Report

CSE432, Network Security

Name: Ahmed Mohammed Salah AbdelAziz ID: 1600171

Project No: (2)

The main.py file contains a major class DES() which has all the needed functions to perform the DES encryption algorithm:

First: the class contains some members needed in the algorithm which are the s-box, permutation choice 1, permutation choice 2, initial and final permutation matrices, all these matrices are needed to perform the encryption.

Second: the class also contains the functions:

“permutaion_choice1”: to perform the pc1 step in the algorithm using “pc1 matrix”.

“left_circular_shift”: to perform the left circular shift giving the amount to shift.

“permutaion_choice2”: to perform the pc2 step in the algorithm using “pc2 matrix”.

“initial_permutation”: to perform the initial permutation step in the plain text.

“initial_permutation_inverse”: to perform the final permutation step at the end.

“expansion”: to expand the R of the text into 48 bits.

“sbox”: uses s-boxes to convert 48 bits to 32 bits.

“permutation”: performs another permutation.

“F”: where the magic happens, first expand the R then perform Xor with the key then give the output to s-box finally do the permutation step

“round”: performs the DES round.

“swap”: swaps the L and R after the final round

“generate_key”: takes 64 bit key and generates 16 keys each one is 48 bits one for each round.

“encrypt”: performs the encryption, does the 16 rounds of the DES.

“decrypt”: performs the decryption, does the same 16 round with the keys reversed.

The main.py file also contains two auxiliary functions:

“prepare_text”: takes the 16 hexa-decimal characters and converts them to binary.

“return_text”: takes a binary text and converts it to 16 hexa-decimal characters.

Finally the main part in the file takes the plain text, the key and the number of times to perform the DES encryption, also it takes a value 1 or 0, 1 for encryption and 0 for decryption.