Q1 Team Name

0 Points

Enciphered

Q2 Commands

10 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 -> go -> wave -> back -> back -> thrnxxtzy -> read -> 134721542097659029845273957 -> c -> read-> password -> poptdzfims

Q3 CryptoSystem

5 Points

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

DES (block cipher) of 6 rounds.

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.)

Introduction: The foremost task was to retrieve the wand from river bottom. It was followed by going back and freeing up the spirit in 3rd level. Thereafter, we went to first screen and typed "read" and got to see the instruction from the spirit. We saw the hints on the glass pane and as per the instructions, we typed "password" to go further. Finally, we got the ciphertext - **jnrkdgqefggnglmromdppogkhfnkreie** . The last task was to decrypt this ciphertext in order to pass this level.

When we typed "read" on first screen of level 4, we got to see the whisper of the spirit. This message contained sufficient hints for us to determine that the encryption method used here was DES. Moreover, it also mentioned that the encryption could be 4-round or 6-round but not 10-round. It was mentioned explicitly that this encryption method is not 10-round.

Firstly we assumed it to be a 6-round DES and tried to break it using chosen plaintext attack. In this attack methodology for cryptanalysis, we created a number of samples of plaintexts, then got the encrypted form of each particular item and then used the obtained pairs of plaintexts and ciphertexts to find the key used for encryption.

IP(M) - This is applied on the plaintext M that is to be encrypted.
IP_INV (M) - This is applied after all 6 rounds of DES are done on message M.
E (M) - Expand 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-box has 6-bit input and a 4-bit output.
PC1 - Key permutation that maps 64 bits of key to 56 bits and removes the parity bits
Shift - Shift that is performed on the key obtained as output of PC1
PC2 - Key permutation that maps 56 bits of Shift’s output to 48 bits

Methodology:

- We perform differential cryptanalysis using two 3-round characteristics and used chosen-plaintext attack for cryptanalysis of 6-round DES. The characteristics used are 40080000 04000000 and 00200008 00000400.

- We know from the spirit's instructions that one byte contains

two characters, hence one character is represented by four bits. We can only represent 16 characters with 4 bits, therefore we attempted a few plaintexts and compared the ciphertexts to see which 16 characters are utilised in the game. After analysing the ciphertexts we inferred that alphabets d to s are used in the game.
Therefore, we proceeded by mapping letters d-s to 0-15 respectively:
{d : 0000, e : 0001, f: 0010, g : 0011, h : 0100, i : 0101, j : 0110, k : 0111, l: 1000, m : 1001, n : 1010, o : 1011, p : 1100, q : 1101, r : 1110, s : 1111}

- The input and output size of one DES block is 64 bits which is 8 bytes (block size) which means 16 letters. So, we decided to work on plaintexts of size 16 letters.

Step 1: Obtaining Plaintext Pairs

The differential characteristic $40\ 08\ 00\ 00\ 04\ 00\ 00\ 00$ with probability 1/161/16 and $00\ 20\ 00\ 08\ 00\ 00\ 04\ 00$ with probability 1/161/16 are used. We generated 4000 pairs of plaintexts and ciphertexts corresponding to each characteristic to break 6-round DES. The first 2000 plaintext pairs are generated such that their XOR was $00\ 00\ 80\ 10\ 00\ 00\ 40\ 00$, which is obtained by applying inverse initial permutation on the characteristic $40\ 08\ 00\ 00\ 04\ 00\ 00\ 00$ and another 1000 plaintext pairs such that their XOR was $00\ 00\ 08\ 01\ 00\ 10\ 00\ 00$, which is obtained by applying inverse initial permutation on the characteristic $00\ 20\ 00\ 08\ 00\ 00\ 04\ 00$. These inputs are stored in $plaintexts1.txt$ and $plaintexts2.txt$ respectively. The code for generation of plaintext pairs is in generate_plaintexts.ipynb.

Step 2: Obtaining Ciphertext corresponding to t

For automating the collection of ciphertexts corresponding to the plaintexts, we used Python's pexpect library to establish connection to the server using valid credentials. We used $server1.py$ to generate the ciphertexts for the plaintexts stored in $plaintexts1.txt$ and $server2.py$ to generate the ciphertexts for the plaintexts stored in $plaintexts2.txt$. These ciphertexts are stored in $ciphertexts1.txt$ and $ciphertexts2.txt$ respectively.

Step 3: Obtaining the key bits of round key K6

Steps 3.1 to 3.4 were carried out for the ciphertexts obtained corresponding to each of the two characteristics.

- $\mathbf{3.1}$: We used the mapping of characters defined above to convert the obtained ciphertext to binary and then, we used MainCode.ipynb to apply reverse final permutation on these binary ciphertexts to get (L_6R_6) and $(L'_6R'_6)$, which is output of the 6^th round of DES. We know that, $R_5 = L_6$, therefore using the values R_5 and R'_5, we computed output of Expansion box and input XOR of S-boxes for 6^th round.

$\mathbf{3.2}$: For the first characteristic mentioned above, $L_5 = 04000000$ and for the second characteristic $L_5 = 00000400$. We found output of 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.

$\mathbf{3.3}$: Let $E(R_5) = \alpha_1\alpha_2\cdots\alpha_8$ and $E(R'_5) = \alpha'_1\alpha'_2\cdots\alpha'_8$ 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 $k_6 = k_{6,1}k_{6,2}\cdots k_{6,8}$. At this point, we know $\alpha_i,\ \alpha'_i,\ \beta_i \oplus \beta'_i$ and $\gamma_i \oplus \gamma'_i$. We created a 8 * 64 key matrix to store the number of times a key $k \in [1, 64]$ satisfies the possibility of being a key to S_i box, where $i \in [1, 8]$. .

- *3.4* : We computed the set $X_i = (\beta, \beta')|\beta \oplus \beta' = \beta_i \oplus \beta'$ and $S(\beta) \oplus S(\beta') = \gamma_i \oplus \gamma'_i$.
Then, we found the key k, such that $\alpha_i \oplus k = \beta$ and $(\beta, \beta') \in X_i$ for some β'. 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 40 08 00 00 04 00 00 004008000004000000:

S-box	Key	Max_Key_frequency	Mean_Key_frequency
S1	45	129	66
S2	51	298	74
S3	37	124	63

S4	7	100	64
S5	54	165	73
S6	41	292	76
S7	28	187	70
S8	55	182	71

For this 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_6K
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 which further assures of these key values being correct. We proceeded by taking the key bits for S2, S5, S6, S7 and S8 boxes as 51, 23, 52, 28 and 54 respectively.

- The above analysis gave the following results for characteristic 00 20 00 08 00 00 04 000020000800000400:

S-box	Key	Max_Key_frequency	Mean_Key_frequency
S1	45	173	74
S2	51	176	68
S3	37	136	72
S4	7	311	77
S5	54	165	71
S6	41	313	79
S7	28	121	68
S8	48	116	68

For this characteristic, in round 4, XOR will be zero for S1, S2, S4, S5 and S6. Therefore, in round 6 these S-boxes will give the corresponding key bits of K_6K
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 S4, S6, S2, S1 and S5 boxes as 7, 41, 51, 45 and 54 respectively.

Both the characteristics have S2, S5 and S6 as common S-boxes and we obtained same key values for these three S-boxes 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, 54, 41, 28 and 48 for round key K_6K
6

 . Thus, at this point we know 42 bits of the 56 bit key.

Step 4: Obtaining the Actual Key from 42 know

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:

X11XX1XX01011X100XX11X11000X1110110X01110100X11X1011X00
1   (Master Key)
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}2
14
  possible permutations of the key to find the correct key. We took plaintext= dddddddddddddddd and the corresponding ciphertext= mnlmisgkerjkileg 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 condition:

$$\textbf{Actual 56 Bit Key} = \mathbf{011011100101111001111101}$$

After obtaining the 56 bit key, we found the 48 bit round key for each round.

ROUND	KEY IN BINARY
Round 1	111011000100111100000111110101101111101100101011
Round 2	011011110011011101100010001111011100110100001111
Round 3	111010101101010011101101110011100101010011010110
Round 4	110110011100001101011010110011011110001111101101
Round 5	001001001101101110111011101100101101111011001001
Round 6	101101110011100101000111110110101001011100110111

Step 5: Decryption of Password

-The ciphertext corresponding to our password is "jnrkdgqefggnglmromdppogkhfnkreie" and therefore to obtain the password we performed decryption on this ciphertext. This ciphertext consists of 32 characters. Since, each character is represented by 4 bits, so this is 128 bit string, that is, 2 blocks of DES ciphertext. As per our mapping this is {106, 231, 3, 209, 35, 58, 56, 158, 185, 12, 203, 55, 66, 167, 225, 81}

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

-The plaintext obtained is - poptdzfims000000. We removed the zeroes as they must have been used for padding.

- We entered the plaintext **poptdzfims** in the game and were directed to the next level. Finally, this was our password!

References: https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/

No files uploaded

Q5 Password

5 Points

What was the password used to clear this level?

poptdzfims

Q6 Codes

0 Points

Unlike previous assignments, this time it is MANDATORY that you upload the codes used in the cryptanalysis. If you fail to do so, you will be given 0 marks for the entire assignment.

▾ Enciphered.zip Download

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

Assignment 4

● GRADED

GROUP

Gargi Sarkar
Anindya Ganguly
Utkarsh Srivastava
View or edit group

TOTAL POINTS

72.5 / 100 pts

QUESTION 1	
Team Name	**0** / 0 pts
QUESTION 2	
Commands	**10** / 10 pts
QUESTION 3	
CryptoSystem	**5** / 5 pts
QUESTION 4	
Analysis	**80** / 80 pts
QUESTION 5	
Password	**5** / 5 pts
QUESTION 6	
Codes	**-27.5** / 0 pts