

CS4003701 資訊安全導論 Introduction to Information Security

HW2 Report

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

[Structure tree]

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

[encrypt]

[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

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \text{ xor } F(R_i, K_i)$$

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 !