Q1 Team Name
0 Points

Lazarus

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 -> dive -> dive -> back -> pull -> go -> back -> enter -> wave -> back -> back -> thrnxxtzy -> read -> 134721542097659029845273957 -> c -> read

Q3 CryptoSystem
5 Points

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

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.)

After entering "password" on the level 4 screen, we get our ciphertext which is

”leohimkjkshooerdqqngsgmnjjmjpfdm”

From the spirit's hint, we understood that the cryptosystem for this level is either 4 round DES or 6 round DES. The chances of it being 10 round DES were significantly less. So we started with assuming 6 round DES.
From the hint "two letters for one byte" we inferred that each letter is represented using 4
bits, so only 16 out of 26 letters are possible. By giving multiple random plaintexts as input, we observed that letters from d to s were present in ciphertext, so while generating plaintext for the attack, we used letters only from [d,s]. The block size of DES is 8 bytes, so each block contains 16 letters.
Chosen plain text attack is used to break DES encryption. In this, we used differential cryptanalysis to generate plain text pairs, pass them to the system to get corresponding ciphertext pairs, and then used these to find the key and then used it to decrypt the above ciphertext.

METHOD
1. Using plaingen.py we generated 5000 pairs of plain text for each characteristic. We have used 2 3 round characteristic with probabilty of 0.0625 each. The characteristics are

40 08 00 00 04 00 00 00
00 20 00 08 00 00 04 00

To generate 5000 pairs satisfying 40 08 00 00 04 00 00 00 characteristic we ensured that their xor is
00 00 80 10 00 00 40 00 which is obtained by applying inverse initial permutation on the before mentioned charateristic.
Similarly, we generated 5000 pairs of plaintext satisfying
00 20 00 08 00 00 04 00 characteristic we ensured that their xor is 00 00 08 01 00 10 00 00 which is obtained by applying inverse initial permutation on the before mentioned characteristic. These plain texts are stored in plaintexts1.txt and

plaintexts2.txt, respectively.
2. We executed robot.py to generate ciphertexts corresponding to plaintexts and stored them in ciphertexts1.txt and ciphertexts2.txt.
3. Differential cryptanalysis was performed to find the key -
The below process is done using differentialanalysis.py
We first read ciphertext1.txt and for each ciphertext, we convert each letter into binary using the mapping where d is 0000 and s is 1111.

We applied the inverse final permutation. To get $(L6, R6)$ and $(L'6, R'6)$. We know that $R5 = L6$, so we use $R5$ and $R'5$ to find the output of the expansion box and input XOR of sboxes for the 6th round.

$L5 = 04\ 00\ 00\ 00$ for first characteristic and $L5 = 00\ 00\ 04\ 00$ for second charateristic. Then we perform $L5 \oplus (R6 \oplus R'6)$then apply inverse permutation to get output XOR of sboxes for 6th round.

Let

$$E\left(R5\right) = \alpha_1\alpha_2\cdots\alpha_8 \text{ and } E\left(R5'\right) = \alpha'_1\alpha'_2\cdots\alpha'_8$$

where

$$|\alpha_i| = 6 = |\alpha'_i|$$

and

$$k_6 = k_{6,1}k_{6,2}\cdots k_{6,8}$$

and

$$\beta_i = \alpha_i \oplus k_{6,i} \text{ and } \beta'_i = \alpha'_i \oplus k_{6,i}$$

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 $i \in [1, 8]$.

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

$$\alpha_i \oplus k = \beta \text{ and } (\beta, \beta') \in X_i \text{ for some } \beta'$$

If above condition is satisfied for S_i box, then we incremented key[i][k] by 1

Result of above analysis for characteristic
$40\ 08\ 00\ 00\ 04\ 00\ 00\ 00$ is that we get partial key using S2,S5,S6,S7,S8 as 59,6,31,0,50 as input to these sboxes is 0 in round 4

Similarly, we repeat the above procedure for ciphertexts in ciphertexts2.txt

Result of above analysis for characteristic
$00\ 20\ 00\ 08\ 00\ 00\ 04\ 00$ is that we get partial key using S1,S2,S4,S5,S6 as 45,59,7,6,31 as input to these sboxes is 0 in round 4

These characteristics have S2, S5, S6 common, and key bits deduced from both these characteristics are the same for before mentioned sboxes. Therefore we have successfully found 42 out of 56 bits of the key.
48 bit Key for Sbox is

$$101101111011XXXXXX000111000110011111000000110$$

6 'X' are inserted in the position of S3 as input to S3 was never zero. Converting this into a 56-bit key and applying Key schedule PC2, we get

$$X11XX1XX01011X100XX11X11100X1100100X00100010$$

To find the missing bits, we used the brute force method that is we iterated through all 2^{14} possible keys. We passed "defghijklmnopqrs" as input plaintext to the system. We get cipher as "qpnonsgrprdllqml". Then for each possible key, we encrypted the plaintext with this key to check if we got the above cipher. The key with which output of encryption and above cipher matches is the actual key. The key is

$$01101110010111100111101110001100100100100010001$$

4. Decryption of password -
We convert $\mathrm{leohimkjkshooerdqqngsgmnjjmjpfdm}$ 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 $\{129, 180, 89, 118, 127, 75, 177, 224\}$ and $\{221, 163, 243, 154, 102, 150, 194, 9\}$ where each block is 8 bytes and this is passed one at time in des.cpp.
After Decryption we got

$$\mathrm{rtrmibcrhe000000}$$

We thought '000000' at the end might be padding, so we tried 'rtrmibcrhe' as the password, and we successfully cleared the level.

References -
Differential Crypt analysis of DES-like Cryptosystems(Extended Abstract)
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjNg_zIirz2AhWqGaYKHdNVA2EQFnoECAMQAQ&url=https%3A%2F%2Flink.springer.com%2Fcontent%2Fpdf%2F10.1007%252F3-540-38424-3_1.pdf&usg=AOvVaw3LFLfLsA4_y83jxT8n8rTH

No files uploaded

Q5 Password
5 Points

What was the password used to clear this level?

rtrmibcrhe

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.

a4.zip Download

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

Assignment 4

GRADED

GROUP
Varun Vankudre
Aditya Loth
Harsh Agarwal
View or edit group

TOTAL POINTS
45 / 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

✔	+ 10 pts	Mentioning that the plaintext and ciphertext contain letters in the range *dd* to *ss* and the mapping of these letters to bytes.
✔	+ 20 pts	Mentioning the method (or code) used to attack the server to collect plaintext-ciphertext pairs.
✔	+ 5 pts	Mention the characteristics used.
✔	+ 5 pts	Mentioning the probability and thus how many pairs are required.
✔	+ 20 pts	How the characteristics help find certain key bits.
✔	+ 10 pts	Brute-forcing for the rest of the key bits and finding the main key.
✔	+ 5 pts	Mentioning the plaintext password, i.e., the password padded with 0's.
✔	+ 5 pts	Figuring out the final command from the plaintext password.
	+ 0 pts	NA

QUESTION 5

Password **5** / 5 pts

QUESTION 6

Codes **-55** / 0 pts