CSE350: Network Security
Assignment 2

Mohammed Kaif - 2020084
Arya Sinha - 2020498

Project - 0 : DES encryption and decryption

Program structure
The DES implementation consists of 2 helper modules and a main.py file -

1) ***helper.py*** - Contains F-box implementation

   Global variables
   - (*list*) **exp_d** - Permutation table to perform expansion permutation from 32 bits to 48 bits
   - (*list*(*list*)) **s_box** - List of 8 s-boxes
   - (*list*) **p_table** - Straight permutation table to permute 32-bit input
   - (*list*) **initial_permutation** - Permutation table to permute initial 64-bit input
   - (*list*) **final_perm** - Permutation table to permute the final swapped output at the end of 16 rounds. This is the reverse of the initial permutation table.

   Functions
   - **expand_permutation ()**
     - Parameters: ((*str*) right_half)
     - Return Value: (*str*) expanded_right_half
     - Definition: This is the expansion P-box. 32-bit input is expanded to 48 bits.
   - **xor ()**
     - Parameters: ((*str*) op1, (*str*) op2)
     - Return Value: (*str*) ans
     - Definition: Here, expanded 48-bit input is xor-ed with 48-bit subkey.
   - **sbox_substitution ()**
     - Parameters: ((*str*) inp)
     - Return Value: (*str*) result
     - Definition: Pre-defined 8 s-boxes each receive a 6-bit input and provide the corresponding 4-bit output
   - **permutation ()**
     - Parameters: ((*str*) inp)
     - Return Value: (*str*) permuted_result
     - Definition: This is the straight P-box. It permutes 32-bit input using a fixed P-box.

2) ***subkey_gen.py***

   Function:

   - **generate_subkeys ()**
     - Parameters: ((*str*) key)
     - Return Value: (*list)* subkeys
     - Definition: Generates a list of 16 48-bit subkeys from the given 64-bit key.

3) ***main.py***

   Functions

   - **bin2str ()**
     - Parameters: ((*str*) text)
     - Return Value: (*str*) string
     - Definition: Converts a binary string to utf-8 encoded string.
   - **str2bin ()**
     - Parameters: ((*str*) data)
     - Return Value: (*str*) permuted_result
     - Definition: Converts utf-8 string to binary string.
   - **encrypt ()**
     - Parameters: ((*string*) binary_data, (*list*) subkeys, (*bool*) flag=True)
     - Return Value: (*str*) cipher
     - Definition: To perform encryption on the input data using the DES (Data Encryption Standard) algorithm. This involves an initial permutation, 16 rounds of encryption, and a final permutation.
   - **decrypt ()**
     - Parameters: ((*string*) cipher, (*list*) subkeys)
     - Return Value: (*str*) data
     - Definition: It first reverses the order of the subkeys, then uses the "encrypt" function to decrypt the cipher by performing the same operations as in encryption, but in reverse order.

Assumptions:

- Plaintext and key are always 64-bit long.

Code Explanation:
We first define 64-bit key and plaintext (in this case, 8 *1-byte* characters) and convert them to binary using the *str2bin()* function. Then we generate 16 subkeys using the key schedule algorithm. The *generate_subkeys()* function used for this is present in the subkey_gen.py. The key scheduling algorithm includes parity-bit dropping using the *pc1_table*, left shifting according to the round number, and compression using the *pc2_table*. The plaintext is then encrypted

using the generated 16 subkeys. The *encrypt()* function takes in binary data and a list of subkeys as input and performs the DES encryption process.
The input data is first permuted using an initial permutation table present in the helper.py file. Then, 16 rounds of encryption are performed. In each round, the right half of the input data is expanded and permuted (*expand_permutation()*), then XORed with the current subkey (*xor()*). The result is then substituted using S-boxes (*sbox_substitution()*) and permuted again using a fixed permutation table. The left half of the input data is then XORed with this result. The left and right halves are then combined and swapped. After 16 rounds, a final permutation is performed on the output. The S-box, permutation tables, and functions are all defined in the helper.py file for simplicity.

The *decrypt()* function takes in the cipher generated by the encrypt function and the list of subkeys and performs the DES decryption algorithm on the cipher using the subkeys. It simply calls the encrypt function with the cipher and the subkeys in reverse order.

Steps performed:

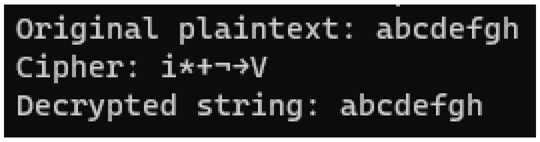
1. Defining 64-bit key and plaintext
2. Converting key and plaintext to binary data
3. Subkeys generation using Key scheduling algorithm
4. Plaintext encryption
    a. Initial permutation
    b. 16 rounds of encryption
        i. Expand Permutation
        ii. XOR with subkey
        iii. S-box substitution
        iv. P-box permutation
        v. XOR with left half and swap
    c. Final Permutation
5. Decryption
    a. Reverse subkeys
    b. Perform encryption on ciphertext using reversed subkeys to get the plaintext

Sample inputs and outputs

key = “qwertyui”

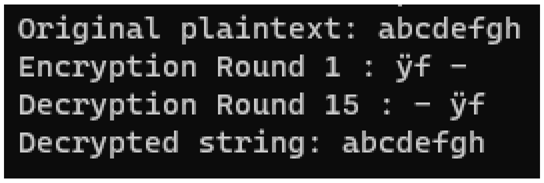
1. data = “abcdefgh”

a) Following cipher and decrypted texts are produced



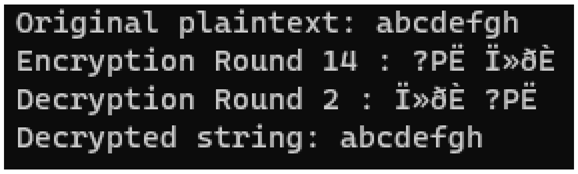
```
Original plaintext: abcdefgh
Cipher: i*+¬→V
Decrypted string: abcdefgh
```

b) 1st encryption round and 15th decryption round



```
Original plaintext: abcdefgh
Encryption Round 1 : ÿf –
Decryption Round 15 : – ÿf
Decrypted string: abcdefgh
```

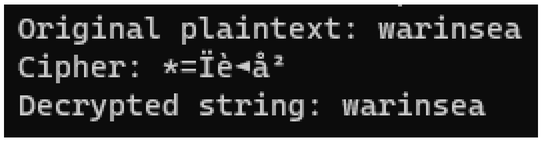
c) 14th encryption round and 2nd decryption round



```
Original plaintext: abcdefgh
Encryption Round 14 : ?PË Ï»ðÈ
Decryption Round 2 : Ï»ðÈ ?PË
Decrypted string: abcdefgh
```

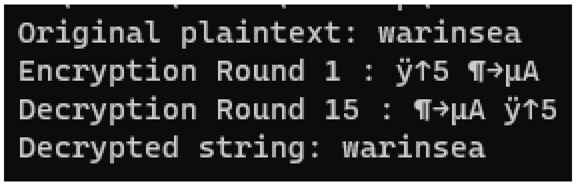
2. data = “warinsea”

a) Following cipher and decrypted texts are produced



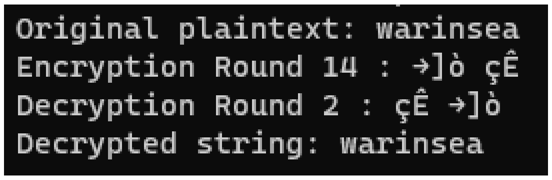
```
Original plaintext: warinsea
Cipher: *=Ïè◄å²
Decrypted string: warinsea
```

b) 1st encryption round and 15th decryption round



```
Original plaintext: warinsea
Encryption Round 1 : ÿ↑5 ¶→µA
Decryption Round 15 : ¶→µA ÿ↑5
Decrypted string: warinsea
```

c) 14th encryption round and 2nd decryption round



```
Original plaintext: warinsea
Encryption Round 14 : →]ò çÊ
Decryption Round 2 : çÊ →]ò
Decrypted string: warinsea
```

Thus, it is verified that :

a) The cipher text when decrypted yields the original plain text.
b) The left half of the output of the 1st round of encryption is the same as the right half of the 15th round of decryption and vice versa
c) The left half of the output of the 14th round of encryption is the same as the right half of the 2nd round of decryption and vice versa