CS4003701 資訊安全導論 Introduction to Information Security
HW2 Report

Team members : B10832018 官澔(encrypt)、B10832008 蔡芸軒(decrypt)

[Structure tree]

```
110-Information-Security
| ----HW1
| ----HW2
      | ---- encrypt.py
      | ---- decrypt.py
      | ---- Report.pdf
| ----HW3
```

[encrypt]

The encryption separate the key to 16 48-bits keys. First we do the permutation to the plaintext. Then cut the plaintext to half. Then the two parts of plaintext will be encrypted by turns for 16 times. The right side of the plaintext will be switch to the left and the left one will do the `XOR` operation with the result which produced by the `F_function`. `F_function` first expand the 32-bit data into 48-bit. Then do the `XOR` operation with the separated 48-bit keys. The result will put in the `S-box` to change the 6-bit to 4-bit data.Finally do the `p-permutation` and output the 32-bit result.

The process is basically the same as the decryption. The only thing should be changed is the order of using the separated keys.

[decrypt]

First, I do the function `generate_keys` to generate the 16 48-bits subkeys that would be used later, permuting the initial key according to the table `PC-1`, then cutting it into left and right (28 bits each). In each 16 turns, shift each 1 or 2 bit left, then concat them and permute it according to the table `PC-2`.

After subkeys are prepared, I do the initial `IP_permutation` of the 64 bits initial cipher, (this step is the reverse of the final step in `encryption)`, then cutting it into left and right (32 bits each). After that, repeat 16 times

```
Li+1 = Ri
Ri+1 = Li xor F(Ri, Ki)
```

The keys were used from 16th to 1th since I'm doing decryption, the process is reversed.

In `F_function`, I first do `E_permutation` to input, and xor the result to the key (48 bits). After that, cut the result into 8 parts(6 bits each), and use `S_box` to reach the new 4 bits results.

Finally, do the `IP_reverse_permutation`, which is the reverse of the first step of encryption, and then return the final result !