Q1 Team Name
0 Points

Group Name

da_vinci

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 -> dive -> dive -> back -> pull -> c-> go -> back -> enter ->
wave -> c-> back -> back -> thrnxxtzy -> read ->
the_magic_of_wand -> c -> read -> password -> c ->
pquxdbegfs
```

Q3 Cryptosystem
10 Points

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

6 Round DES(Data Encryption Standard)

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.)

We obtained ciphertext as **tkmnsomlkktikprpgkgjtikiriqrkkll** (after using read and then password command) that needs to be decrypted using des and then used as a command to clear level 4.

To crack 6-round DES, we used a chosen plaintext attack. The attacker creates a number of plaintexts in this kind of cryptanalysis attack, persuades the sender to encrypt them nevertheless, and then uses the collected pairings of plaintexts and ciphertexts to determine the encryption key.

Following functions are used in the DES algorith

IP(M): This is applied on the plaintext M which is to be encrypted. E(M): Expnad 32-bits of text M to 48-bits.
P(M): This step permutes the 32-bit input M.
S: there are 8 S-boxes. Each S-boxes has 6-bit input and a 4-bit output.
PC1: key permutation that maps 64 bits of keys to 56 bits and removes the parity bits.
shift- a shift that is performed on the key obtained as the output of PC1.
PC2: key permutation that maps 56 bits of shift's output to 48 bits.
IP INV(M): This is applied after all 6 rounds of DES are done on message M

Procedure followed to break the 6 round DES:

● Differential cryptanalysis was performed by us using two 3-round characteristics and for cryptanalysis of 6-round DES we used a chosen-plaintext attack. The characteristics used are 40080000 04000000($characteristic1$) and 00200008 00000400($characteristic2$).
● Since one byte is made up of two letters, one letter is represented by 4 bits.
● Only 16 letters can be represented using 4 bits. In order to determine which 16 letters are used in the game for this level, we experimented with a number of plaintexts and examined the corresponding ciphertexts.
● We deduced that the game for this level uses alphabets

from f to u after performing the analysis on the ciphertexts. As a result, we assigned numbers 0 to 15 to the letters from f to u respectively:
$\{f : 0000, g : 0001, h : 0010, i : 0011, j : 0100, k : 0101, l : 0110, m : 0111, n : 1000, o : 1001, p : 1010, q : 1011, r : 1100, s : 1101, t : 1110, u : 1111\}$
The input and output size of one DES block is 64 bits i.e. 8 bytes (block size) which means 16 $(i.e., 64/4 = 16)$ letters. Thus, we decided to generate the plaintexts of size 16 letters.

Step 1:Generating plaintexts:

The differential characteristics 0020000800000400 and 4008000004000000 with probabilities of $1/16$ each are used. To crack the 6-round DES, we created 2500 pairs of plaintexts and ciphertexts corresponding to each characteristic. The first 2500 plaintext pairs are created with an XOR of 0000801000004000 obtained by applying inverse initial permutation to the characteristic 400800000000000, and the remaining 2500 plaintext pairs with an XOR of 0000080100100000 obtained by applying inverse initial permutation to the characteristic 0020000800000400 These inputs are kept in separate $plaintexts1.txt$ and $plaintexts2.txt$ files.The code for the generation of plaintext pairs is in $generate\_plaintext.ipynb$.

Step 2: Obtaining Ciphertexts corresponding to

The task of collecting ciphertexts corresponding to the plaintexts from the server was automated and we also Python's pexpect for establishing a connection to the server. To generate the ciphertexts we used $plain\_to\_cipher1.py$, corresponding to plaintexts stored in $plaintexts1.txt$ and $plain\_to\_cipher2.py$ to generate the ciphertexts corresponding to plaintexts stored in $plaintexts2.txt$. The resultant ciphertexts are stored in $ciphertexts1.txt$ and $ciphertexts2.txt$ respectively.

Step 3: Finding the key bits of round key K6:

We carried out steps from 1 to 4 for the ciphertexts obtained corresponding to each of the two characteristics.

$\#1$: We used the mapping of letters defined above to convert the obtained ciphertext to binary and then, we used $Differential\_CryptAnalysis.ipynb$ to apply to reverse the final permutation on these binary ciphertexts to get (L_6, R_6) and$(L_6^{'}, R_6^{'})$ which is the output of 6^{th} round of DES. We know that,$R_5 = L_6$.Thus, using the values R_5 and $R_5^{'}$, we computed the output of the Expansion box and input XOR of S-boxes for 6^{th} round.

$\#2$: For the first characteristic mentioned above L_5, =04000000 and for the second characteristic L_5 =00000400. We found the output of the permutation box by performing $L_5 \oplus (R_6 \oplus R_6^{'})$, then we applied inverse permutation on this value to obtain output XOR of S-boxes for 6^{th}round.

$\#3$: Let $E(R_5) = \alpha_1\alpha_2......\alpha_8$ and $E(R_5^{'}) = \alpha_1^{'}\alpha_2^{'}......\alpha_8^{'}$ and $\beta_i = \alpha_i \oplus k_{6,i}$ and $\beta_i^{'} = \alpha_i^{'} \oplus k_{6,i}$, where $|\alpha_i| = 6 = |\alpha_i^{'}|$ and . At this point, we know $\alpha_i, \alpha_i^{'}, \beta_i \oplus \beta_i^{'} and \gamma_i \oplus \gamma_i^{'}$ .We created an 8 * 64 key matrix to
store the number of times a key k ∈ [1,64], satisfies the possibility of being a key to $S_i$box, where i ∈ [1,8].

$\#4$: We computed the set X_i=$(\beta, \beta^{'})$ | $\beta \oplus \beta^{'}$=$\beta_i \oplus \beta_i^{'}$ and $S(\beta_i) = \gamma_i \oplus \gamma_i^{'}$. Then we found the key k , such that $\alpha_i \oplus k = \beta and (\beta, \beta^{'}) \in X_i$ for some $\beta^{'}$. For all the keys k which satisfied this condition for S_i box, we incremented their count in the key matrix i.e., key_matrix[i][k] was incremented.

●After performing the above analysis to find the keys, we obtained the following results for characteristic 4008000004000000:

S-box	Max	Mean	Key	Diff
S1	338	174	45	164
S2	789	200	51	589
S3	306	165	37	141
S4	280	165	7	115
S5	401	178	60	223

S6	746	192	56	554
S7	445	171	13	274
S8	452	176	54	276

For the above characteristic, in round 4, XOR will be zero for S2, S5, S6, S7 and S8. Therefore, in round 6 these S-boxes will give the corresponding key bits of K_6. It can also be observed that a significant difference is there in the maximum key frequency and mean key frequency for these S-boxes which further assures of these key values are correct. We proceeded by taking the key bits for S2, S5, S6, S7 and S8 boxes as 51, 60, 56, 13 and 54 respectively.

●The above analysis gave the following results for characteristic 0020000800000400:

S-box	Max	Mean	Key	Diff
S1	408	171	45	237
S2	417	179	51	238
S3	314	170	37	144
S4	731	205	7	526
S5	443	175	60	268
S6	749	194	56	555
S7	283	159	13	124
S8	247	164	54	83

For the above characteristic, in round 4, XOR will be zero for S1, S2, S4, S5 and S6. Thus, in round 6 these S-boxes will give the corresponding key bits of K_6 . Also, it can be observed that a significant difference is seen in the maximum key frequency and mean key frequency for these S-boxes. We proceeded by taking the key bits for S1, S2, S4, S5 and S6 boxes as 45, 51, 7,60 and 56 respectively.

Both the characteristics have S2, S5 and S6 as common Sboxes and we obtained the same key values for these three Sboxes which further verified that our computations so far are
correct. Therefore, we proceeded by taking key values for S1, S2, S4,S5, S6, S7 and S8 as 45, 51, 7, 60, 56, 13 and 54 for round key K_6.Thus, at this point we know 42 bits of the 56 bit key

Step 4: Find the Actual Key from 42 known bits:

●Next, we applied key scheduling algorithm to obtain the actual positions of these known 42 bits in the 56 bit key and obtained the following result: Masterkey=X11XX1XX01011X100XX11X11000X0000111X01111000X11X1111X001, here X denotes unknown bits.

●At this point we have 14 unknown bits and for these 14 unknown bits of DES key, we iterate through all 2^{14} possible permutations of the key to find the correct key. We took plaintext =$fghijklmnopqrstu$ and the corresponding ciphertext= $piihftijqhlksnsp$ and performed 6-round DES encryption. The key which encrypts this plaintext to produce the correct ciphertext is the final key. From this step, we obtained the following key which satisfied the above results:

$\#$ The final 56 bit key is
011011100101111001111011000000001110011110001111
$\#$ The 48-bit round key for all 6 rounds was generated as follows:
$Round1$ key is
111011000100111100000111010101011111111101010100
$Round2$ key is
011011110011011101100010001110001111111100000110
$Round3$ key is
111010101101010011101101111110100101010010110111
$Round4$ key is
110110011100001101011010100011010110101110101010
$Round5$ key is
001001001101101110111011101101100111101001010000
$Round6$ key is
101101110011100101000111111100111000001101110111

Step 5: Decrypting password(ciphertext):

●The ciphertext corresponding to our password is `tkmnsomlkktikprpgkgjtikiriqrkkll` and thus to obtain the password we performed decryption on this ciphertext.

● This ciphertext has 32 letters. Since each letter is represented by 4 bits, this is 128-bit string, that is, 2 blocks of DES ciphertext. As per our mapping, this is {229,120,217,118,85,227,90,202,21,20,227,83,195,188,85,102}

●Now that we have our key, we perform decryption on this ciphertext by considering 16 letters(=64 bits) at a time using `des.cpp`, which uses decryption function of DES implementation for 6 rounds.

●The plaintext obtained is - **pquxdbegfs000000**. We removed the zeroes as they might have been used for padding.

●We entered the plaintext **‘pquxdbegfs’** in the game and were directed to the next level. This is the code to clear the level.

Q5 Password

5 Points

What was the password used to clear this level?

pquxdbegfs

Q6 Code

0 Points

Please add your code here. It is MANDATORY.

da_vinci_Assignment_4.zip Download

```
Large file hidden. You can download it using the button above.
```

Assignment 4

● Graded

Group

ANSHUL SHARMA
SUMIT KUMAR CHAUDHARY
PRADEEP CHALOTRA
View or edit group

Total Points

68 / 100 pts

Question 1

Team Name **0** / 0 pts

Question 2

Commands **5** / 5 pts

Question 3

Cryptosystem **10** / 10 pts

Question 4

Analysis **48** / 80 pts

Question 5

Password **5** / 5 pts

Question 6

Code **0** / 0 pts