ASSIGNMENT NO. 7

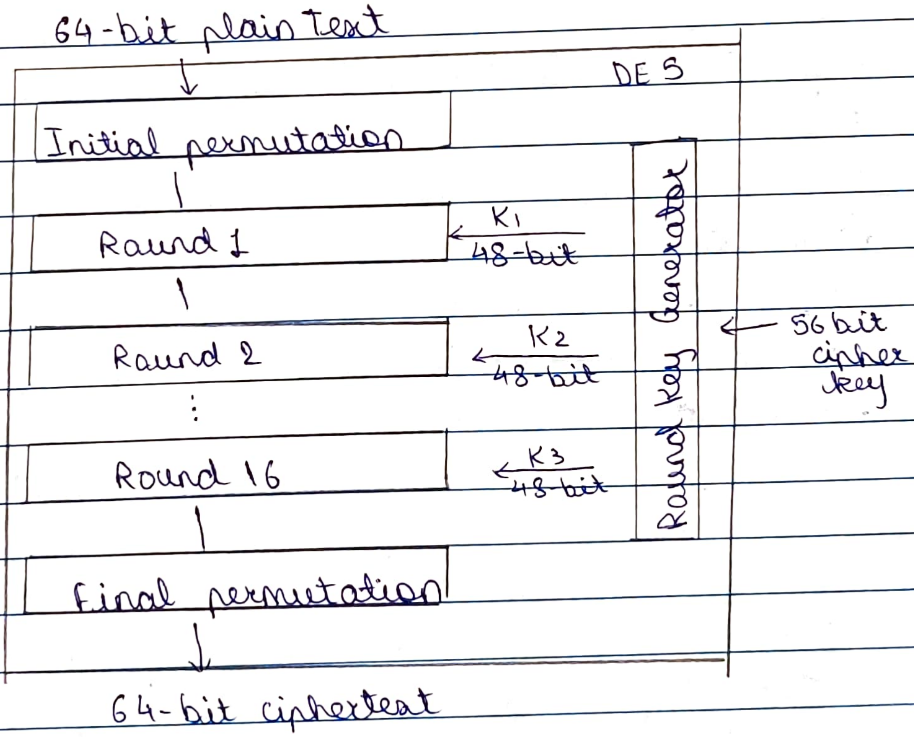
* Aim

Implement a client and server on different computers using python. Perform the encryption of message of sender between these two entities by using DES Algo and use Diffie Hellman method for exchange of keys.

* Theory

The Data Encryption Standard is a symmetric key block cipher. It uses 16 round Feistel structure. The block size is 64-bit, but has effective key length of 56 bits.

General structure of DES:



All that is required to specify DES is

- Round Function
- Key Schedule
- Any additional processing - initial & final permutation

Initial and Final Permutation

These are straight permutation boxes (P-boxes) that are inverse of each other. They have no cryptography significance in DES.

Round Function

The heart of cipher is the DES function, f. The DES function applies a 48-bit key to the rightmost 32-bits to produce a 32-bit output.

Expansion Permutation Box

Since right input is 32-bit and round key is 48-bit, we first need to expand right input to 48 bits.

- XOR (Whitener) -

  After the expansion permutation, DES does XOR operation on the expanded right section and the round key.
- Substitution Boxes -

  These S-boxes carry out real mixing. DES uses 8 Sbox each with 6-bit input and 4-bit output.
- Straight Permutation -

  The 32-bit output of S-boxes is then subjected to a straight permutation. The round key generator creates sixteen 48-bit keys out of 56 bit cipher key.

DES Analysis

It satisfies both desired properties of block cipher.

- Avalanche effect :- A small change in plaintext results in the very great change in ciphertext
- Completeness:- Each bit of ciphertext depends on many bits of plaintext

Diffie Hellman Key Exchange

This algorithm is used to exchange the secret key between the sender and receiver. This algorithm also satisfies the exchange without actually transmitting it.

Algorithm :

Let,

Private key of sender X_s

Public key of sender Y_s

Private key of receiver X_r

Public key of receiver Y_r

Steps,

1) One of the parties choose two numbers 'a' and 'n' and exchange with other party. After this exchange, both the parties know the value of 'a' and 'n'.

2) Both parties already know their private key. They calculate value of public key and exchange.

Sender calculates public key as $Y_s = a^{X_s} \bmod n$

Receiver calculates public key as $Y_r = a^{X_r} \bmod n$

3) Both parties receive public key of each other. They calculate value of secret key.

Sender calculates secret key as $= (Y_r)^{X_s} \bmod n$

Receiver calculates secret key as $= (Y_s)^{X_r} \bmod n$