Q1 Team Name
0 Points

Group Name

ela

Q2 Commands
5 Points

List all the commands in sequence used from the start screen of this level to the end of the level. (Use -> to separate the commands)

go->jump->dive->back->pull->back->back->enter->wave->back->back->thrnxxtzy->read->the_magic_of_wand->c->read->password->c->tjwyhydplx

Q3 Cryptosystem
10 Points

What cryptosystem was used at this level? Please be precise.'

6-Round DES

Q4 Analysis
80 Points

Knowing which cryptosystem has been used at this level, give a detailed description of the

cryptanalysis used to figure out the password. (Use Latex wherever required. If your solution is not readable, you will lose marks. If necessary, the file upload option in this question must be used TO SHARE IMAGES ONLY.)

After getting the wand, using the above mentioned command, we wrote 'read' and got the message by the spirit. We said 'password' and got $krjlmpjukfsupngljhrkmqpsfrmosgui$ as the ciphertext password.

It was told by the spirit that the system is 4-round or 6-round DES. Next, we saw that we could send some inputs and we get the encrypted text. There was no text of input of 2 alphabets. However, the output was of constant length of 16 alphabets for input of 3 to 18 length of alphabets. This suggested that encryption was happening in a group of (18 - 3) +1 = 16 length of alphabets.

This confirmed for block cipher and DES encryption. Now, spirit talked about 2 letters being 1 byte, which means one letter is 4 bits. 16 alphabets can be represented from that. Upon careful analysis of the output texts, it was found that it only contained alphabets from f to u, which are 16 in number. The mapping is {f,g,h,....,u} -> {0,1,2,...,15} in binary.

Next next task is to generate input.
Here, we are using 2 different type of characteristics to decrpyt the 6 round des(using the reference "https://medium.com/lotus-fruit/breaking-des-using-differential-cryptanalysis-958e8118ff41") .
1st characteristics:
(40080000,04000000 , 1/4 ,04000000,00000000, 1, 00000000,04000000, 1/4 ,04000000,40080000)
2nd characteristics:
(00200008,00000400 , 1/4 ,00000400,00000000, 1, 00000000,00000400, 1/4 ,00000400,00200008)
From gen_input_pair1.py, we generate 10000 pairs of input for each of the characteristics such that input xor after initial permutation with give the xor value (40080000 04000000) or (00200008 00000400) depending upon the characteristics.
Now, we use make_script.py to generate a shript1.sh file taking the input pairs. Then, using command.sh we generated an output1.txt file which after cleaning using "ciphertext_cleaned1.ipynb" gives us "cipher.txt" which is the corresponding ciphertext pairs wrt

to input pairs.
To find the Key:
Considered the 3-round characteristic:
(40080000,04000000 , 1/4 ,04000000,00000000, 1, 00000000,04000000, 1/4 ,04000000,40080000).
Found that XOR of the input to S-box in Round 4 = E(R3)^E(R3').
Observed that the XOR of the input corresponding to S-boxes S2,S5,S6,S7,S8 is zero, so the XOR of output of these S-boxes will also be zero.
Found the XOR of the output of S-boxes in Round 6 by using the XOR of output of permutation in Round 6.
Used the inverse permutation to find the XOR of the input to S-boxes in Round 6.
Found the Round 6 key, K = (K1, K2, ..., K8) by using the methods discussed in class.
Hence,
To find the K2, K5, K6, K7, K8 using the first 3-round characteristic, 5000 input pairs were generated with an XOR of 40080000 04000000 (after initial permutation) and
The corresponding ciphers were obtained using above techniques specified.
Then, using the function "find_the_Round6_Key" we find the set of possible values of keys K1,K2,K3,K4,K5,K6,K7,K8 and for 5000 pairs ,we find the most occurring value. So we get ,
K1=45, K2=59, K3=37,K4=7,K5=12,K6=61,K7=50,K8=53
similarly, we have done for Second 3-round characteristic: (00200008,00000400 , 1/4 ,00000400,00000000, 1, 00000000,00000400, 1/4 ,00000400,00200008).
we get:
K1=45, K2=59, K3=37,K4=7,K5=12,K6=24,K7=15,K8=53
only two bits are fluctuating here, but then we saw there frequencies, for 1st characteristics K6 and K7 frequencies are very comparable to the average but these bits frequencies are very much higher in the second characteristics.
So, we just tried to take all the 48 bits of round 6 key to be as same as that of 2nd characteristics due to their high frequencies and then for final key we use key_schedule1.cpp giving the idea of how the bits are permuted and then we use key_set1.cpp to map to the final_key.
Then, we applied brute force on rest of these 8 bits and got the key final key using the function "find_the_key" in code1.cpp and finally we decrpyt the password we got "tjwyhydplx000000".