

Assignment 5

COMP 7401 - Feistel Cipher with CBC

Filip Gutica - A00781910

Shahin Nikvand - A00809275

Table of Contents

COMP 7401 - Feistel Cipher	0
Table of Contents	1
Introduction	2
Constraints	2
Action Plan	2
Pseudo Code	2
Encryption	2
Decryption	3
Instructions	4
Testing	4
Summary	4

Introduction

The purpose of this assignment is to explore the Feistel cipher and also to help us understand how DES and Triple DES function. It teaches us about multi round ciphers but also the importance of the algorithm and method used to encrypt data for easy encryption and decryption with the right key but making it near impossible to reverse.

Constraints

We will be limiting our development to the following:

- Python as the development language
- 8 Rounds of the feistel cipher
- Using the same key for all the rounds for ECB
- Do subkey generation for CBC
- Being able to encrypt and decrypt a few paragraphs of plain text

Action Plan

Development Plans are as follows:

1. Create a simple “scramble” function which will hide the data while following the rules of the feistel cipher and be easy enough to encrypt and decrypt
2. Execute the encrypt by hand on paper to verify the validity of the approach and that indeed is reversible and functional
3. Write Pseudo of how we initially intend to execute the idea
 - a. Please note that our actual code does differ from this but this was the initial approach to the application

Pseudo Code

This is our initial plan of approach to writing the application. As we continued our development it resulted in slight changes from this to better follow along with the code.

Additionally we are assuming that efficiency and speed are not top priority so we will be using array which will cause additional data. The garbage collector should handle the removal of the data but this is a statement that there are better ways to execute this code.

Encryption

Assumptions

- 8 rounds
- 64 bit blocks (8 bytes)
- Use sub key generation for CBC

Formulas:

1. $f(x, k) = [(2i * k)^x] \% (2^{32} - 1)$
2. $L_i = R_i - 1$
3. $R_i = L_{[i-1]} \text{ XOR } f(R_{[i-1]}, k_i)$

Function Encrypt(Plaintext, key)

Key initial = key

Initialize empty string Ciphertext

For every 64 bit block of the plaintext

Initialize array L[8]

Initialize array R[8]

Set L[0] to first 32 bits of plaintext block

Set R[0] to second 32 bits of plaintext block

For i = 1 to max rounds

L[i] = R[i - 1]

If mode is cbc

If first round

Key = key initial

Else key = subkeygen(L[i], key initial)

R[i] = L[i - 1] XOR Scramble(R[i - 1], i, key)

// Add together the final results of each L and R (Lfin and Rfin)

//Append to the Ciphertext

Ciphertext += (L[8] + R[8])

Return Ciphertext

Function Scramble(x, i, k)

Return $((x * k)^i) \% (2^{32} - 1)$

Function SubkeyGen(s1, s2)

sha256(s1 + s2)

Decryption

```
Function Decrypt(Ciphertext, Key)
    Key_initial = key
    Initialize empty string
    For every 64 bit block of the ciphertext
        Array L[8]
        Array R[8]
        L[0] first 32 bit
        R[0] second 32 bit
        For i = 8 to 1
            If mode is cbc
                Key = subkeygen(L[i], key_initial)
                If i is 1
                    Key = key_initial
                R[i+1] = L[i]
                L[i+1] = R[i] Scramble(L[i ], i, key)
            Ciphertext = L[8] + R[8]
    Return ciphertext
```

```
Function Scramble(x, i, k)
    Return ((xi * k)^i) % (2^32-1)
```

```
Function SubkeyGen(s1, s2)
    sha256(s1 + s2)
```

Instructions

We have two modules: feistel.py and feistel-decrypt.py

Feistel.py is run using python version 2.

To encrypt a file please run:

Feistel.py -e -m <cbc,ecb> -t <plaintext file> -k <key> -o <ciphertext file>

Feistel-decrypt.py is run using python version2.

To decrypt a file please run:

Feistel.py -d -m <cbc,ecb> -t <ciphertext file> -k <key> -o <resulting plaintext file>

Please make sure you have the input files for both feistel.py and feistel-decrypt.py in the same directory as the script.

Testing

Test	Test Case	Expected Result	Actual Result	Pass/Fail
1.	Run fesistel.py with key: password, on file plaintext.txt with output file ciphertext.txt	Run with no errors, produce ciphertext file: ciphertext.txt	Ran with no errors. Produced ciphertext file: ciphertext.txt	Pass
2.	Run feistel-decrypt.py with key: password, on file ciphertext.txt with output file result.txt	Run with no errors, produce the original plaintext to result.txt	Ran with no errors, produce the original plaintext to result.txt	Pass
3.	Run feistel-decrypt.py with wrong key: passwordp, on file ciphertext.txt with output file result.txt	Run with no errors, produce result.txt file still encrypted	Run with no errors, produce result.txt file still encrypted	Pass

Plaintext used:

Test 1

```
/BCIT/COMP7401/COMP7401_Shared/A4/src — Atom

result.txt      ciphertext.txt  plaintext.txt

1 Alice was beginning to get very tired of sitting by her sister
2 on the bank, and of having nothing to do: once or twice she had
3 peeped into the book her sister was reading, but it had no
4 pictures or conversations in it, 'and what is the use of a book,'
5 thought Alice 'without pictures or conversation?'
6
7 So she was considering in her own mind (as well as she could,
8 for the hot day made her feel very sleepy and stupid), whether
9 the pleasure of making a daisy-chain would be worth the trouble
10 of getting up and picking the daisies, when suddenly a White
11 Rabbit with pink eyes ran close by her.
12
```

Run feisteil.py

```
src: bash — Konsole

File Edit View Bookmarks Settings Help

filip@filip-Inspiron-7537:~/BCIT/COMP7401/COMP7401_Shared/A4/src$ python feisteil.py -t plaintext.txt -k
password -o ciphertext.txt
filip@filip-Inspiron-7537:~/BCIT/COMP7401/COMP7401_Shared/A4/src$
```

No errors. Resulting Ciphertext:

```
~/BCIT/COMP7401/COMP7401_Shared/A4/src — Atom

result.txt      ciphertext.txt  plaintext.txt

1 Czx0a80nc00n00000Jx05j0"d000o00R|00<#000v000)00b000wt00l00,um00d0Xif0T0sh
2 0hX/dc00m00{00%h0073Z|=/|btb[0u00x0{s0'j7-_{0_00co0zH0`0#`035,s0>'n00g0l0s000,06.(00a3GuX0N0s00Rc>0m`}0cd#0J}000p0'/00k0|jai
3 d-c4`0i0z#uw/02`0le0a050|i )j000f%?To00,y/0;
4 UEv000q00-h0]0e%00800zy0AD0040|0n00iK0`%000xT0og/0xb 0Zj00mh2!vc00f00.hp0s8-nt"00jn|0=5g!0tgZGc000x0Zg00c-00sd%0\u00tWg0 0p
5 0q000xLTi0uMqF:0!0l0*400%
6 *0
```

Test 2

Run feistel-decrypt.py

```
src : bash — Konsole
File Edit View Bookmarks Settings Help
filip@filip-Inspiron-7537:~/BCIT/COMP7401/COMP7401_Shared/A4/src$ python feistel-decrypt.py -t ciphertext
t.txt -k password -o result.txt
filip@filip-Inspiron-7537:~/BCIT/COMP7401/COMP7401_Shared/A4/src$
```

No Errors, Resulting plaintext:

```
BCIT/COMP7401/COMP7401_Shared/A4/src — Atom

result.txt      ciphertext.txt  plaintext.txt

1 Alice was beginning to get very tired of sitting by her sister
2 on the bank, and of having nothing to do: once or twice she had
3 peeped into the book her sister was reading, but it had no
4 pictures or conversations in it, 'and what is the use of a book,'
5 thought Alice 'without pictures or conversation?'
6
7 So she was considering in her own mind (as well as she could,
8 for the hot day made her feel very sleepy and stupid), whether
9 the pleasure of making a daisy-chain would be worth the trouble
10 of getting up and picking the daisies, when suddenly a white
11 Rabbit with pink eyes ran close by her.
12
```

Test 3

Run decrypt with wrong password

```
src : bash — Konsole
File Edit View Bookmarks Settings Help
filip@filip-Inspiron-7537:~/BCIT/COMP7401/COMP7401_Shared/A4/src$ python feistel-decrypt.py -t ciphertext
t.txt -k passwordp -o result.txt
filip@filip-Inspiron-7537:~/BCIT/COMP7401/COMP7401_Shared/A4/src$
```

Result.txt still encrypted:

```
result.txt      ciphertext.txt  plaintext.txt

1 @00{030a=0vZ00cBc0}\b;tg]@n02Rw#70mN0050
2 0o00l?0g0rM=000-f00fH'0v0yr=-0A0s0_10T04$Z0g:0w00f00e000z0rpy0begNB00k00#
3 0q(00ib0x&j00p00p#;0#)0Z/z00|.00k|0j|0r0 NgN3wo000n0s9jL-rj^0iY(20Kq07y+b1(0e0SuY0~
4 00100051i0j?!(KETQ0xw+0q0j00u4000q0000al0y730uE00)z0Ag00-e00{.0Nm00jy0qt01x+(0e00p00un0_c0g00Dc00:0h00LBp;r0*0r?R0
5 0gk_0|k96g0Eo0i0L0C0d_)0[0h0bv0 05a00m00(^*!g0e&'0#0,y0%/z^0c001px0t0Qma000+00300Y20#s]
```


Summary

Writing the code for the feistel cipher made us realize how brilliant yet simple the cipher is. It's incredible to see how you have achieve such complexity and security with such a simple code. CBC is very powerful and provides great security at the loss of error propegation.