Toc465205989)

Table.1.2.5.a.(PC-1)………………………………………………………………….………..………….6

[1.2.6.Left circulat shifts schedule 6](#_Toc465205989)

Table 1.2.6.a Left circular shifts…………………………………………………………….…….6-7

[1.2.7.Permuted Choice 2](#_Toc465205989) 7

Table.1.2.7.a.(PC-2)………………………………………………………………………….…...7-8

[1.2.8.XOR with round key 8](#_Toc465205989)

[1.2.9.S-Boxes 9](#_Toc465205989)-11

Figure.1.2.9.a.Calculation of F(R,K): ……………………………………………………..……….9

Table.1.2.9.b.Des S-Box:……………………………………………………………………….....9-10

[1.3.Permutation P after S-boxes 11](#_Toc465205989)

Table.1.3.1.Permutation P after S-boxes:…………………………………………………… 11

[1.3.2.XOR with left half 11](#_Toc465205989)-12

[1.3.3.Swap of the halves 12](#_Toc465205989)

1.3.4.Padding:……………………………………………………………………………………..……………………..13

1.3.4.Cipher Feedback (CFB) mode:…………………………………………………………………………13-14

Figure.1.3.4.a.s-bite Cipher Feedback mode:……………………………………………………………….14

2. [Tools Description](#_Toc465205989) 14

* Vim (Text Editor)…………………………………………………………………………………….………..14
* Pyuic4 (Converter)…………………………………………………………………………………….……..14
* Qt designer(interface designer)………………………………………………………………………..14

3[.Developed Program Description](#_Toc465205989) 14-15

4[.User Guide 15](#_Toc465205989)

* Packages(for linux):……………………………………………………………………………………………15
* Database used:…………………………………………………………………………………………………15
* Librarie:……………………………………………………………………………………………………………..15

[5.Conducted Tests Description 15](#_Toc465205989)-28

5.1.Mandatory Access Control (MAC):………………………………………………………………………….15

5.2.Administrator……………………………………………………………………………………………………….15

* Login Panel (as administrator):…………………………………………………………………………15

Figure.5.2.a.Login Panel…………………………………………………………………………15-16

* Administrator Panel:…………………………………………………………………………………………16

Figure.5.2.b.Administrator Panel:………………………………………………………………………16

* Introducing a new user:…………………………………………………………………………………….17
* Creating new user:………………………………………………………………………………….17

Figure.5.2.1.a.Introducing new user ……………………………………………………….17

* Disabling/enabling a user:…………………………………………………………………………….18-19

Figure.5.2.3.a.Disable/Enable User:……………………………………………………..18

* Enable User:……………………………………………………………………………………………18

Figure.5.2.3.b.Enabling user: …………………………………………………………………18

* Disable User:………………………………………………………………………………………….19

Figure.5.2.3.c.Disabling Users……………………………………………………………….19

* Giving/revoking users rights:……………………………………………………………………….19-20

Figure.5.2.4.a.Grant/revoke users rights:…………………………………………………………..19

* Granting/Giving:……………………………………………………………………………………19-20
* Revoking:……………………………………………………………………………………………………20
* Administrator Rights: …………………………………………………………………………………20-22
* Creating object: ………………………………………………………………………………………….20

Figure.5.2.5.a.Creating Objects:………………………………………………………………………………….20

* Deleting object: ………………………………………………………………………………21

Figure.5.2.5.b.Deleting Objects: …………………………………………………………..21

* View Files: …………………………………………………………………………………21-22

Figure.5.2.6.a.View Files: ……………………………………………………………………………21-22

5.3.User…………………………………………………………………………………………………………………22-26

* Login Panel(User): ………………………………………………………………………………………22

Figure.5.3.1.a.User panel: …………………………………………………………………………..22

* Accessing objects: ……………………………………………………………………………………….23
* Creating object: …………………………………………………………………………..….23

Figure.5.3.2.a.Creating Objects: …………………………………………………..…23

* Deleting object: ……………………………………………………………………………….23

Figure.5.3.2.b.Deleting Objects: ………………………………………………………23

* Viewing object: ………………………………………………………………………………..24

Figure.5.3.2.c.Viewing Objects: ………………………………………………………..24

* Encrypting/decrypting:……………………………………………………………………………25-26

Figure.5.3.3.a.Ecncryptor box:………………………………………………………………………25

Figure.5.3.3.b.Encrypted text:………………………………………………………………………25

Figure.5.3.3.c.Decrypted text:…………………………………………………………………….26

[References 27](#_Toc465205989)

[Conclusion 27](#_Toc465205989)

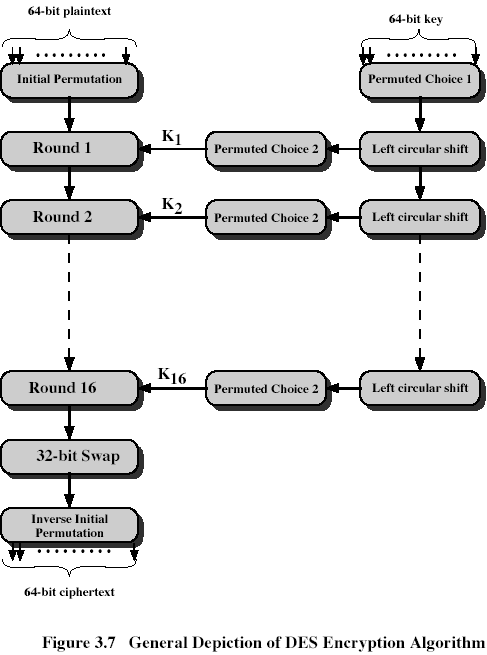
**Problem Definition:**

Problem Definition:

The aim of this report is to get the understanding of selected problem variant from ACM (Mandatory Access Control) and DES (Cipher Feedback) mode. Implementing solution of the problem as a software application using appropriate tools .Furthermore, to show the conducted tests of the application to show the validity of its correctness. Moreover, this report aim is to show the description of the algorithm for solving the problem, description of used tools and details of their installation, developed program description, user guide to show how to use the program, and the conducted tests and results description. The whole project is done in linux.

1. **Algorithm Description:**

**Figure.1.1.General Deception:**



*As Shown in the figure.1.1.* The Plaintext is broken into blocks of length 64 bits. Encryption is block wise. A message block is first gone through an initial permutation IP, then divided into two parts L0,where L0 is the left part of 32 bits and R0 is the right part of the 32 bits. After round 16,L16 and R16 are swapped, so that the decryption algorithm has the same structure as the encryption algorithm. Finally, the block is gone through the inverse the permutation IP-1 and then output

**1.2.Test it by checking correctness of the particular transformations used:**

**1.2.1. Initial permutation:**

The plaintext block 64 bits pass through an initial permutation (IP) that the rearranged bits and produces the permuted input. Initial Permutation takes the plaintext as input. The table consists of 64 bits numbered from 0 to 63 *as shown below*.

**Table.1.2.1.a. Simple 64 bits table:**

[0,1, 2, 3, 4, 5, 6, 7,

8, 9, 10, 11, 12, 13, 14, 15,

16, 17, 18, 19, 20, 21, 22, 23,

24, 25, 26, 27, 28, 29, 30, 31,

32, 33, 34, 35, 36, 37, 38, 39,

40, 41,42, 43, 44, 45, 46, 47,

48, 49, 50, 51, 52, 53, 54, 55,

56, 57, 58, 59, 60, 61, 62, 63 ]

Then the initial permutation will be permuted input as 64 bits.

**Table.1.2.1.b.Initial Permutation table:**

ip = [57, 49, 41, 33, 25, 17, 9, 1,

59, 51, 43, 35, 27, 19, 11, 3,

61, 53, 45, 37, 29, 21, 13, 5,

63, 55, 47, 39, 31, 23, 15, 7,

56, 48, 40, 32, 24, 16, 8, 0,

58, 50, 42, 34, 26, 18, 10, 2,

60, 52, 44, 36, 28, 20, 12, 4,

62, 54, 46, 38, 30, 22, 14, 6 ]

**1.2.2.Inverse of the initial permutation:**

**Table. 1.2.2.a.The Inverse Initial Permutation table:**

IP^-1 = [39, 7, 47, 15, 55, 23, 63, 31,

38, 6, 46, 14, 54, 22, 62, 30,

37, 5, 45, 13, 53, 21, 61, 29,

36, 4, 44, 12, 52, 20, 60, 28,

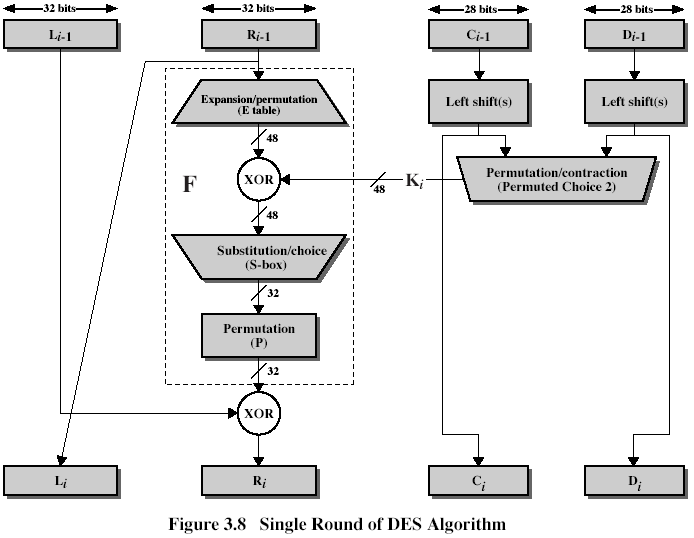
35, 3, 43, 11, 51, 19, 59, 27,

34, 2, 42, 10, 50, 18, 58, 26,

33, 1, 41, 9, 49, 17, 57, 25,

32, 0, 40, 8, 48, 16, 56, 24]

**Figure. 1.2.2.b.Single Round Des Algorithm:**



*As shown above how the single round des algorithm works.*

*As shown in the Figure.1.3.1.b.*The permuted input block split into two halves each is 32 bits. The first 32 bits are called L and the last 32 bits are called R. Expand R 32 bits to 48 bits to fit the sub key by preforming the Expansion permutation (E):

[**1.2.3. Expansion/Permutation**:](#_Toc465205989)

**Table.** **1.2.3.a.Expansion/Permutation (E table):**

expansion\_table = [

31, 0, 1, 2, 3, 4,

3, 4, 5, 6, 7, 8,

7, 8, 9, 10, 11, 12,

11, 12, 13, 14, 15, 16,

15, 16, 17, 18, 19, 20,

19, 20, 21, 22, 23, 24,

23, 24, 25, 26, 27, 28,

27, 28, 29, 30, 31, 0 ]

*The round key Ki is 48 bits. The R input is 32 bits. This R input is first expanded to 48 bits by Expansion/Permutation (E table).*

**1.2.4. Round key generation:**

The key generator algorithm takes 64 bits key as input. The input key number table from 1 to 64 is *as follows*:

* Input key has 64 bits. But each 8th bit is not used: bits 8,16,24,32,40,48,56,64 are not further used. The 56-bit key is first subjected to permutation **Permuted Choice 1.**

**Table. 1.2.4.a.8th bit Permuted Choice 1.**

[0,1, 2, 3, 4, 5, 6, 7,

8, 9, 10, 11, 12, 13, 14, 15,

16, 17, 18, 19, 20, 21, 22, 23,

24, 25, 26, 27, 28, 29, 30, 31,

32, 33, 34, 35, 36, 37, 38, 39,

40, 41,42, 43, 44, 45, 46, 47,

48, 49, 50, 51, 52, 53, 54, 55,

56, 57, 58, 59, 60, 61, 62, 63 ]

* Every eighth bit is ignored and produces 56 bits. *As shown in the table below.*

**Table. 1.2.4. c.Permuted Choice 1 without 8th bit:**

[0,1, 2, 3, 4, 5, 6,

7,8, 9, 10, 11, 12, 13,

14, 15,16, 17, 18, 19,20,

21, 22, 23,24, 25, 26, 27,

28, 29, 30, 31,32, 33, 34,

35, 36, 37, 38, 39,40, 41,

42, 43, 44, 45, 46, 47,48,

49,50, 51, 52, 53, 54, 55,

56, 57, 58, 59, 60, 61, 62 ]

**1.2.5.Permuted choice 1 :**

56 bits pass through a permutation Choice one (PC-1) and *displays as follows*:

**Table. 1.2.5.a.(PC-1):**

pc1 = [56, 48, 40, 32, 24, 16, 8,

0, 57, 49, 41, 33, 25, 17,

9, 1, 58, 50, 42, 34, 26,

18, 10, 2, 59, 51, 43, 35,

62, 54, 46, 38, 30, 22, 14,

6, 61, 53, 45, 37, 29, 21,

13, 5, 60, 52, 44, 36, 28,

20, 12, 4, 27, 19, 11, 3]

Code:

key = self.\_\_permutate(des.\_\_pc1, self.\_\_String\_to\_BitList(self.getKey()))

i = 0

# Split into Left and Right sections

self.L = key[:28]

self.R = key[28:]

while i < 16:

**1.2.6.Left circular shifts schedule :**

The output is separated into two 28 bits C and D. The first 28 bits are called  *C*0 (left part) and the last 28 bits are called *D*0. At each round, a circular left shift is performed on  *Ci*−1 and  *Di*−1 by 1 or 2 bits. *As shown in the table below*:

**Table 1.2.6.a Left circular shifts.**

left\_rotations = [1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1]

*These shifted values serve as input to the next round. They also serve as input to Permuted Choice 2, which produces a 48-bit output that serves as input to the function.*

Code for shift: # Perform circular left shifts

while j < des.\_\_left\_rotations[i]:

self.L.append(self.L[0])

print(self.L)

print("end of append left")

del self.L[0]

self.R.append(self.R[0])

print(self.R)

print("end of right")

del self.R[0]

j += 1

**1.2.7.Permuted choice 2:**

Then *Ci*−1 and  *Di*−1 in each round, passes through permutation choice two (PC-2) to produce 48 bits. The permutation Choice Two output in each round is uses as input to the encryption algorithm.

**Table.** **1.2.7.a.(PC-2).**

pc2 = [

13, 16, 10, 23, 0, 4,

2, 27, 14, 5, 20, 9,

22, 18, 11, 3, 25, 7,

15, 6, 26, 19, 12, 1,

40, 51, 30, 36, 46, 54,

29, 39, 50, 44, 32, 47,

43, 48, 38, 55, 33, 52,

45, 41, 49, 35, 28, 31

]

Code: # Create one of the 16 sub keys through pc2 permutation

self.Kn[i] = self.\_\_permutate(des.\_\_pc2, self.L + self.R)

print("mallory is deploying his crack")

print(self.Kn[i])

i += 1

**1.2.8.XOR with round key :**

The resulting 48 bits are XORed with Ki. This 48 bit result passes through a substitution function that produces 32-bit output, which is permuted by Permutation function (P):

p = [

15, 6, 19, 20, 28, 11,

27, 16, 0, 14, 22, 25,

4, 17, 30, 9, 1, 7,

23,13, 31, 26, 2, 8,

18, 12, 29, 5, 21, 10,

3, 24

]

**Code:**

# Exclusive or R[i - 1] with K[i], create B[1] to B[8] whilst here

self.R = list(map(lambda x, y: x ^ y, self.R, self.Kn[iteration])) //lambda is an anonymous function.

B = [self.R[:6], self.R[6:12], self.R[12:18], self.R[18:24], self.R[24:30], self.R[30:36], self.R[36:42], self.R[42:]]

# Optimization: Replaced below commented code with above

#j = 0

#B = []

#while j < len(self.R):

# self.R[j] = self.R[j] ^ self.Kn[iteration][j]

# j += 1

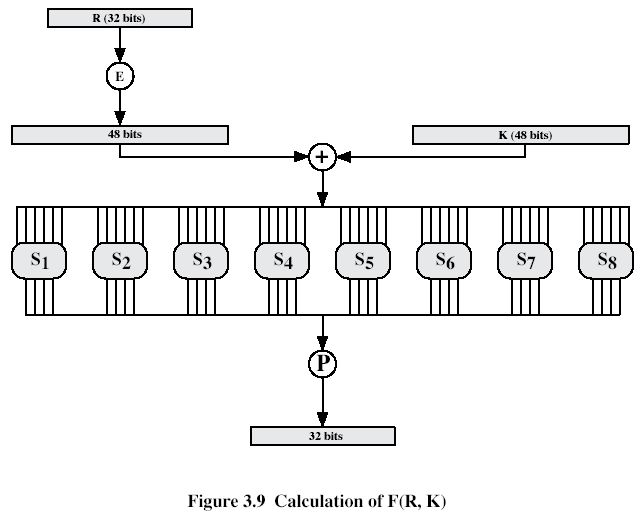
# if j % 6 == 0:

# B.append(self.R[j-6:j])

**1.2.9.S-boxes:**

It takes 6 bits as input and gives 4 bits as output. Break the result of E(*Ri*-1)⊕ *Ki* into 8 blocks, each containing 6 bits. These blocks are numbered from 1 to 8.

**Figure.** **1.2.9.a**.**Calculation of F(R,K):**



*8 "Substitution boxes" or S-boxes. Each S-box maps 6 bits to 4 bits*

**Table.1.2.9.b.Des S-Box:**

sbox = [

# S1

[14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7,

0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8,

4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0,

15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13],

# S2

[15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10,

3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5,

0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15,

13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9],

# S3

[10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8,

13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1,

13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7,

1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12],

# S4

[7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15,

13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9,

10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4,

3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14],

# S5

[2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9,

14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6,

4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14,

11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3],

# S6

[12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11,

10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8,

9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6,

4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13],

# S7

[4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1,

13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6,

1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2,

6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12],

# S8

[13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7,

1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2,

7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8,

2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11],

]

*Each row of an S-box defines a general reversible substitution: middle 4 bits of each group of 6-bit input are substituted by S-box output, 1st and last 6th bits define what particular substitution out of four to use.*

The first and the last bits of each block together as 2 bit value indicate the number of rows in the same number S-Box. The middle four bits of each block together as-bit value, indicating the number of columns in the same number S-Box. The decimal value, which is selected by the row and the column convert to-bit value in all S-Boxes.

**1.3.Permutation P after S-boxes:**

The resulting 48 bits are XORed with Ki. This 48 bit result passes through a substitution function that produces 32-bit output, which is permuted by Permutation function (P):

**Table.1.3.1.Permutation P after S-boxes:**

p = [

15, 6, 19, 20, 28, 11,

27, 16, 0, 14, 22, 25,

4, 17, 30, 9, 1, 7,

23,13, 31, 26, 2, 8,

18, 12, 29, 5, 21, 10,

3, 24

]

**1.3.2.XOR with left half:**

Perform Exclusive-OR between the output of the Permutation Function(P) and *Li*−1. Then, put the result in *Ri* , and put *Ri*−1 in *Li*. The overall formulas for DES Encryption Algorithm:  
*Li* = *Ri*−1.  
*Ri* = *Li*−1 ⊕ F(*Ri*−1,*Ki*).

As shown below in the form of code how X0R works.

# Permutate B[1] to B[8] using the S-Boxes

j = 0

Bn = [0] \* 32

pos = 0

while j < 8:

# Work out the offsets

m = (B[j][0] << 1) + B[j][5]

n = (B[j][1] << 3) + (B[j][2] << 2) + (B[j][3] << 1) + B[j][4]

# Find the permutation value

v = des.\_\_sbox[j][(m << 4) + n]

# Turn value into bits, add it to result: Bn

Bn[pos] = (v & 8) >> 3

Bn[pos + 1] = (v & 4) >> 2

Bn[pos + 2] = (v & 2) >> 1

Bn[pos + 3] = v & 1

pos += 4

j += 1

# Permutate the concatination of B[1] to B[8] (Bn)

self.R = self.\_\_permutate(des.\_\_p, Bn)

# Xor with L[i - 1]

self.R = list(map(lambda x, y: x ^ y, self.R, self.L))

# Optimization: This now replaces the below commented code

#j = 0

#while j < len(self.R):

# self.R[j] = self.R[j] ^ self.L[j]

# j += 1

# L[i] becomes R[i - 1]

self.L = tempR.3

**1.3.3.Swap of the halves:**

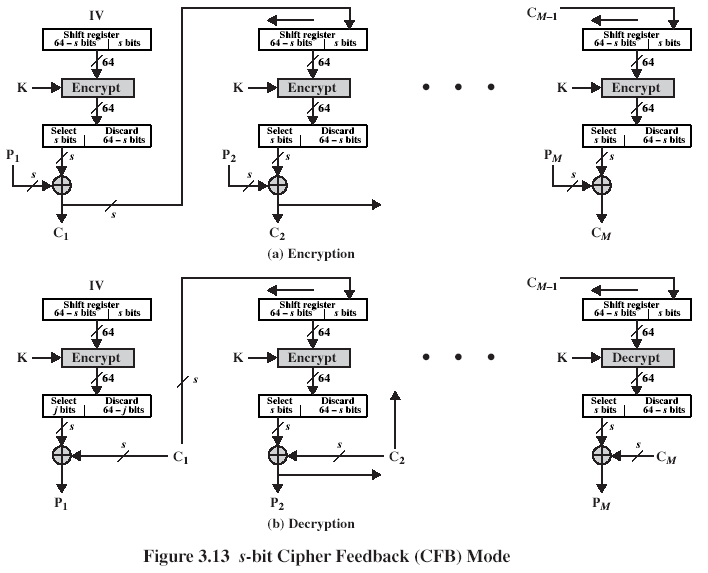
Perform a 32-bit swap on the result of the final round. Then, perform Inverse Initial Permutation (IP−1) on the swapped data to produces the cipher text 64 bits.64-bits are produced after the swap on the final round, after performing the inverse of the initial permutation. key and plain text has been entered which shows the output performed on32-bit swap resulting in the output of 64 bits and showing encrypted text.

**1.3.4.Padding:**

The value of each pad byte is the total number of bytes that are added. Of course, the total number of pad bytes depends on the block size. For example, if the message is 3 bytes shorter than an integer multiple of the block size, then 3 pad bytes should be added, each of them of value 3. If 5 bytes should be added, then each of them should be 5.Furthermore, If the message is longer than an integer multiple of the block size, then subtraction is made and according to how longer it is number of bytes are subtracted.

1.3.4.Cipher Feedback (CFB) mode:

Figure.1.3.4.a.s-bite Cipher Feedback mode:



*Input is processed s bits at a time. Preceding cipher-text is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of cipher-text.*

*As Shown in the Figure.1.3.4.a.* it is assumed that the unit of transmission is s bits; usually, s=8. As with CBC, the units of plaintext are chained together, so that the ciphertext of any plaintext unit is a function of all the preceding plaintext. In this case, rather than units of 64 bits, the plaintext is divided into segments of s bits. Consider encryption. The input to the encryption function is a 64-bit shift register that is initially set to some initialization vector (IV). The leftmost (most significant) s bits of the output of the encryption function are XORed with the first segment of plaintext P1 to produce the first unit of cipher-text C1, which is then transmitted. In addition, the contents of the shift register are shifted left by s bits and C1 is placed in the rightmost (least significant) s bits of the shift register. This process continues until all plaintext units have been encrypted. For decryption, the same scheme is used except that the received cipher-text unit is XORed with the output of the encryption function to produce the plaintext unit.

**2.Tools Description :**

* **Vim (Text Editor):**

One of the most powerful text editors accessible from the command line is the vim editor. Built on the foundation of "vi". It can be found (or can be installed) on every type of system. Vim is like a “note pad” which is found in systems that use windows. There is not installation or preparation usage for vim as it is by default already inside the Linux.

* **Pyuic4 (Converter):**

Pyuic4 is a converter which converts xml to python.

* **Qt designer:**

Qt designer is an interface designer. it has a simple and intuitive interface, features a code editor with syntax highlighting and auto-completion, drag-and-design UI creation, visual debugging & profiling tools and many other tools to support the project. Website to download/install from “<https://www.qt.io/>”.

**3.Developed Program Description:**

* Installation requires the packages like “python-qt4-sql libqt4-sql-psql “
* Requirements: Python 2.7.

**4.User Guide :**

* Unrar the file in a folder
* Run login.py to execute the program
* **Packages(for linux):**
* “apt-get install python-qt4-sql libqt4-sql-psql”
* The package should include SQlite and pyQt.
* **Database used: “**SQlite”
* **Librarie:** base64, byqt4, binascii, python-qt4-sql libqt4-sql-psql, OS sys, pyqt4.qtsql=> qsqlquerymodel, qsqldatabase, qsqlquery

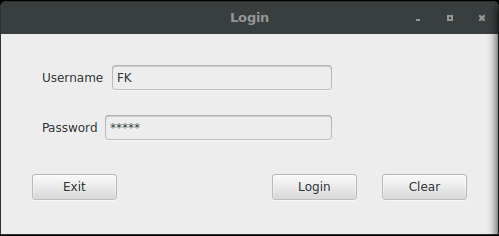
**5.Conducted Tests Description:**

**5.1.Mandatory Access Control (MAC):**

BLP model is derived from the military multilevel security paradigm (Top secret, Secret, Confidential, Unclassified).Each document has 1 out of 4 security levels, and each user has “clearance”, also 1 out of 4.A document of a certain level can be accessed only by users with the same or higher clearance level, “no read-up” rule. A user can write only in the documents of his or higher level of security, “no write down” rule, ‘\*’ property

**5.2.Administrator:**

* **Login Panel (As Administrator):**

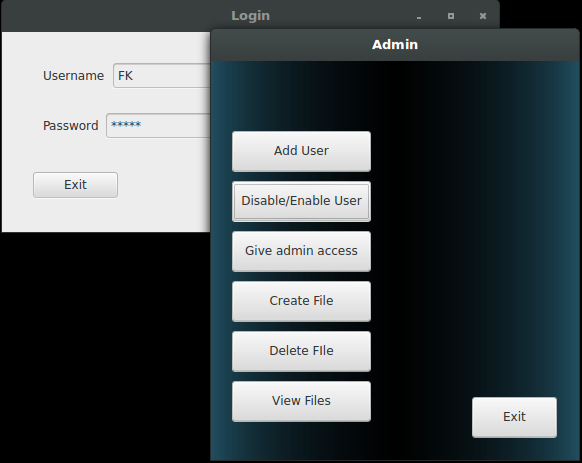
**Figure.5.2.a.Login Panel**

*As shown above, This login panel is for both user and admin to login into their panels.*

*As shown in the Figure.5.2.a.*The login panel allows users/administrator to gain access to system/files. This login panel allows users to enter into user’s login panel and allows administrators to enter into the administrators login panel. The user/administrator enters their username and password to login into the system.

* **Administrator Panel:**

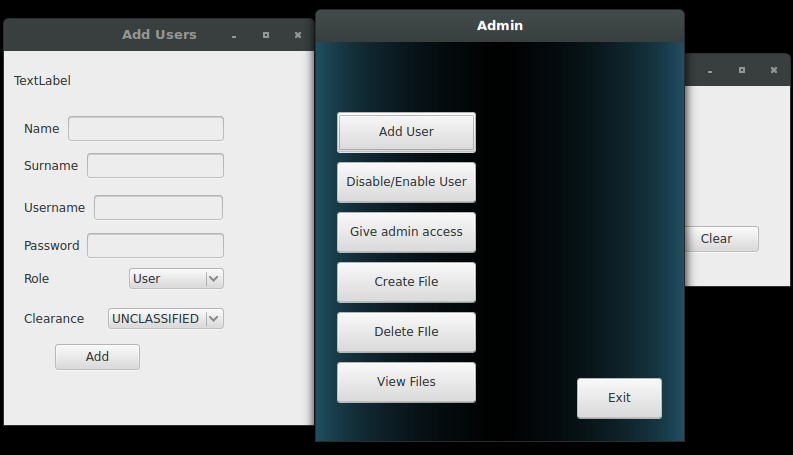
**Figure.5.2.b.Administrator Panel**

****

*As shown above, when the admin enters his username and password. The admin enters into his panel .*

*As shown in the Figure.5.2.b.*The admin enters into administrator panel. The admin can add user, disable/enable users, grant/revoke access of documents to the users. Furthermore, admin can create/delete files and view files of his/her own and of everyone.

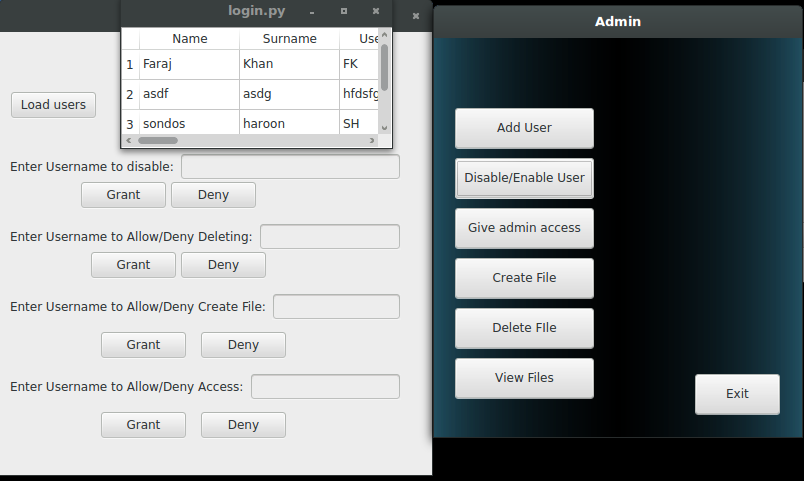
* **Introducing a new user:**
* **Creating new user:**

**Figure.5.2.1.a.Introducing new user:**

*As shown above, the admin can add new users to the system by filling some details about the user.*

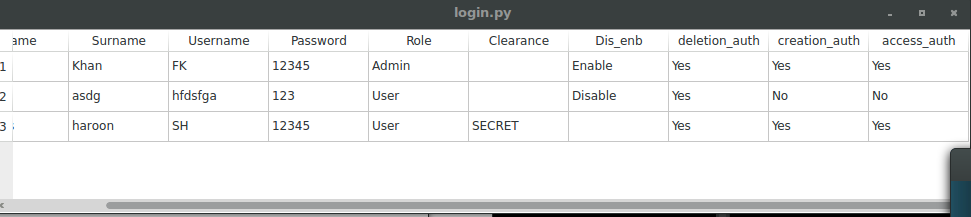
*As shown in the Figure.5.2.1.a.*The admin can create new users into the system by giving some user details like name, surname, username and unique password to each of the user and set the role of user to handle the system either as user or as an admin. Moreover, Admin sets the clearance level of the user like Top secret, secret, confidential, unclassified. Clearance level allow the user to access files either of the same level or higher level and to write either if on the same level or higher level.

* **Disabling/enabling a user:**

**Figure.5.2.3.a.Disable/Enable User:**

*As shown above the admin can enable/disable user from the system and grant/revoke access to the deletion of files, creation of files and access to the documents.*

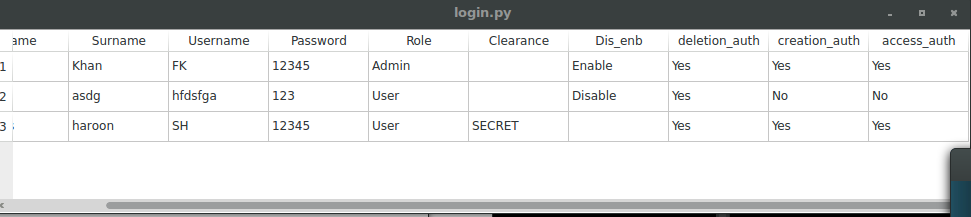
*As shown in the Figure.5.3.1.a.*The admin can enable/disable user from the system means can lock the user to enter the system or login again or free the user. Furthermore, admin can give and take back permission of deleting, creating, and accessing files from the users.

* **Enable User:**

**Figure.5.2.3.b.Enabling user:**

*As shown above the user is enable. Which mean he/she can enter/login into the system.*

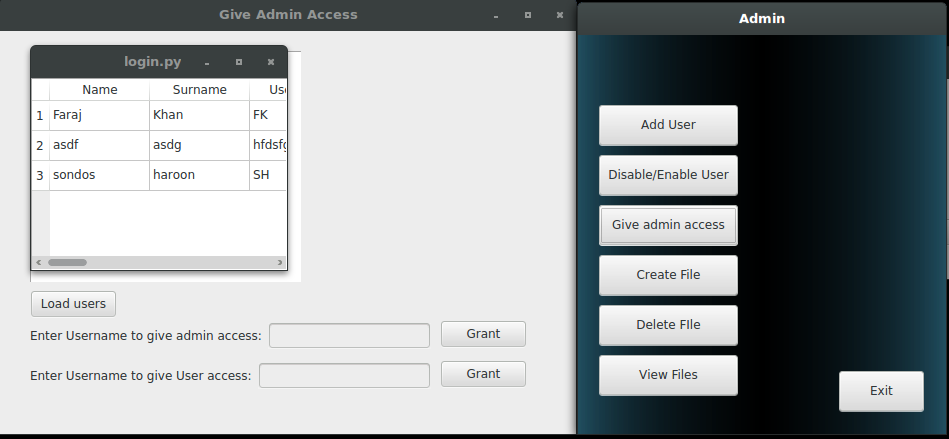
* **Disable User:**

**Figure.5.2.3.c.Disabling Users.**

*As shown above the user is disabled. Which mean he/she cannot enter/login into the system.*

* **Giving/revoking users rights:**

**Figure.5.2.4.a.Grant/revoke users rights:**

****

*As shown above the admin can give admin access to the users by entering username.*

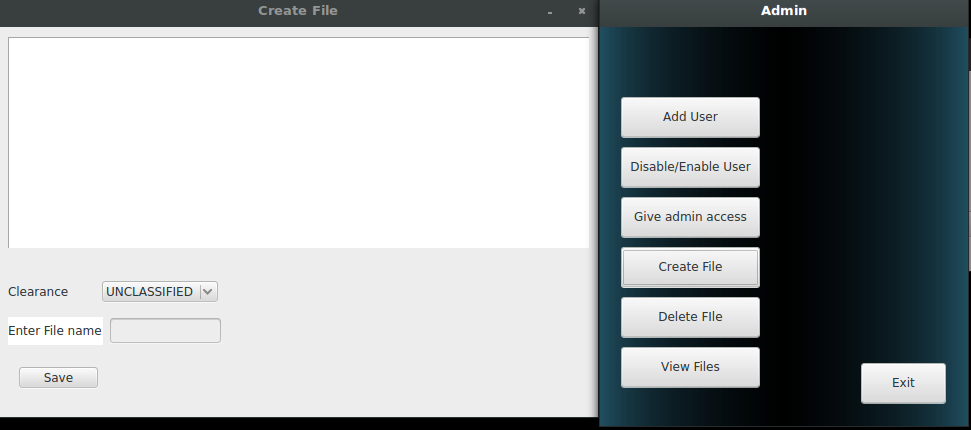
* **Granting/Giving:**

*As shown in the Figure.5.2.4.a.*Granting means, giving permission or allowing the access to the files/documents of others users. When admin gives the admin access to the user then user can log in into the admin log in panel as an admin and can do stuff like creating/deleting, enable/disable other users and view files of others user.

* **Revoking:**

*As shown in the Figure.5.2.4.a* .Revoke means, taking back or cancelling. By means the admin can change the user from admin access to user access where user log in into system as user and cannot view files of everyone or enable/disable other users.

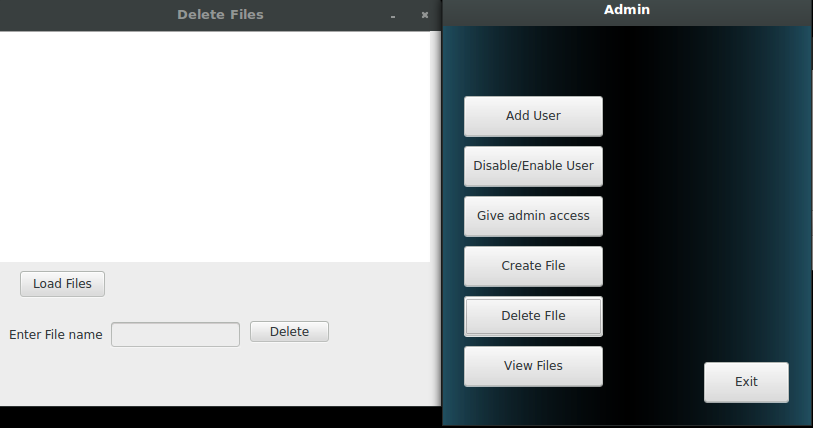
* **Administrator Rights:**
* **Creating object:**

**Figure.5.2.5.a.Creating Objects:**

*As shown above the admin can create file in the create file text box.*

*As shown in the Figure.5.2.5.a.*The admin can create files in the text box he/she can write and give file a number by entering in the “Enter File name” text box and set the clearance level of file to top secret, secret, confidential or unclassified. So only same or higher level may read the file.

* **Deleting object:**

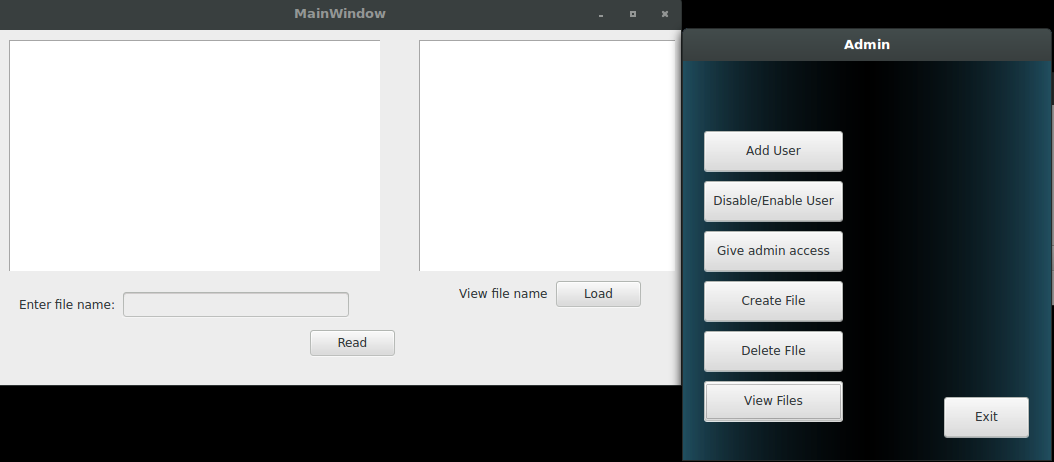
**Figure.5.2.5.b.Deleting Objects:**

*As shown above the admin can delete files in the delete files text box.*

*As shown in the Figure.5.2.5.b.*The admin can delete files by entering file name in the “Enter File name” text box and pressing delete tab. Admin can view files name by load files button. The admin can delete files of anyone as admin wish of any users.

* **View Files:**

**Figure.5.2.6.a.View Files:**



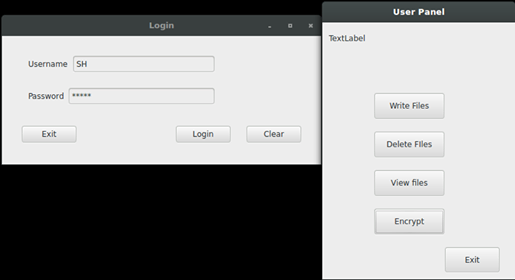
*As shown above the admin can view files in the view files box by entering file name.*

*As shown in the Figure.5.2.6.a.*The admin can read files and views files name by entering the file name in the “enter file name” text box and clicking on “read”. The admin can view files by “load” button to see overall files in the system.

**5.3.User:**

* **Login Panel(User):**

**Figure.5.3.1.a.User panel:**

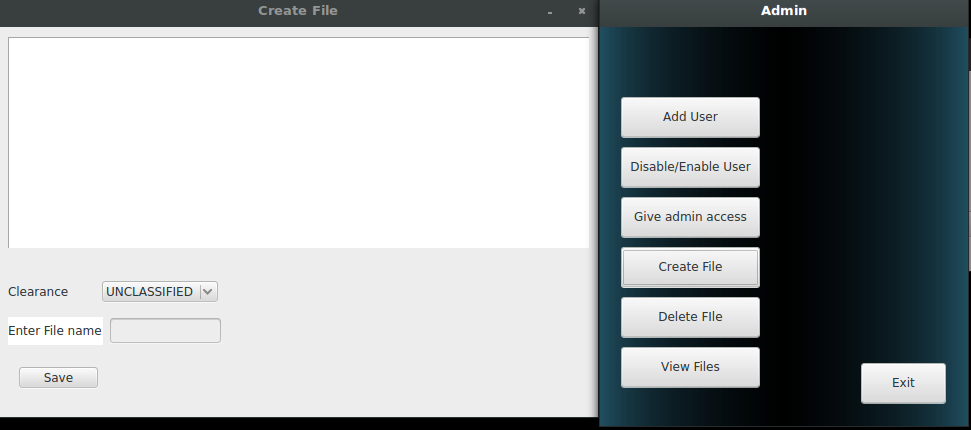
****

*As shown above, when the user enters his/her username and password. The user enters into user panel .*

*As shown in the Figure.5.3.1.a.*The user enters into user panel. The user can create files , delete files and view files of his/her own and of the higher level.

* **Accessing objects:**
* **Creating object:**

**Figure.5.3.2.a.Creating Objects:**

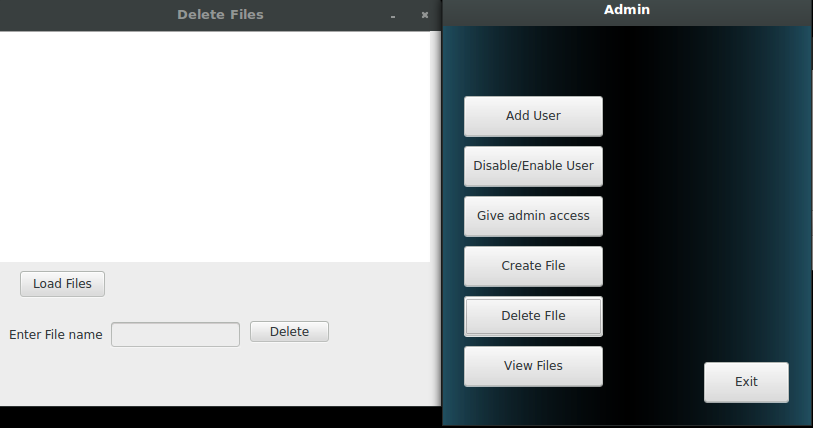
****

*As shown above the user can create file in the create file text box.*

*As shown in the Figure.5.2.5.a.*The user can create files in the text box he/she can write and give file a number by entering in the “Enter File name” text box and set the clearance level of file to top secret, secret, confidential or unclassified. So only same or higher level may read the file.

* **Deleting object:**

**Figure.5.3.2.b.Deleting Objects:**

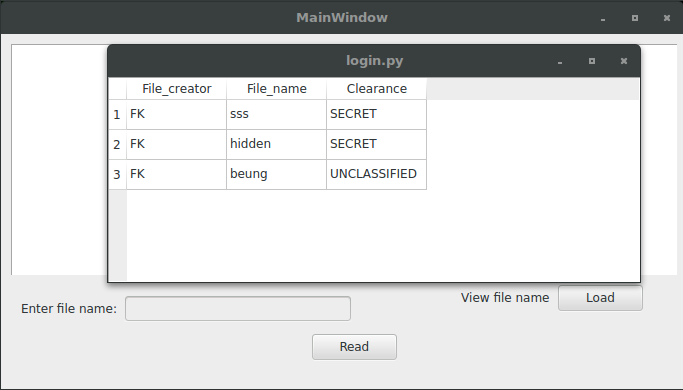
****

*As shown above the user can delete files in the delete files text box.*

*As shown in the Figure.5.3.2.b.*The user can delete files by entering file name in the “Enter File name” text box and pressing delete tab. The user can only delete his/her file “not of any other user”.

* **Viewing object:**

**Figure.5.3.2.c.Viewing Objects:**

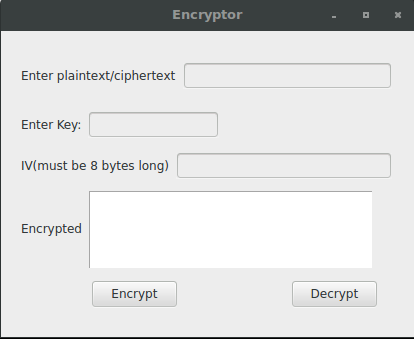
****

*As shown above the user can view files in the view files box by entering file name.*

*As shown in the Figure.5.3.2.c.*The user can read files and views files name by entering the file name in the “enter file name” text box and clicking on “read”. The user can view files by “load” button, user will only be able to see his/her level files with the files of same level or higher level to the user.

* **Encrypting/decrypting:**

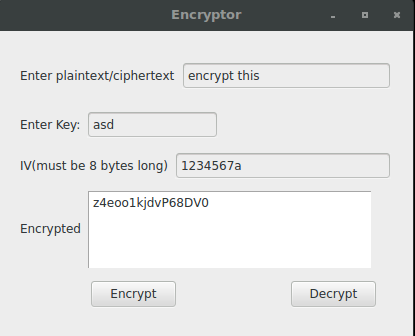
**Figure.5.3.3.a.Ecncryptor box:**

****

*As shown above is the encryptor box for the des Cipher Feedback mode.*

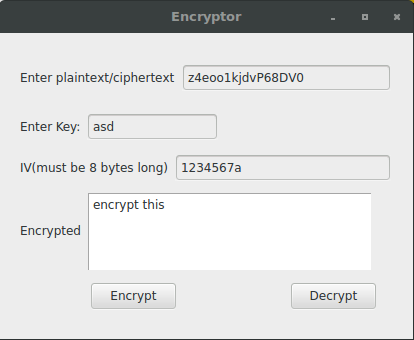
*As shown in the Figure.5.3.3.a.*In plaintext box a simple plain text is entered and a key of 8 bytes and initial value (IV) 8 bytes exactly no more than or less than 8 bytes otherwise padding is used.

**Figure.5.3.3.b.Encrypted text:**

****

*As shown above after entering the plain text, key and Iv the Decrypted/Encrypted text box shows the encrypted text when clicked on encrypt tab.*

**Figure.5.3.3.c.Decrypted text:**

****

*As shown above Decrypted/Encrypted text box shows the decrypted text after putting the encrypted text in the plain/encrypted text box and pressing to the decrypt tab. The encrypted text then gets decrypted.*

**References:**

[**https://docs.python.org/3/library/sqlite3.html**](https://docs.python.org/3/library/sqlite3.html)

[**https://docs.python.org/2/library/binascii.html**](https://docs.python.org/2/library/binascii.html)

**Conclusion:**

In Conclusion, this report shows the understanding of selected problem variant from ACM (Mandatory Access Control) and DES (Cipher Feedback) mode. Full implementation is done on the solution of the problem as a software application and usage of appropriate tools which has been used .Furthermore, conducted tests of the application are shown. Moreover, knowledge of algorithm used for problem solving, tools and details are all shown .The program works fully fine and all requirements are fulfilled.