Q1 Team Name
0 Points

Noobs

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)

1. go
2. dive
3. dive
4. back
5. dive
6. pull
7. back
8. back
9. enter
10. wave
11. back
12. thmxxtzy
13. read
14. 13472154209765902984527395 7
15. c
16. read
17. password

Note: Command number 15(c) is not a regular command. It was just asked by the prompt to enter c to continue to next level, so, we mentioned it.

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. (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 jumped into the water and drew out the magic wand at first. We returned to level 3 after discovering the magical wand to free the spirit there. Now we return to level 4, where we type "read" on the screen. The spirit reads out the 4th level hints and instructs us to write 'password.' We obtain the ciphertext 'kqfegpfhogpsgfgrqkieensekesfhmkq' after typing 'password.'

We see the message that the ciphertext is encrypted in DES and that it can be any 4,6,10-round DES after we acquire the ciphertext. 'He doesn't remember which round DES it is, but it is definitely not the 10 round DES,' the ghost adds. So, it's either a 4-round or 6-round DES. Now that the 4-round DES is simple to decrypt let's try the 6-round DES first. Using the code in des.txt, we decrypt the 6-round DES. We can always fall back to the 4-round DES if this attempt fails.

We broke 6-round DES using the chosen-plaintext attack method. Now, we send plain text from the sender side and receive an encrypted text from the receiver side in this approach. After that, we perform a pairwise analysis of the plaintext and ciphertext to obtain all the encrypted words.

Now we have used following steps to get the password.

Step 1: Making of Plaintext Pairs

In this step, we created some plaintext pairings that, when XORed, yield "0x405C0000 04000000." The adjusted XOR will be "0x00009010 10005000" after some early permutations, and the following permutation of this, we will have the desired XOR value. We've created 100000 plaintext pairs that satisfy the stated XOR criterion. Code used in this step: - Plain Texts Generation.ipnyb
Plaintext pairs: - plaintexts.txt

Step 2: Obtaining corresponding Ciphertexts

We've now used the python script to connect to the server and generate the plaintext's corresponding ciphertexts using the code in 'CipherTextGeneration.ipynb.' Now it's time to convert the retrieved ciphertext to binary format.
We were suggested in the question that each character has a 4bit binary value. As a result, there must be a total of 16 characters. We know the ciphertext characters are between 'd' and 's'; as a result, the characters' binary representations must be as follows.
d=0000, e= 0001, f=0010, … , s=1111

Files for Code used: - CipherTextGeneration.ipynb and CipherTextProcessing.ipynb
Output stored in file: - binciphertexts.txt and binaryciphertexts.txt

Step 3: Finding the key bits of round key

The binary format of the ciphertext is obtained from the preceding step. The L6(left) and R6(right) parts of the sixth round are obtained by using the inverse permutation. In the DES algorithm, the left component L6 of the 6th round equals the right part R5 of the 5th round. The R5 is currently undergoing expansion and will be XOR'ed.
Because we're utilizing differential cryptanalysis, the EBox (eop.txt) pairings are now XORed to yield SBox(sip.txt). The R6 pair is XOR'ed with the L5 pair 04000000, which can be obtained by translating characters to binary. The inverse permutation of the XOR'ed value yields the appropriate SBox values (sop.txt).

Code file :-Sixth Round Differential Cryptanalysis.ipynb
Input file:- binciphertexts.txt
Output file :- eop.txt,sip.txt,sop.txt

We have three values from the foregoing analysis: S-box input XOR values, S-box output XOR values, and E-Box output XOR values. We can produce key values depending on features using these values. For example, after four rounds, the XOR of text will be "0x00540000 04000000 " with a chance of 0.00038. As a result, the 6th round key for S-Box "S0, S1, S4, S5, S6, S7" can be predicted with 1 probability. We also know that the S-Box key size will be 6 bits. As a result, the frequency of occurrence of keys is determined by applying the above three XOR values to key values in the range 000000 to 111111. The key for that S-Box will be the one with the highest frequency.
Hence the possible keys are as follows:
S-Box0 Possible Key is 101101
S-Box1 Possible Key is 111011
S-Box4 Possible Key is 100101
S-Box5 Possible Key is 110000
S-Box6 Possible Key is 000000
S-Box7 Possible Key is 111101

Code file :- Extract SBox Keys.ipynb

Step 4: Finding the Key from 42 known bits
In this step we have applied the algorithm to obtains the real positions of the known 42 bits in the 56 bit keys and the result is
XX1XX1XXX10XX1X10XXX11XX11X0X0001000X10010100X11X0111X001 (Key K6)
here `X` denotes unknown bits.

Because we have 14 unknown bits, we'll utilise a brute force strategy to find the proper key. So we took "ssssssss sssssss" as plain text and "kdmsmlpo phehehkp" as ciphertext and performed 6-round DES. This plaintext is encrypted by using the final key. The obtained will be
01101110010111100111011100000010001100101000111011000 1 (Actual 56 Bit Key)

Code file:- GeneratePossible56BitKeys.ipynb and BruteForceKeys.ipynb

Step 5: Password Decryption

Our encrypted password is "kqfegpfhogpsgfgrqkieensekesfhmkq." In this phase, we'll decrypt the password. This password is 32 characters lengthy; we divided it into two halves, each 16 characters long, and applied decryption to each portion separately. However, the text value we get isn't valid as a password.

Based on the hint, "2 characters occupy 1 -byte(8 bits)", we've translated the given password into binary form and applied decryption twice (once for each half of the password) to obtain the text to clear this level.

After examining the values acquired, we discovered that they fell within the ASCII range, and when we convert them to ASCII, we get "qownjyhsht0000000." We removed the zeroes as they must have been used for padding. Finally, "qownjyhsht" is accepted as the password.

As we enter 'qownjyhsht' in the game, we are directed to the next level.
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?

qownjyhsht

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.

Noobs.zip Download

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

Assignment 4

GRADED

GROUP
Manu Shukla
Rishabh Lakhwani
Divyansh Bisht
View or edit group

TOTAL POINTS
44 / 100 pts

QUESTION 1	Team Name	**0** / 0 pts
QUESTION 2	Commands	**9** / 10 pts
QUESTION 3	CryptoSystem	**5** / 5 pts
QUESTION 4	Analysis	**80** / 80 pts
QUESTION 5	Password	**5** / 5 pts
QUESTION 6	Codes	**-55** / 0 pts