Q1 Teamname
0 Points

Cipherberg

Q2 Commands
10 Points

List the commands used in the game to reach the ciphertext.

go, dive, dive, back, pull, back, back, go, wave, back, back, thrnxxtzy, read, 360852885036840078603672 5, c, read

Q3 Cryptosystem
5 Points

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

6-round DES (Block cipher)

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. (Explain in less than 150 lines and 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 first needed to retrieve the magic wand from the bottom of the river. Then we went back and freed the spirit in level 3. After coming back to first screen of level 4, we typed ‘read’, read a few

hints by the spirit from the glass panel and typed ‘password’ as was instructed in the message. We then got the ciphertext - ‘nhtkjqjupqholpqflhmnlsjrhfkjitnt’. We had to decrypt this ciphertext to cross level 4.

The screen had several hints of the cipher at Level 4 being encrypted using DES. It was mentioned that it can be 4, 6 or 10-round DES. Since, 4-round DES is easy to break and 10-round DES will be quite difficult to break, also, the spirit says, "but this one surely is not 10- round..." which makes it being 10-round DES even less likely. Hence we went ahead with 6-round and tried deciphering the password by breaking down 6-round DES with the code provided in des.txt, thinking that if this did not work we could later apply the same approach for 4-round DES.

We used chosen-plaintext attack to break the 6-round DES. In this attack model, for cryptanalysis the attacker creates samples of plaintexts, gets the sender to encrypt them and then uses 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.

- It was mentioned that one byte consists of 2 characters, therefore 4 bits are used to represent one character. Using 4 bits we can only represent 16 characters, hence, we tried several plaintexts and evaluated the corresponding ciphertexts to identify which 16 characters are used in the game. After analysing the ciphertexts we inferred that alphabets f to u are used in the game. Therefore, we proceeded by mapping letters f-u to 0-15 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 letters. Therefore, we decided to work on plaintexts of size 16 letters.

Step 1: Generation of Plaintext Pairs

The differential characteristic $40\ 08\ 00\ 00\ 04\ 00\ 00\ 00$ with probability $1/16$ and $00\ 20\ 00\ 08\ 00\ 00\ 04\ 00$ with probability $1/16$ are used. We generated 1000 pairs of plaintexts and ciphertexts corresponding to each characteristic to break 6-round DES. The first 1000 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_inputs.ipynb.

Step 2: Obtaining Ciphertexts corresponding to th

To automate collection of ciphertexts corresponding to the plaintexts, we used Python's pexpect 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: Finding 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.

- 3.1 : We used the mapping of characters defined above to convert the obtained ciphertext to binary and then, we used $CryptAnalysis.ipynb$ 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 $R_5 \text{ and } R'_5$, we computed output of Expansion box and input XOR of S-boxes for 6^{th} round.

- 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.

- 3.3 : Let $E\left(R_5\right) = \alpha_1\alpha_2\cdots\alpha_8 \text{ and } E\left(R'_5\right) = \alpha'_1\alpha'_2\cdots\alpha'_8$ and $\beta_i = \alpha_i \oplus k_{6,i}$ and $\beta'_i = \alpha'_i \oplus k_{6,i}, \text{where } |\alpha_i| = 6 = |\alpha'_i| \text{ and } k_6 = k_{6,1}k_{6,2}\cdots k_{6,8}$. At this point, we know $\alpha_i, \alpha'_i, \beta_i \oplus \beta'_i \text{ and } \gamma_i \oplus \gamma'_i$. 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 $X_i = (\beta, \beta')|\beta \oplus \beta' = \beta_i \oplus \beta'_i \text{ and } S(\beta) \oplus S(\beta') = \gamma_i \oplus \gamma'_i$. Then, we found the key k, such that $\alpha_i \oplus k = \beta \text{ and } (\beta, \beta') \in X_i \text{ for some } \beta'$. 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	Key	Max_Key_frequency	Mean_Key_frequency
S1	45	132	68
S2	51	293	77

S3	37	128	67
S4	7	114	69
S5	23	136	67
S6	52	292	77
S7	28	175	75
S8	54	175	69

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_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 0020000800000400:

S-box	Key	Max_Key_frequency	Mean_Key_frequency
S1	45	152	70
S2	51	170	66
S3	37	124	65
S4	7	297	77
S5	23	146	69
S6	52	279	76
S7	28	116	69
S8	54	112	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_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, 7,23 and 52 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, 23, 52, 28 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:

X11XX1XX01011X100XX11X11000X1010010X11110000X11X1101X011 (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} possible permutations of the key to find the correct key. We took plaintext= ffffffff ffffffff and the corresponding ciphertext= nuljnlif npfklkup 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:

Actual 56 Bit Key = 01101110010111100111101100

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

ROUND	KEY IN BINARY
Round 1	11101100 01001111 00000111 01010110 11110111 00100100
Round 2	01101111 00110111 01100010 00011100 11011000 01001111
Round 3	11101010 11010100 11101101 01000110 11110100 11110100
Round 4	11011001 11000011 01011010 10101001 10101101 11101001
Round 5	00100100 11011011 10111011 10101010 11011110 00010011
Round 6	10110111 00111001 01000111 01011111 01000111 00110110

Step 5: Decryption of Password

-The ciphertext corresponding to our password is “nhtkjqjupqholpqflhmnlsjrhfkjitnt” 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 {130, 229, 75, 79, 171, 41, 106, 176, 98, 120, 109, 76, 32, 84, 62, 142}

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

-The plaintext obtained is - owtveaeekc000000. We removed the zeroes as they must have been used for padding.

- We entered the plaintext 'owtveaeekc' 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 final command used to clear this level?

owtveaeekc

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.

Assignment-4-Cipherberg.zip Download

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

Assignment 4

GRADED

6 DAYS, 23 HOURS LATE

GROUP

SAMBHRANT MAURYA
DEEKSHA ARORA
SHRUTI SHARMA
View or edit group

TOTAL POINTS

90 / 100 pts

QUESTION 1

Teamname — **0** / 0 pts

QUESTION 2

Commands — **0** / 10 pts

QUESTION 3

Cryptosystem — **5** / 5 pts

QUESTION 4

Analysis — **80** / 80 pts

QUESTION 5

Password — **5** / 5 pts

QUESTION 6

Codes — **0** / 0 pts