Q1 Team Name
0 Points

Group Name

the_boys

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->go->back->enter->wave->back->back->thrnxxtzy->read->the_magic_of_wand->c->read

Q3 Cryptosystem
10 Points

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

6-Round-DES

After analyzing the spirit's hint, it was concluded that the cryptosystem used in level 4 was either 4 round DES or 6 round DES. It was highly unlikely that it was 10 round DES. We assumed it to be 6 round DES and started our analysis. And, finally we concluded that the system was 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.)

Upon inputting the password in the level 4 screen, the resultant ciphertext obtained is "mgktrirhioiggupnkmipojgrsupqgquk".

After analyzing the spirit's hint, it was concluded that the cryptosystem used in level 4 was either 4 round DES or 6 round DES. It was highly unlikely that it was 10 round DES. We assumed it to be 6 round DES and started our analysis. The hint "two letters for one byte" indicated that each letter was represented using 4 bits, allowing only 16 out of 26 letters to be used. By trying various random plaintexts as input, we found that the ciphertext contained letters from f to u. Thus, we only used letters from [f,u] while generating plaintext for the attack. The block size of DES is 8 bytes, meaning that each block contains 16 letters.
To break the DES encryption, we employed the chosen plain text attack. We used differential cryptanalysis to generate plain text pairs and sent them to the system to obtain corresponding ciphertext pairs. We then used these pairs to find the key and used it to decrypt the given ciphertext.

Method:

- The method used to perform differential cryptanalysis involved generating pairs of plaintexts that satisfied certain characteristics. These characteristics were defined as 2 3 round characteristics with a probability of 0.0625 each and were represented as hexadecimal values: 40 08 00 00 04 00 00 00 and 00 20 00 08 00 00 04 00.

To generate 5000 pairs of plaintexts that satisfied the first characteristic (40 08 00 00 04 00 00 00), we used gen_plaintext.py file ,we ensured that their XOR value was 00 00 80 10 00 00 40 00, which was obtained by applying the inverse initial permutation on the aforementioned characteristic. Similarly, 5000 pairs of plaintexts that satisfied the second characteristic (00 20 00 08 00 00 04 00) were generated with an XOR value of 00 00 08 01 00 10 00 00 obtained by applying the same inverse initial permutation using gen_plaingen.py.
The generated pairs of plaintexts were stored in two

separate files named pIaintexts1.txt and pIaintexts2.txt. These plaintexts were later used to generate corresponding ciphertexts using the run_script.py program, which were then stored in ciphertexts1.txt and ciphertexts2.txt files.

- The next step in the differential cryptanalysis process was to generate the corresponding ciphertexts for the previously generated plaintext pairs. This was done using the "run_script.py" program. For each plaintext pair, "run_script.py" executed the encryption algorithm and produced the corresponding ciphertexts. These ciphertexts were then stored in two separate files named "ciphertexts1.txt" and "ciphertexts2.txt". The ciphertexts would be used in the next step of the differential cryptanalysis process to deduce information about the encryption key.

- Differential cryptanalysis was performed to find the key using diffCryptoAnalysis.py. Firstly, we read ciphertext1.txt, and for each ciphertext, we converted each letter into binary using a mapping where f is mapped to 0000 and u is mapped to 1111. Then, we compared pairs of plaintexts and ciphertexts to identify a difference in the output, known as a "differential". Next, we analyzed the differentials to identify a pattern in the key. This process was repeated for ciphertext2.txt to further refine the possible keys. Finally, we narrowed down the possible keys to a few candidates and tested them until they found the correct key.

To find the output of the expansion box and input XOR of sboxes for the 6th round, we applied the inverse final permutation to obtain (L6, R6) and (L'6, R'6). For the first characteristic, we know that R5=L6 and L5 = 04 00 00 00. For the second characteristic, L5 = 00 00 04 00 . Using these values, we then perform L5⊕ (R6 ⊕ R'6) and apply the inverse permutation to obtain the output XOR of sboxes for the 6th round.

Let

$$E(R5) = \alpha 1\ \alpha 2\ \alpha 3\ \ldots\ldots\ \alpha 8 \quad \text{and } E(R5') = \alpha'1\ \alpha'2\ \alpha'3\ \ldots\ldots\ \alpha'8$$

where

$$|\alpha i| = 6 = |\alpha' i| \quad \text{---(1)}$$

$$k6 = k6,1 \quad k6,2\ \ldots\ldots\ k6,8 \quad \text{----(2)}$$

$$\beta i = \alpha i \oplus k6,i \text{ and } \beta i' = \alpha' i \oplus k6,i \quad \text{-----(3)}$$

we know so far,

$$\alpha i\ ,\ \alpha' i\ ,\ \beta i \oplus \beta i' \text{ and } \gamma i \oplus \gamma' i$$

We first created a key matrix which has size 8x64. This matrix was used to store the number of times a key $k \in [1, 64]$, and satisfies the possibility of being a key to Si box, where $i \in [1, 8]$.

Once this key matrix was created, we proceeded to find a set based on certain conditions. This set was found using the equation:
$Xi = \{(\beta, \beta') \mid \beta \oplus \beta' = \beta i \oplus \beta i' \text{ and } Si(\beta) \oplus Si(\beta') = \gamma$
This set played a crucial role in the next step of our procedure.

After creating the 8x64 key matrix and finding the set of possible keys that satisfy the given condition, the next step is to check each key in the set. For each key $k \in [1, 64]$, we check whether the following conditions are satisfied:
$\alpha i \oplus k = \beta$ and $(\beta, \beta') \in Xi$ for some β'
If both these conditions are satisfied for a particular key k, we increment the count of the corresponding entry in the 8x64 key matrix. This process is repeated for all keys in the set [1, 64].

To perform the differential cryptanalysis, we analyzed the ciphertexts generated from plaintexts in pIaintexts1.txt

and pIaintexts2.txt using diffCryptoAnalysis.py. For If the condition was satisfied for Si box, then we incremented key[i][k] by 1.
The result of the above analysis for the first characteristic 40 08 00 00 04 00 00 00 was a partial key using S2, S5, S6, S7, S8 as 51, 60, 56, 29 and 54, respectively. The input to these sboxes was 0 in round 4. Similarly, we repeated the above procedure for ciphertexts in ciphertexts2.txt, and the result of the analysis for the second characteristic 00 20 00 08 00 00 04 00 was a partial key using S1, S2, S4, S5, S6 as 45, 51, 7, 60, and 56, respectively. The input to these sboxes was 0 in round 4.

The characteristics obtained from the previous steps have S2, S5, S6 in common, and from these characteristics, we were able to deduce some key bits for these specific sboxes. The deduced key bits were found to be the same for both characteristics. Using these results, we have successfully found 42 out of the 56 bits of the key.
The resulting 48-bit key for the specific sbox is represented as:
101101110011XXXXXX000111111100111000011101110110
Here, 'X' are inserted in the position of S3 because input to S3 was never zero. This 48-bit key is then converted into a 56-bit key using Key schedule PC2. The resulting 56-bit key is:
X11XX1XX01011X100XX11X11000X0010111X01111000X11X1111X001

To find the missing bits of the key, a brute force method was used. Firstly, all 2^14 possible keys were iterated through. Then, the plaintext "fghijklmnopqrstu" was passed as input to the system, resulting in the ciphertext "lfmfqmsiugogktlk". For each possible key, the plaintext was encrypted with the key and checked if the resulting cipher matched the above ciphertext. The key for which the output of encryption and the above ciphertext matched was identified as the actual key.
The actual key was found to be:
01101110010111100111011100011001001001000100001
With the actual key, it is possible to decrypt any ciphertext encrypted using the same key.

• Decryption of password -
The password given to us, "mgktrirhioiggupnkmipojgrsupqgquk", was first converted from ASCII characters into binary code. Then, we split the resulting binary code into two blocks of 64 bits each, since the DES algorithm processes 64-bit blocks of plaintext at a time.
The first block, {113, 94, 195, 194, 57,49, 31, 168}, and the second block, {87, 58, 148, 28, 223, 171, 27, 245}, were then passed one at a time into a program written in the C++ programming language that implements the DES decryption algorithm.
After running the decryption process, the resulting plaintext was "pquxezeglb000000". We believed that the "000000" at the end of the plaintext might be padding, so we tried the password "pquxezeglb" without the padding, and we successfully cleared the level.

Final Password ---> pquxezeglb

References -
- https://link.springer.com/content/pdf/10.1007/3-540-48071-4_34.pdf
- https://medium.com/lotus-fruit/breaking-des-using-differential-cryptanalysis-958e8118ff41

Q5 Password
5 Points

What was the password used to clear this level?

pquxezeglb

Q6 Code
0 Points

Please add your code here. It is MANDATORY.

▾ the_boys_assignment4.zip Download

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

Assignment 4

● Graded

Group

SANKET SANJAY KALE
PRATIK MAHIPAL PATIL
ADITYA SUNILKUMAR KANKRIYA
View or edit group

Total Points

100 / 100 pts

Question 1

Team Name — **0** / 0 pts

Question 2

Commands — **5** / 5 pts

Question 3

Cryptosystem — **10** / 10 pts

Question 4

Analysis — **80** / 80 pts

Question 5

Password — **5** / 5 pts

Question 6

Code — **0** / 0 pts