### Q1 Team Name
**0 Points**

Group Name

Hustler

### 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)

enter->dive->dive->back->pull->go->back->enter->wave->back->back->thrnxxtzy->read->the_magic_of_wand->c>read->password->c->fghijklmnopqrstu->c->tqsxiefohf->c

### Q3 Cryptosystem
**10 Points**

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

6 Round DES with blocksize = 64 bits and key size = 56 bits using DIFFERENTIAL CRYPTANALYSIS (chosen plaintext attack) and frequency analysis for deducing f to u mapping.

### 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.)

Here, we are using Chosen plain text attack to break DES encryption. We use differential cryptanalysis to generate plain text pairs, pass them to shellScript.py to get corresponding ciphertext pairs and find key using these and finally decrypt the above ciphertext.

From the Level 4 screen, after entering  we go through the commands "enter"->"dive"->"dive"->"back"->"pull"->"go"->"back"->"enter"->"wave"->"back"->"back"->"thrnxxtzy"->"read"->"the_magic_of_wand"->"c">"read".After entering "password" we get ciphertext as "jlrrpgoiimrkurgpmkruotrgrfujuhfl".

Further from Spirit's hint, we understood that it is either 4-round or 6-round DES. Also, the chances of it being a 10-round DES were less, so we proceeded with a 6-round DES ( We need a 4-round- characteristic to break a 6-round DES, as discussed in the lecture.).
Also, from the hint "two letters for one byte" we figured out that each letter is represented as 4 bits (1 byte - 8 bits, 2 letters in 1 byte => 1 letter 4 bits), therefore only 16 (16 * 4 = 64 bits) out of 26 letters are possible.

After trying out random plaintexts as input, we observed that the cipher text has letters from f to u(16 alphabets) using frequency analysis. so while generating the plain text we used letters from [f-0000,u-1111].

IDEA:
The cryptanalysis of DES with six rounds is more complex. We use two statistical Characteristics
with probability 1/16 = 0.0625  and choose the key value that is counted most often.
Each one of the two characteristics helps us find the key bits of which are used at the input of Sboxes in the sixth round, so the total number of key bits found by the two characteristics is 42. The other 14 key bits can be found later by means of a Brute force search.

The two characteristics used are 40080000 04000000 and 00200008 00000400.

STEP :
1. Using Block 2 from the file attached -> differentialanalysis.ipynb, we generated 5000 pairs of plaintext for each characteristic. We have used characteristics 40080000 04000000 and 00200008 00000400. We have generated 5000 pairs of plaintext satisfying characteristic 40080000 04000000 and ensured that their xor value is 00008010 00004000, which is obtained using inverse initial permutation on the characteristic, and have stored it in PlainTexts1.txt.
Similarly, we have used the same approach for characteristic 00200008 00000400 and ensured the xor value as 000008001 00100000, stored it in PlainTexts2.txt.

2. Now, we have generated ciphertexts corresponding to the stored plaintexts by executing Block3 from the file attached(ran in Linux environment as shellScript.py) -> differentialanalysis.ipynb and have stored them in CipherTexts1.txt and CipherTexts2.txt.

3. Then we perform differential cryptanalysis to find the key. We read CipherTexts1.txt and for each ciphertext, we convert the letter into binary using mapping f-0000, g-0001....u-1111.

IDEA:
We apply inverse final permutation to get (L6, R6) and (L'6, R'6). We know that R5=L6, so we find R5 and R'5 to find the output of the expansion box and input xor of s boxes for the 6th round.

Now, L5=04000000 for 1st characteristic and L5 = 00000400 for the second characteristic. We perform L5 XOR (R6 XOR R'6), then apply inverse permutation to get xor of s-boxes for the 6th round.

As discussed in class,
Let
$T(R5) = \alpha_1\alpha_2 \cdots \alpha_8$ and $T(R5') = \alpha'_1\alpha'_2 \cdots \alpha'_8$
where
$|\alpha_i| = |\alpha'_i| = 6$
and
$k_6 = k_{6,1}k_{6,2} \cdots k_{6,8}$
and
$\beta_i = \alpha_i \oplus k_{6,i}$ and $\beta'_i = \alpha'_i \oplus k_{6,i}$
At this point, we know
$\alpha_i, \alpha'_i, \beta_i \oplus \beta'_i$ and $\gamma_i \oplus \gamma'_i$

A key matrix of 8 by 64 is used to store how many times a particular key k can be a key to the s-box.

We find the set
$X_i = \{(\beta, \beta') \mid \beta \oplus \beta' = \beta_i \oplus \beta'_i \text{ and } S_i(\beta) \oplus S_i(\beta') = \gamma$
Then for each K in [1,64] , we check whether
$\alpha_i \oplus k = \beta$ and $(\beta, \beta') \in X_i$ for some $\beta'$

if this condition is satisfied for the S box, then we increment key[i][k] by 1 (Block 5).

Result of the above analysis for characteristics 40080000 04000000, we get the partial key using S8, S7, S6, S5, and S2 as 50,0,31,6,59 as input to S boxes in 0 to round 4.

We do a similar procedure for ciphertexts in CipherTexts2.txt. The result of the above analysis for characteristic 00200008 00000400 is that we get the partial key using S6, S5, S4, S2, and S1 as 31,6,7,59,45 as input to s boxes is 0 in round 4.

Here we have S6, S5, and S2 as common and key bits from both the characteristic are the same for before mentioned s boxes. Hence we found 42 out of 56 bits of the key.

The 48-bit key for the Sbox is 111101111011XXXXXX000111101000011001011111110100. Here there are 6 X's in S3 which was never 0. Now converting this 56-bit key and applying key schedule PC2 (Block 4), we

get
X11XX1XX01011X100XX11X11101X0110101X01001001X10X11
11X001.

Now to find the missing bits we use the brute force method, we iterate through 2^14 keys. We pass "fghijklmnopqrstu" as input to the shell server of assignment, we get cipher as "ssonrhgkhprsknt n ". Then for each possible key, we encrypt the plaintext with this key to check if we got the above cipher. The key with which the output of encryption and the above cipher matches is the actual key.
The key to it is
01101110010111100111101110100110101001001001010111
111001

4. We convert our password "jlrrpgoiimrkurgpmkruotrgrfujuhfl" first into binary and then into decimal and divided into two parts as at a time DES only works on 8 bytes of plaintext, to get {70,204,161,147,55,197,252,26} and {117,207,158,193,192,244,242,6} where each block is 8 bytes and this passes one at a time in Block 7.

After decryption we got,

tqsxiefohf000000.

We deduced that 000000 (padding) that is added to the decrypted password, And after entering "tqsxiefohf" we got the level cleared.

## Q5 Password
**5 Points**

What was the password used to clear this level?

tqsxiefohf

### Q6 Code
**0 Points**

Please add your code here. It is MANDATORY.

**Assignment4.zip** Download

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

## Assignment 4

Graded

**Group**

ABHIVADAN
SHRUTIKA ANIL JADHAV
SANDULA LAVANYA
View or edit group

**Total Points**

**90 / 100 pts**

**Question 1**

Team Name — **0** / 0 pts

**Question 2**

Commands — **5** / 5 pts

**Question 3**

Cryptosystem — **10** / 10 pts

**Question 4**

Analysis — R **70** / 80 pts

**Question 5**

Password — **5** / 5 pts

**Question 6**

Code — **0** / 0 pts